

# On the in-the-middle algorithm and heuristic and some of its properties

Dag Wedelin, Data Science Division, CSE, Chalmers

Beginning with special cases of linear programming, I will describe these algorithms, and some of their properties. I will also briefly discuss the max-sum problem and related algorithms, and discuss some of the general challenges with numerical propagation algorithms in relation to classical constraint programming.

**CHALMERS**



UNIVERSITY OF GOTHENBURG

# Crew pairing screenshot

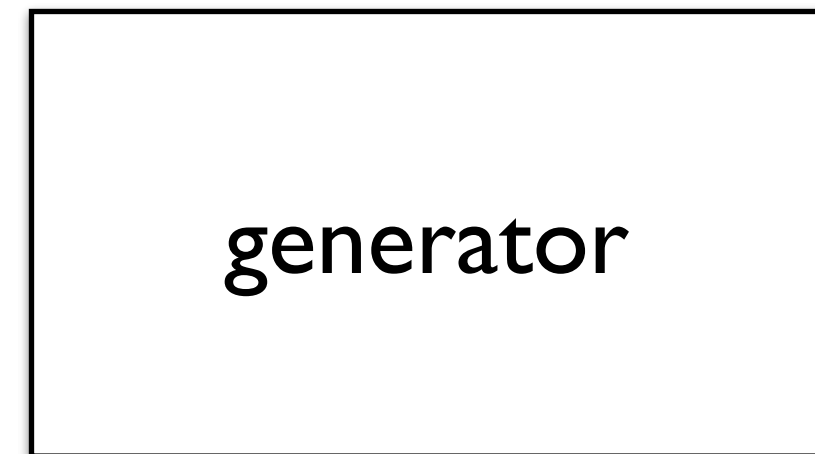
| Graphic Pairing Construction |       |             |             |        |         |              |       |       |             |       |       |              |         |       |
|------------------------------|-------|-------------|-------------|--------|---------|--------------|-------|-------|-------------|-------|-------|--------------|---------|-------|
| Data                         | Plan  | Rule        | APC         | Report | Options | Help         | 15:25 |       |             |       |       |              |         |       |
| Window                       | 01/11 | 02/11       | 03/11       | 04/11  | 05/11   | 06/11        | 07/11 | 08/11 | 09/11       | 10/11 | 11/11 | 12/11        | 13/11   | 14/11 |
| 1   12                       | FRA   | 3 3         | 3           | LIN    | LIN     | 36 325 325   | FRA   |       |             |       |       |              |         |       |
| 1   12345                    | FRA   | 40          | 40 40 40    | HAM    | HAM     | 40 40 41 41  | HAM   | HAM   | 0           | 3726  | ATH   | ATH          | 373 380 | IST   |
| 1   123456                   | FRA   | 2           | 4 4 4 4     | DUS    | DUS     | 4 4 4 4      | DUS   | DUS   | 4 4 4 4     | DUS   | DUS   | 4 4 4 4      | DUS     |       |
| 1   123456                   | FRA   | 0           | 40 40       | HAJ    | HAJ     | 40 40 41 41  | HAJ   | HAJ   | 40 40 41 41 | HAJ   | HAJ   | 40 40 41 41  | HAJ     |       |
| 1   12                       | FRA   | 31 31       | 3210        | SVO    | SVO     | 3213 481 470 | FRA   |       |             |       |       |              |         |       |
| 1   12356                    | FRA   | 3210        | 3211        | GVA    | GVA     | 45 4         | BRU   | BRU   | 4 36 3      | NAP   |       |              |         |       |
| 1   12                       | MUC   | 40 40       | 3           | BUD    | BUD     | 33 33 3      | MUC   |       |             |       |       |              |         |       |
| 1   12345                    | FRA   | 3 34        | 3816 1515   | IST    | IST     | 380 373      | ATH   | ATH   | 3723 3846   | IST   | IST   | 380 373      | ATH     |       |
| 1   12                       | FRA   | 4 4 4       | GVA         | GVA    | GVA     | 45 4065      | FRA   |       |             |       |       |              |         |       |
| 1   12                       | FRA   | 4 4 4       | CDG         | CDG    | CDG     | 4 4075       | FRA   |       |             |       |       |              |         |       |
| 1   123                      | FRA   | 3 35 4      | BRU         | BRU    | BRU     | 4 3 3 4      | BRU   | BRU   | 4 4816 4801 | FRA   |       |              |         |       |
| 1   123456                   | FRA   | 3846 3847   | 0           | HAM    | HAM     | 40 40        | HAM   | HAM   | 40 40       | HAM   | HAM   | 40 40        | HAM     |       |
| 1   1                        | FRA   | 4 46        | 005         | FRA    |         |              |       |       |             |       |       |              |         |       |
| 1   123                      | FRA   | 3736        | 5-0         | ATH    | ATH     | 3723 3       | LIN   | LIN   | 36 325 325  | FRA   |       |              |         |       |
| 1   1                        | FRA   | 4816 4801   | 10-45 45-10 | FRA    |         |              |       |       |             |       |       |              |         |       |
| 1   123                      | FRA   | 3 3 0       | HAJ         | HAJ    | HAJ     | 0 28 28 4    | GVA   | GVA   | 45 4 4      | FRA   |       |              |         |       |
| 1   1234                     | FRA   | 2 3230 3221 | 10-20 20-40 | DUS    | DUS     | 4 4 4 4      | DUS   | DUS   | 3230 10-20  | SVO   | SVO   | 3213 481 470 | FRA     |       |
| 1   123456                   | FRA   | 481 470     | 0-0 0-0     | STR    | STR     | 40 40        | STR   | STR   | 40 40       | STR   | STR   | 40 40        | STR     |       |
| 1   123456                   | FRA   | 33 3 8      | DUS         | DUS    | DUS     | 4 4          | DUS   | DUS   | 4 4         | DUS   | DUS   | 4 4          | DUS     |       |
| 1   1                        | FRA   | 480 475     | 3 3         | FRA    |         |              |       |       |             |       |       |              |         |       |
| 1   1                        | FRA   | 325 3251    | 5-0 5-0     | FRA    |         |              |       |       |             |       |       |              |         |       |
| 1   1                        | FRA   | 341 341     | 4 4         | FRA    |         |              |       |       |             |       |       |              |         |       |
| 1   1                        | FRA   | 47 471      | 43 43       | FRA    |         |              |       |       |             |       |       |              |         |       |
| 1   123                      | MUC   | 35          | 05          | NAP    | NAP     | 3 35 4       | GVA   | GVA   | 45 3 3      | MUC   |       |              |         |       |

Assign value: 0/0/0/0/0/1/2. Crew filter: On  
 SVO - FRA LH 3211 -1 J 123.56. A320 LH3306 0 F000 0144 M000 1/0/1/0/0/1/2  
 Det : 1605 - 1920 GDOR 1 Date(GDOP): 931101 SSIM 320 area :EU LH LH  
 Local : 1905 - 2020 Crew comp: booked:0/0/0/0/0/1/2

931101 - 931114 : READY  
 A320Nov01No cab14\_scrat PTV  
 352 rows. Dated CRs 931101 - 931104  
 0 931101 - 931114

