Exercise 1:
Intelligent Systems in Medicine

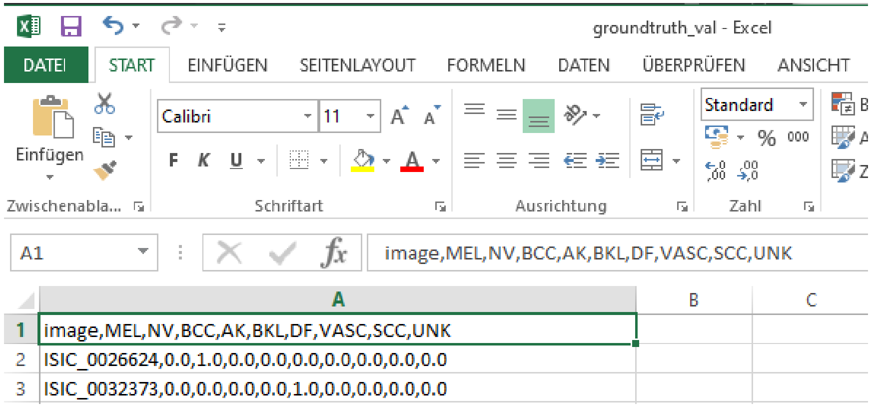
- Project Introduction
- Search & Optimization
  - A* search
  - Constrained optimization

Data Set

- Link to dataset (make sure to use the data from 2019):

  https://challenge.isic-archive.com/data

- 22,798 training images with labels and 2,535 validation images in Stud.IP (split generated from training data)
- Why do we need validation data?
- Two phases to build and evaluate classifier
- Tasks: Multiclass classification
- There is an unknown class!



Submission System

- Link to online submission system:

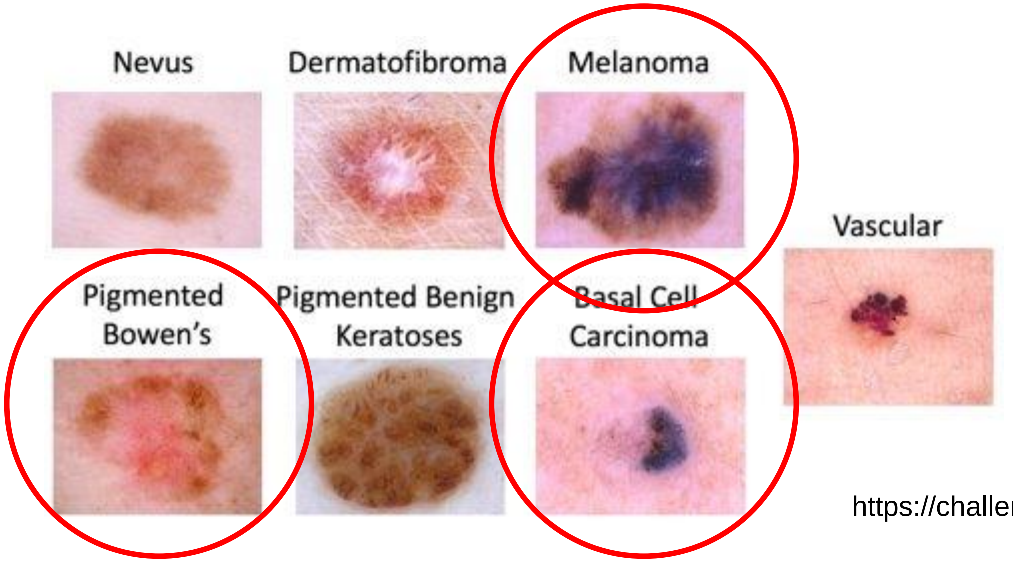
  https://cgi.tu-harburg.de/~c00e1fn1/

- Groups of 4 students are already registered
- Upload validation results to compare your performance with other groups and upload final test submission for leader board
- Introduction to different metrics in the next exercise (2.12.20)
- ISIC evaluation:

  https://challenge2019.isic-archive.com/evaluation.html

Image Features

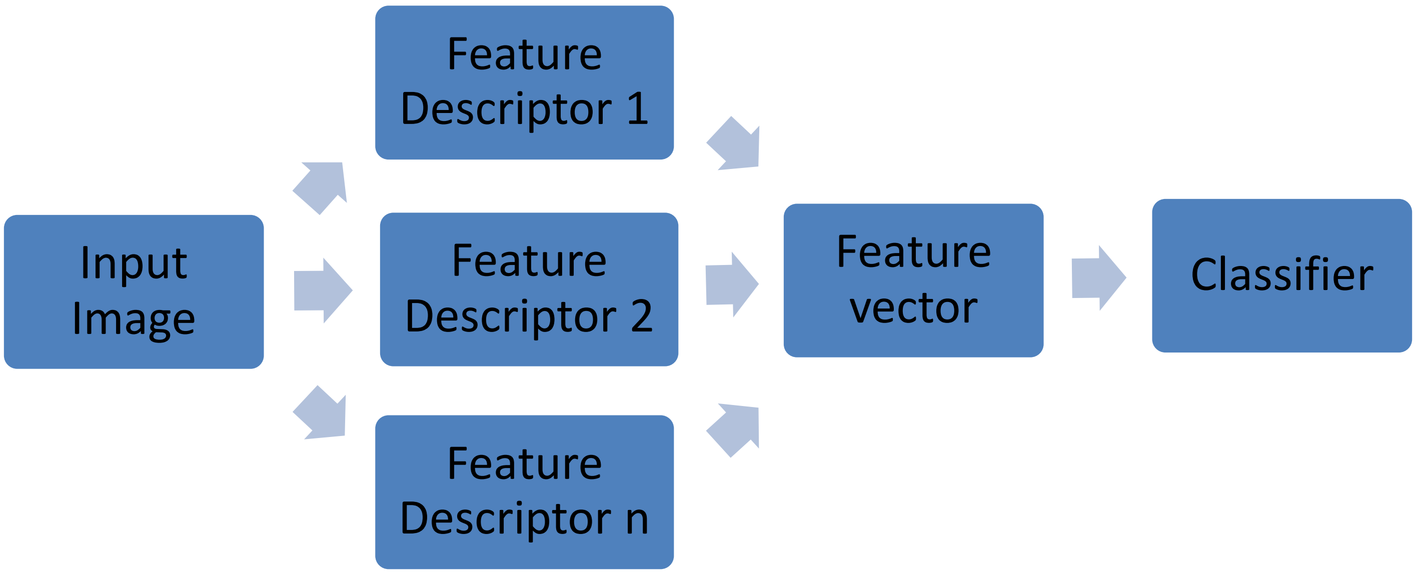
- Goal: Extract information as a list of numbers from images