# In crew pairing system

generate many legal pairings



select an optimal subset of these pairings



solution

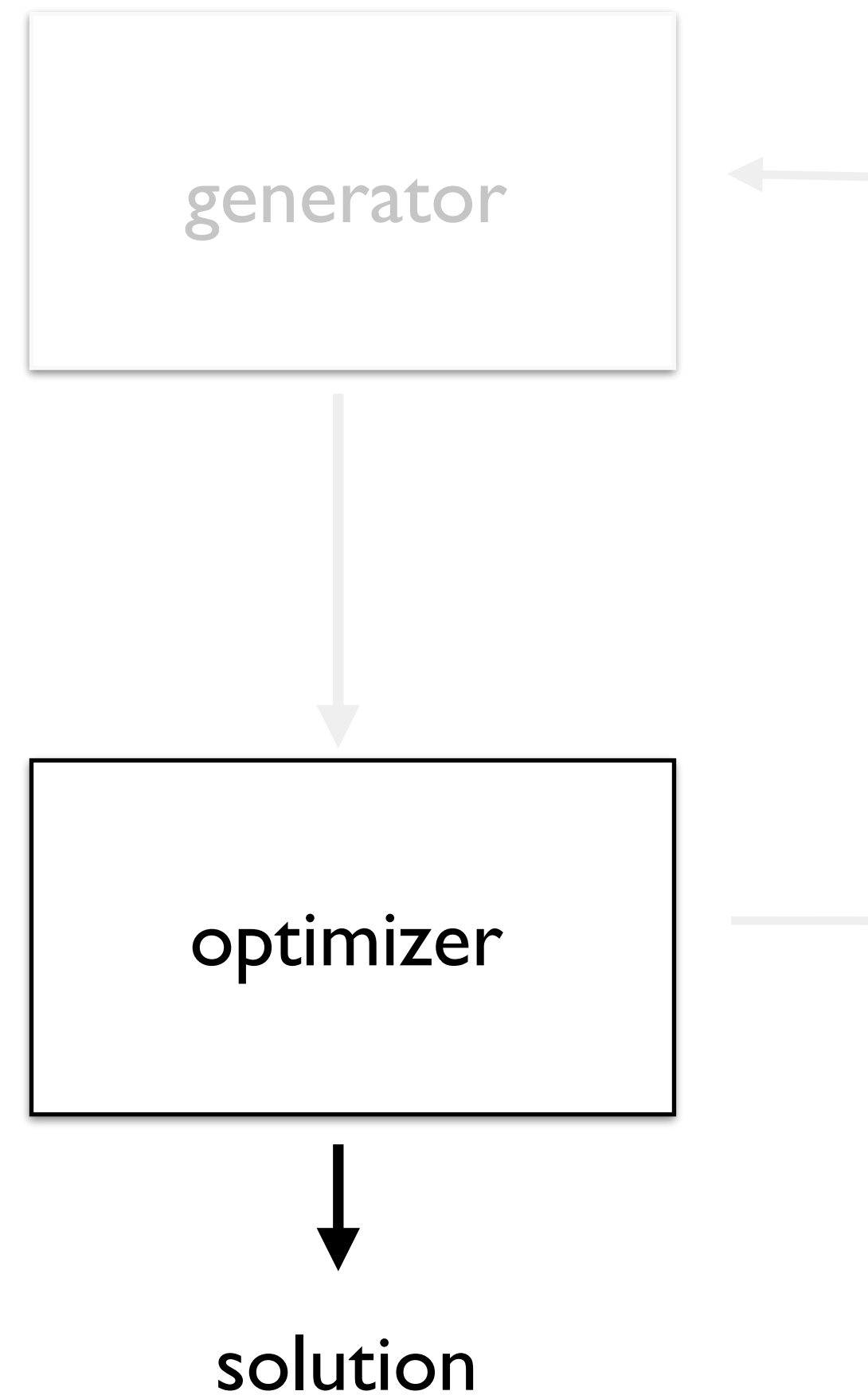
# In crew pariring system

$\min cx$

$Ax \geq 1$  (set covering)

$Cx \leq d$  (base capacity)

$x$  binary vector



# *paqs* optimizer

- in-the-middle algorithm
- in-the-middle heuristic

Regularly benchmarked,  
continuously improved.

In production since  
many years.

Parallel implementation

**the in-the-middle algorithm**

# The simple assignment problem

persons

|   |   |    |   |
|---|---|----|---|
| 3 | 8 | 6  | 4 |
| 4 | 2 | 3  | 0 |
| 2 | 7 | 10 | 8 |
| 6 | 8 | 10 | 5 |

tasks

$$\max C_1 x_1 + C_2 x_2 + \dots + C_{16} x_{16}$$

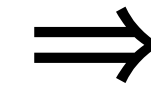
Subject to

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = 1 \\ \vdots \end{cases}$$

$$\begin{cases} x_1 + x_5 + x_9 + x_{13} = 1 \\ \vdots \end{cases}$$

$x$  binary

|   |   |    |   |
|---|---|----|---|
| 3 | 8 | 6  | 4 |
| 4 | 2 | 3  | 0 |
| 2 | 7 | 10 | 8 |
| 6 | 8 | 10 | 5 |



|    |   |    |    |
|----|---|----|----|
| -4 | 1 | -1 | -3 |
| 4  | 2 | 3  | 0  |
| 2  | 7 | 10 | 8  |
| 6  | 8 | 10 | 5  |



|   |   |    |   |
|---|---|----|---|
| 3 | 8 | 6  | 4 |
| 4 | 2 | 3  | 0 |
| 2 | 7 | 10 | 8 |
| 6 | 8 | 10 | 5 |

$\Rightarrow$

|    |   |    |    |
|----|---|----|----|
| -4 | 1 | -1 | -3 |
| 4  | 2 | 3  | 0  |
| 2  | 7 | 10 | 8  |
| 6  | 8 | 10 | 5  |

$\Rightarrow \dots \Rightarrow$

|    |    |    |    |
|----|----|----|----|
| -2 | 1  | -3 | -1 |
| 1  | -3 | -4 | -3 |
| -5 | -2 | -1 | 1  |
| -1 | -1 | 1  | -2 |

Subtract average of largest and second largest numbers.

|   |   |    |   |
|---|---|----|---|
| 3 | 8 | 6  | 4 |
| 4 | 2 | 3  | 0 |
| 2 | 7 | 10 | 8 |
| 6 | 8 | 10 | 5 |

⇒

|    |   |    |    |
|----|---|----|----|
| -4 | 1 | -1 | -3 |
| 4  | 2 | 3  | 0  |
| 2  | 7 | 10 | 8  |
| 6  | 8 | 10 | 5  |

Iterate for all rows and columns until there are no more sign changes.

⇒ ... ⇒

|    |    |    |    |
|----|----|----|----|
| -2 | 1  | -3 | -1 |
| 1  | -3 | -4 | -3 |
| -5 | -2 | -1 | 1  |
| -1 | -1 | 1  | -2 |

Simplest possible algorithm?  
(just subtracting the smallest number does not work)

select assignments with positive cost!

# In-the-middle for 0-1 ILP

$$\max cx$$

$$Ax = b$$

$x$  binary

$A$  contains  $\{-1, 0, 1\}$

$b$  integer

inequalities ok

Consider

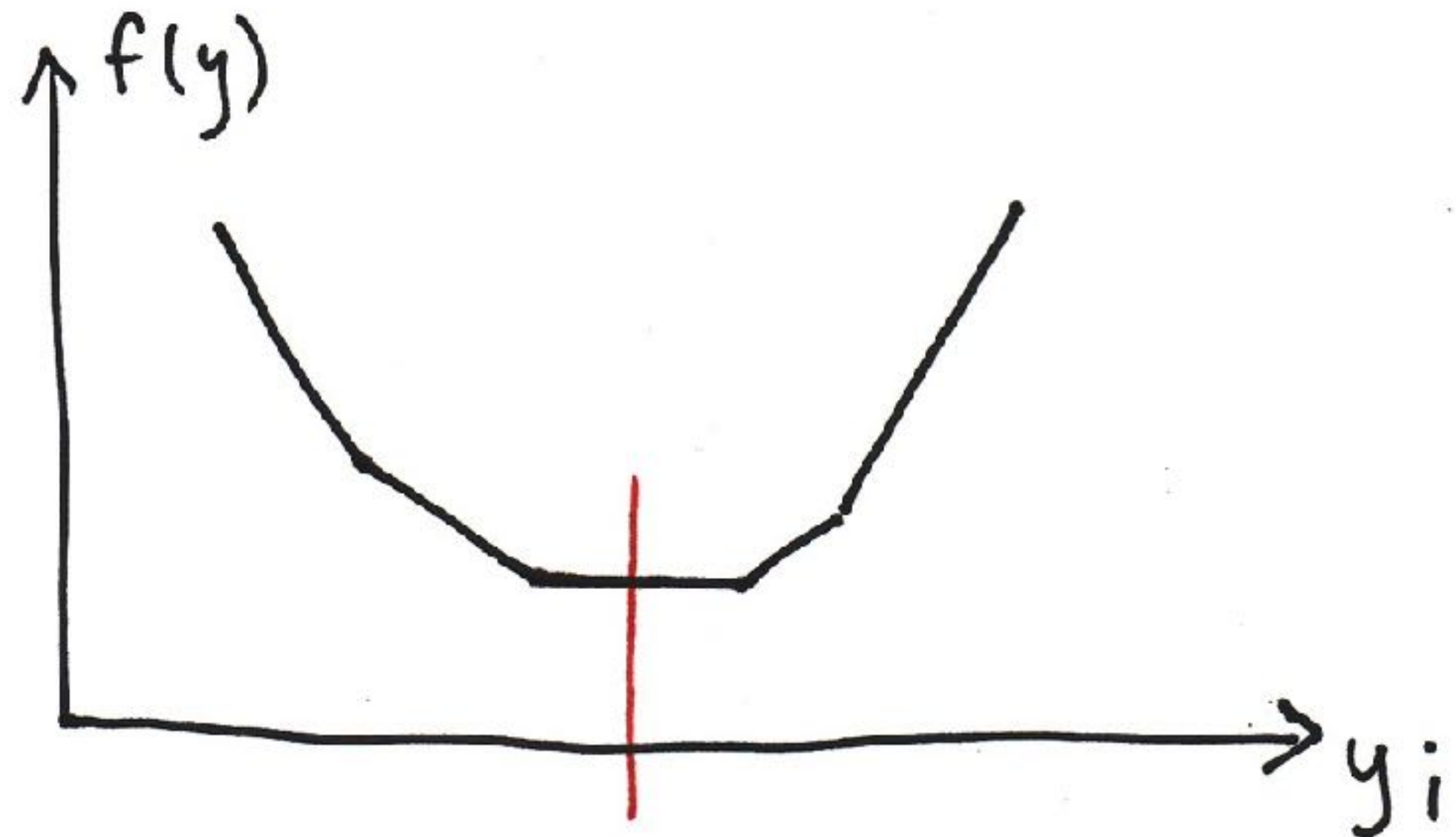
$$\bar{c} = c - yA$$

|    |   |    |    |
|----|---|----|----|
| -4 | 1 | -1 | -3 |
| 4  | 2 | 3  | 0  |
| 2  | 7 | 10 | 8  |
| 6  | 8 | 10 | 5  |

iteratively make single constraints  
feasible by selecting the dual  $y_i$   
in-the-middle of the possible  
interval

# A “dual” algorithm

Minimize piecewise  
linear convex function  
with coordinate  
descent



$$\min_y f(y) = yb + \max_{0 \leq x \leq 1} \bar{c}x$$

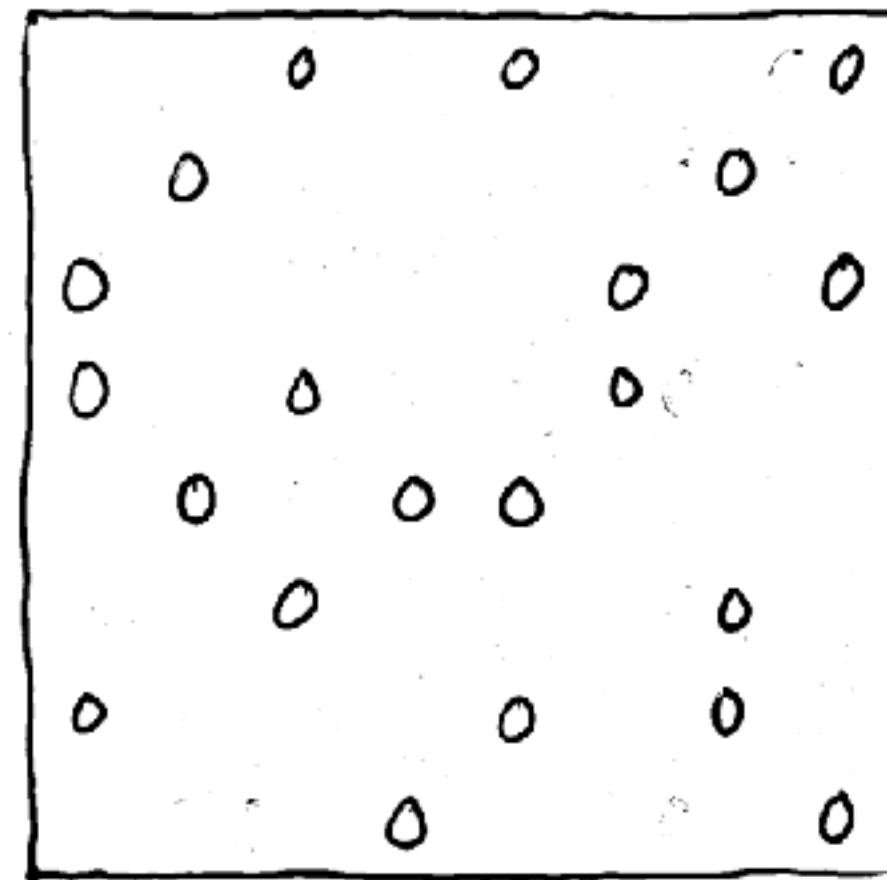
## Theorem:

The in-the-middle algorithm can only solve “easy” ILP problems (the LP relaxation has a unique solution that is integer).

What about convergence?

How does in-the-middle fail  
for difficult ILP's?

a lot of zeros...



eg. 8-queens  
with diagonal  
constraints

## A small difficult problem

$$\max x_1 + x_2 + x_3$$

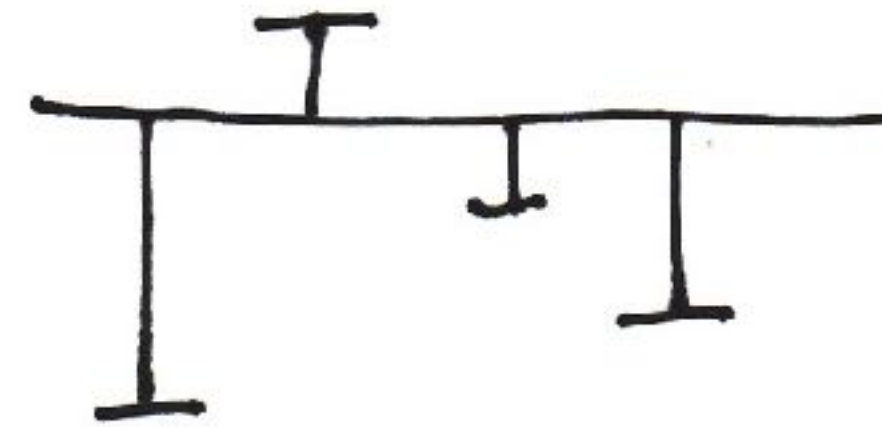
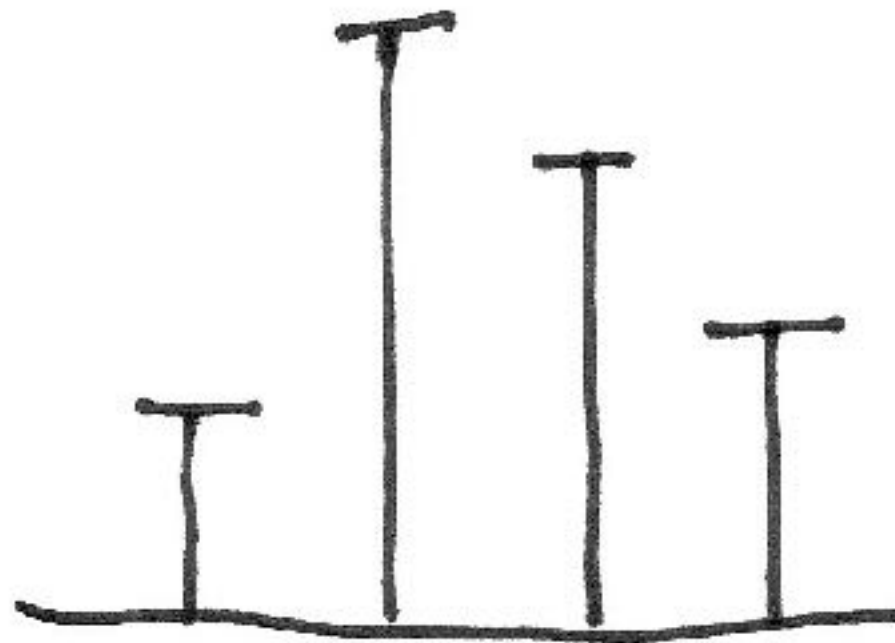
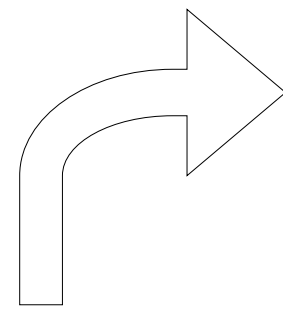
$$x_1 = x_2 + x_3$$

$$x_2 = x_1 + x_3$$

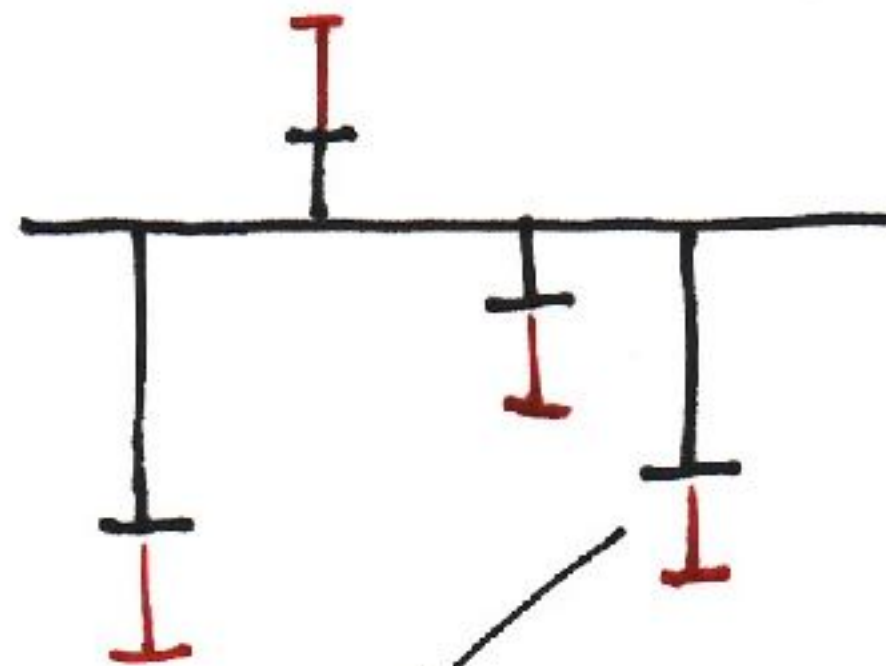
$$x_3 = x_1 + x_3$$

$\bar{c}_1 = \bar{c}_2 = \bar{c}_3 = 0$  is a fixpoint!

# in-the-middle heuristic for difficult 0-1 ILP's



invariant



not invariant!

proportional to  $(r^+ - r^-)$



Now it suddenly works!

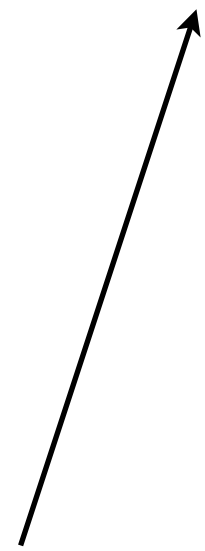
(weighted 8-queens problem)

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 3 | 7 | 5 | 8 | 3 | 2 | 9 |
| 5 | 3 | 8 | 7 | 5 | 1 | 1 | 9 |
| 4 | 3 | 7 | 5 | 8 | 2 | 3 | 8 |
| 7 | 1 | 2 | 4 | 1 | 4 | 6 | 2 |
| 2 | 6 | 2 | 8 | 4 | 4 | 4 | 1 |
| 3 | 5 | 4 | 5 | 4 | 5 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 3 | 2 | 2 | 2 | 2 | 2 | 6 | 8 |

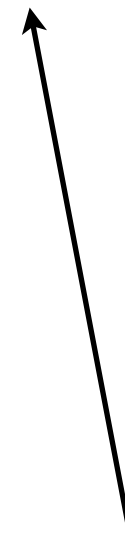
|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

# in-the-middle heuristic

$$y_i = -\frac{r^+ + r^-}{2} \pm \alpha \frac{r^+ - r^-}{2}$$

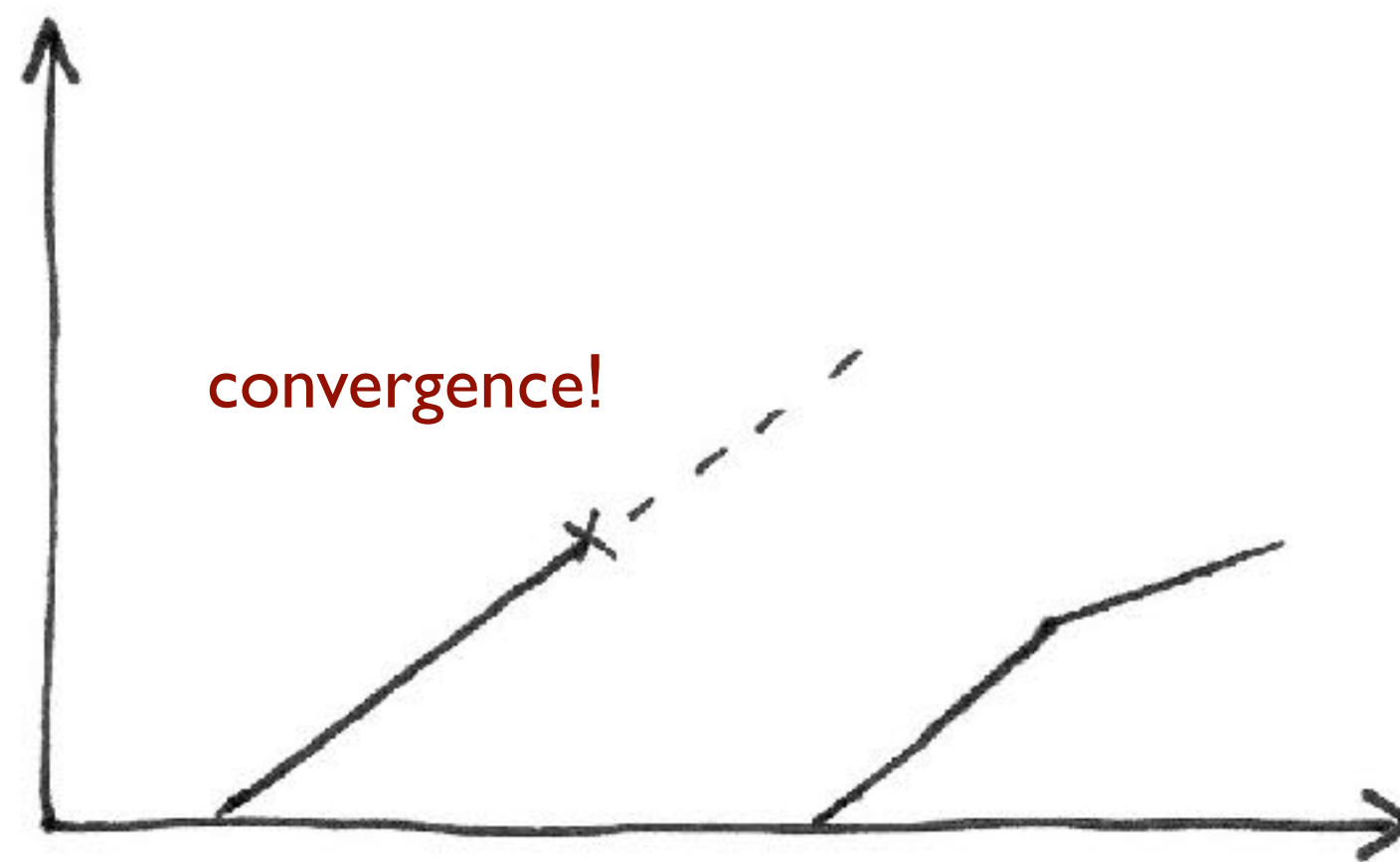


in-the-middle  
algorithm



in-the-middle  
heuristic

# Sweep mechanism



best results with as low disturbance as possible

(also small random costs for resolving ties)

the max-sum problem

# The max-sum problem

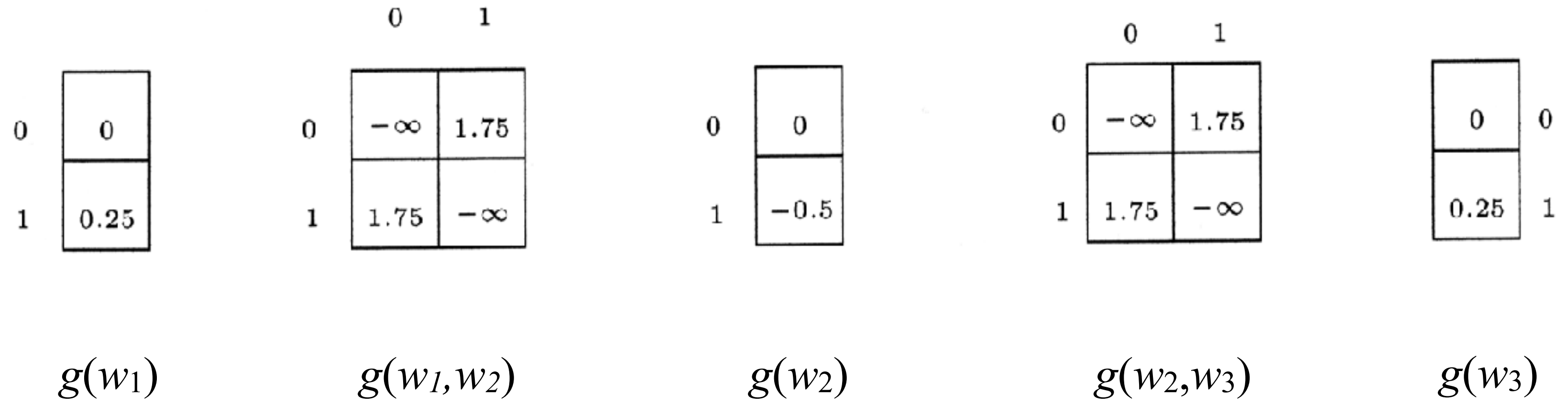
vector of discrete variables

different subsets of  
the variables

$$\max_w f(w) = \sum_k g_k(w^k) + C$$

Allows a highly non-linear cost function

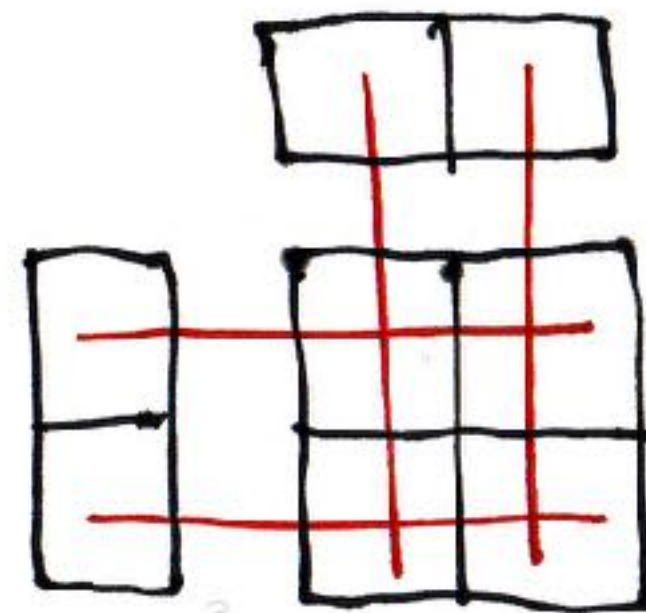
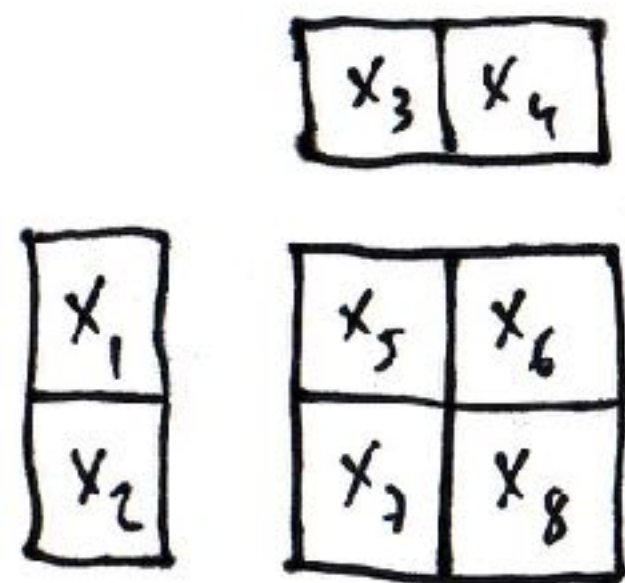
constraint components



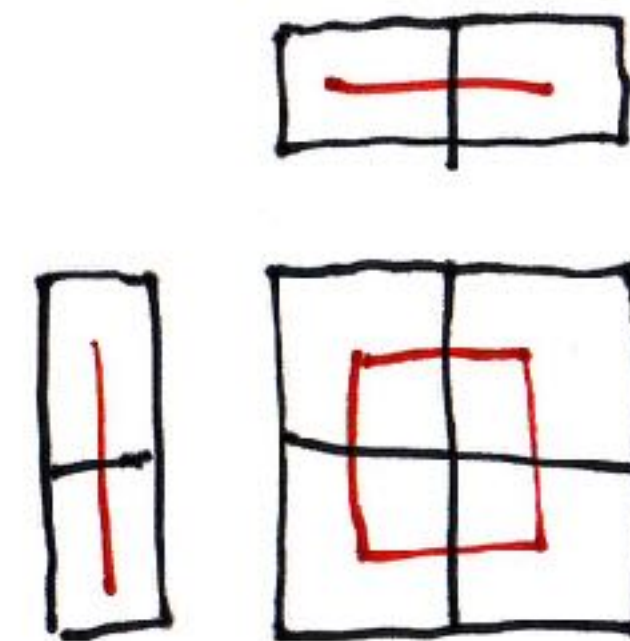
variable components

modelling and solving the max-sum problem as an ILP

# The max-sum ILP



marginalization  
constraints

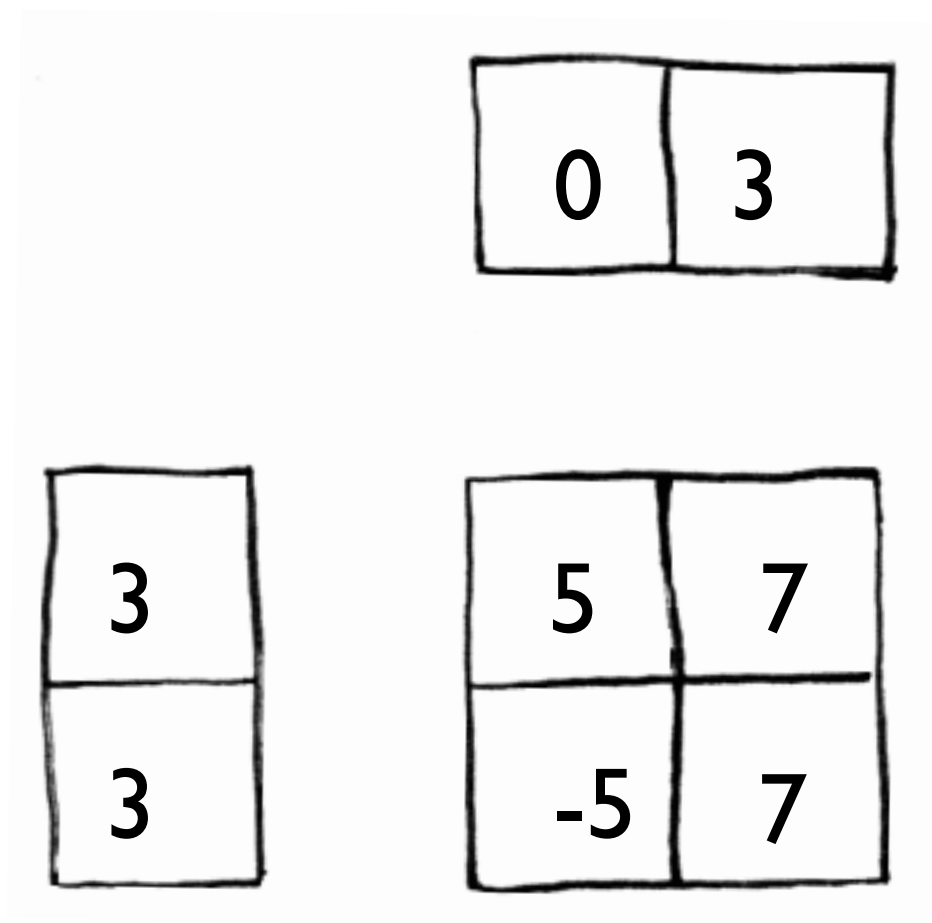


normalization  
constraints

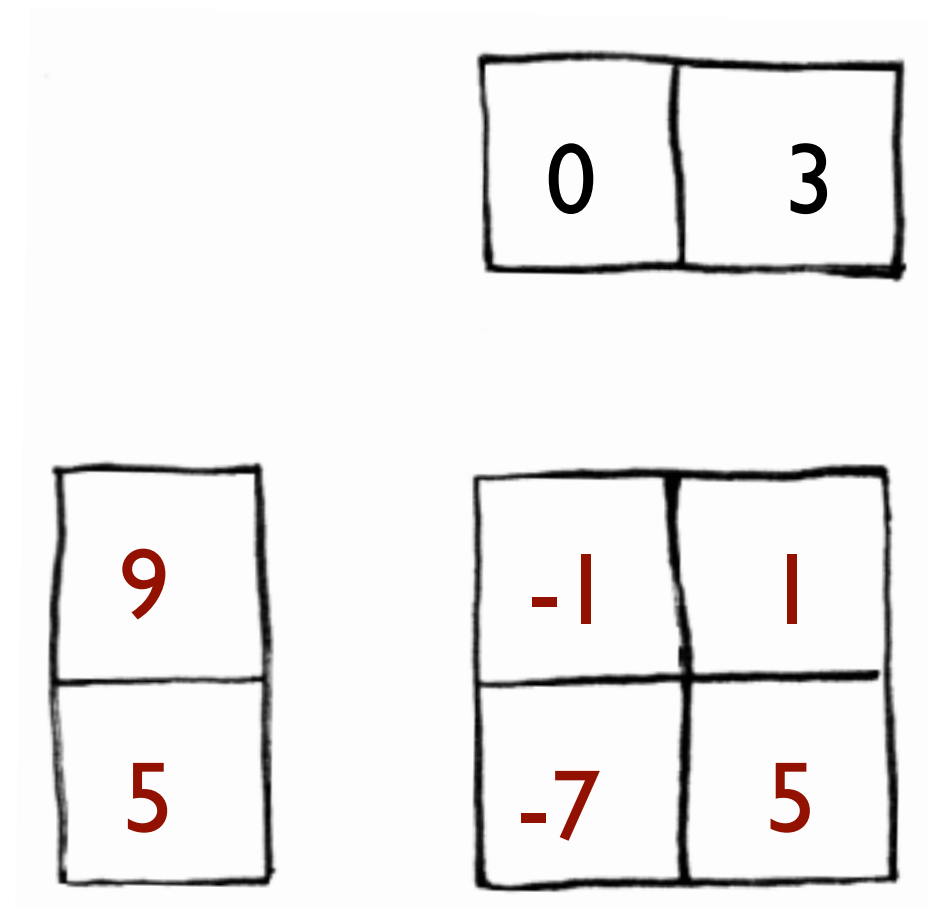
For C, we introduce an extra variable  $x_c$ , together with the constraint  $x_c = 1$ .



# Max-sum with the in-the-middle algorithm



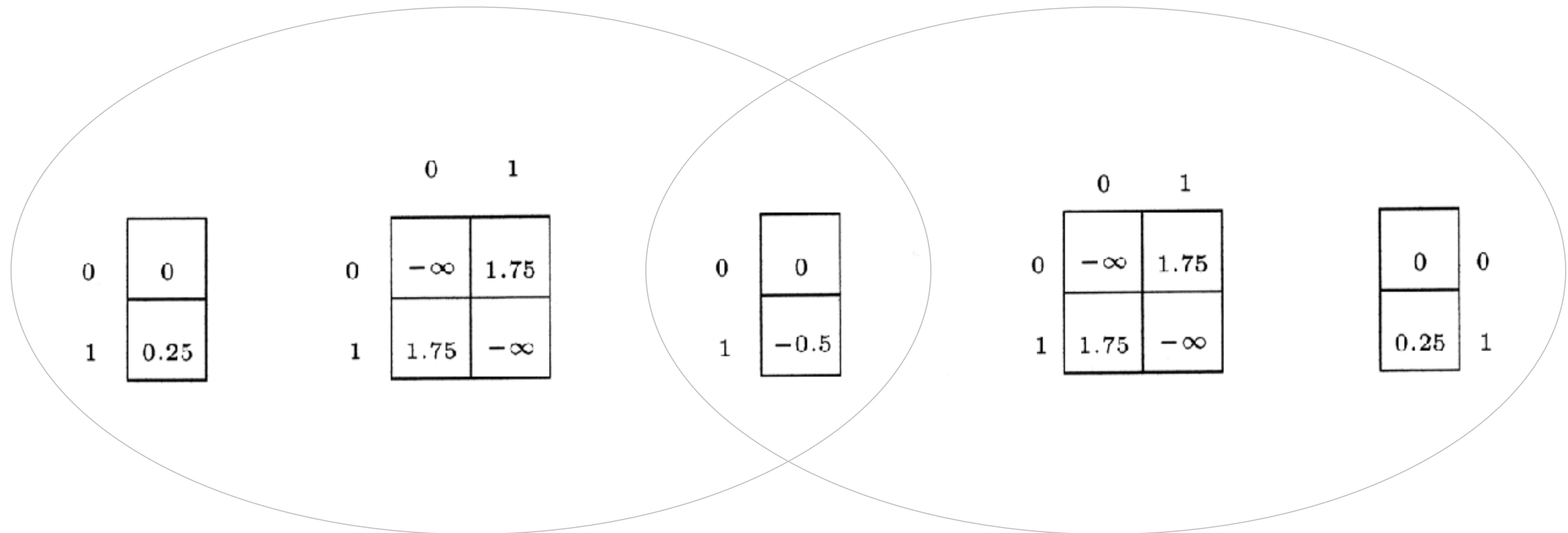
start



in-the-middle

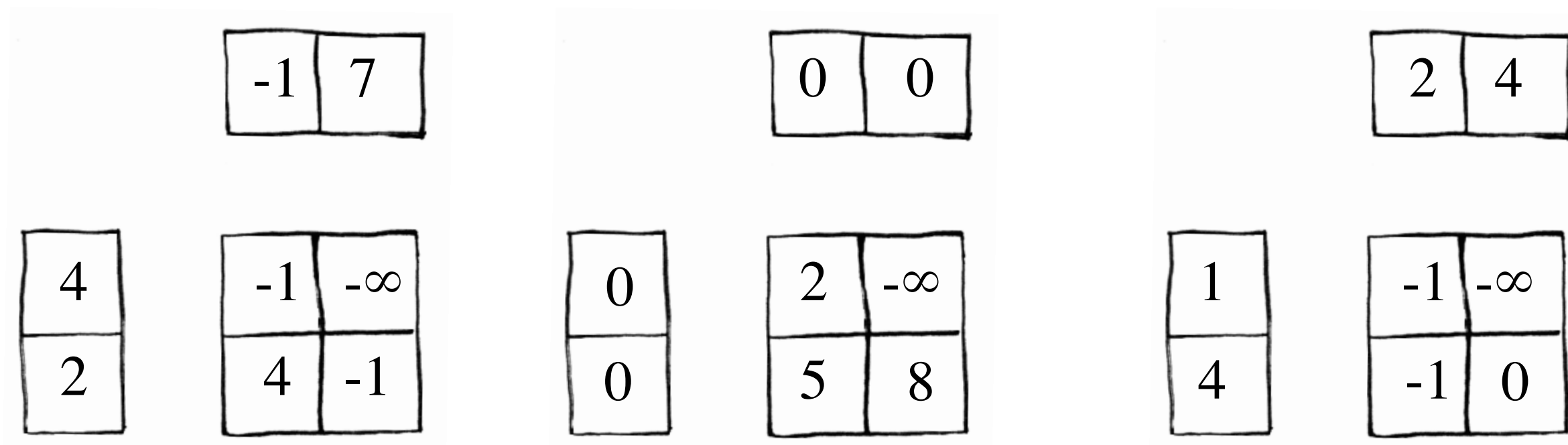
(makes linear constraint feasible)

Or solve entire subproblems with specialised algorithm!



Iteratively update one subproblem at a time

# Move in and move out...



move in variable components

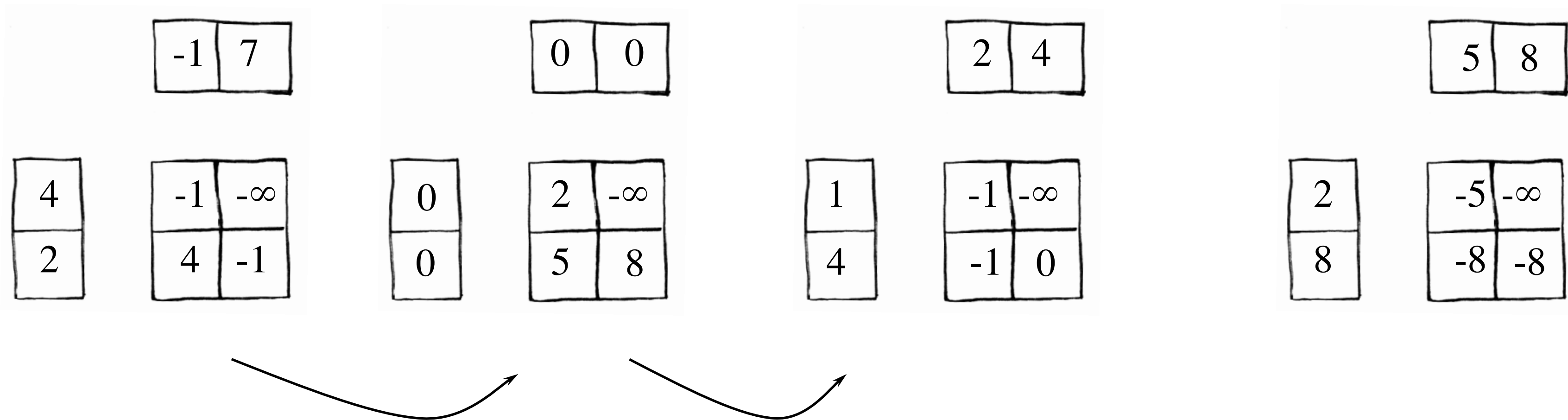
move out, but not too much

We want to keep the best combination still largest in constraint!

“non-conflicting”

minimizes  $f(y)$

What if we move out “too much”?

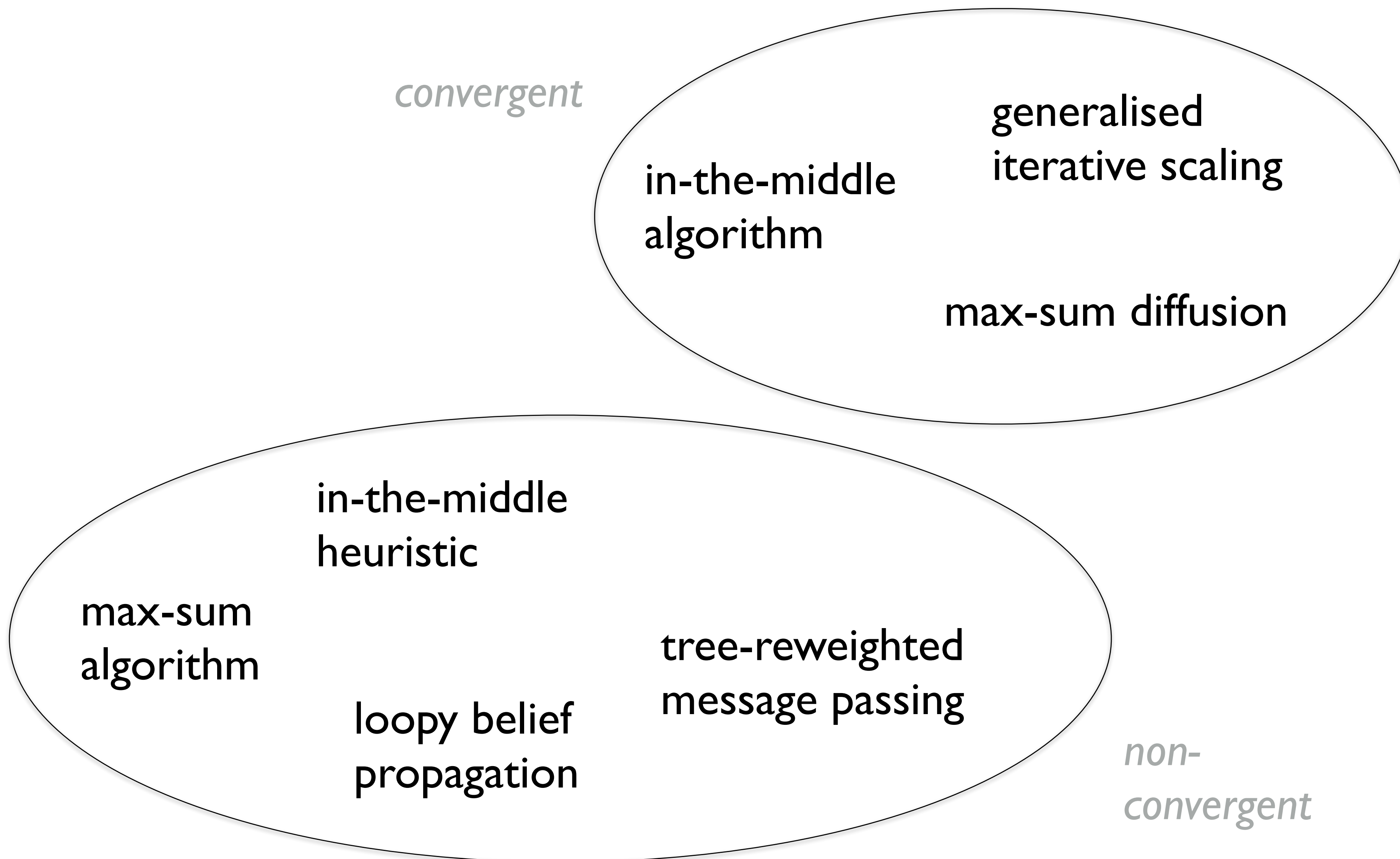


*We can easily explain why the max-sum algorithm does not guarantee optimality!*

here we have moved out even more, we get a “conflict”!

(max-sum algorithm!)

# Many interesting relationships between algorithms!



Can we unify the models?

$$\max_w f(w) = \sum_k g_k(w^k) + C$$

$$\max cx$$

$$Ax = b$$

$$x_j \in \{0, 1\}$$

# Let's model the other way around: ILP to max-sum!

$$\max\{2x_1 + 3x_2 + 2x_3 \mid x_1 + x_2 = 1, x_2 + x_3 = 1, x_j \text{ binary}\}.$$

$$\max g_1(x_1) + g_2(x_2) + g_3(x_3) + g_4(x_1, x_2) + g_5(x_2, x_3).$$

$g_1(x_1)$

|   |   |
|---|---|
| 0 | 0 |
| 1 | 2 |

$g_4(x_1, x_2)$

|   |           |           |
|---|-----------|-----------|
|   | 0         | 1         |
| 0 | $-\infty$ | 0         |
| 1 | 0         | $-\infty$ |

$g_2(x_2)$

|   |   |
|---|---|
| 0 | 0 |
| 1 | 3 |

$g_5(x_2, x_3)$

|   |           |           |
|---|-----------|-----------|
|   | 0         | 1         |
| 0 | $-\infty$ | 0         |
| 1 | 0         | $-\infty$ |

$g_3(x_3)$

|   |   |
|---|---|
|   | 0 |
|   | 1 |
| 0 | 0 |
| 1 | 2 |

$$\max_w f(w) = \sum_k g_k(w^k) + C$$

$$\max cx$$

$$Ax = b$$

$$x_j \in \{0, 1\}$$

if we move out as  
much as possible  
but not too much...

⇒

same as in-the-  
middle algorithm  
for the original ILP

max-sum  
algorithm

⇒

same as in-the-  
middle heuristic  
(for  $\alpha=1$ )

*The distinction between an ILP model  
and a max-sum model is blurred!*

*The in-the-middle updates can be seen as fast  
specialized max-sum constraint updates!*



To note from a constraint programming perspective

Numerical propagation can solve  
many non-trivial problems with  
propagation only!

(no combinatorial search!)

# Summary

in-the-middle algorithm  
in-the-middle heuristic

unify  
algorithms!

unify  
models!

END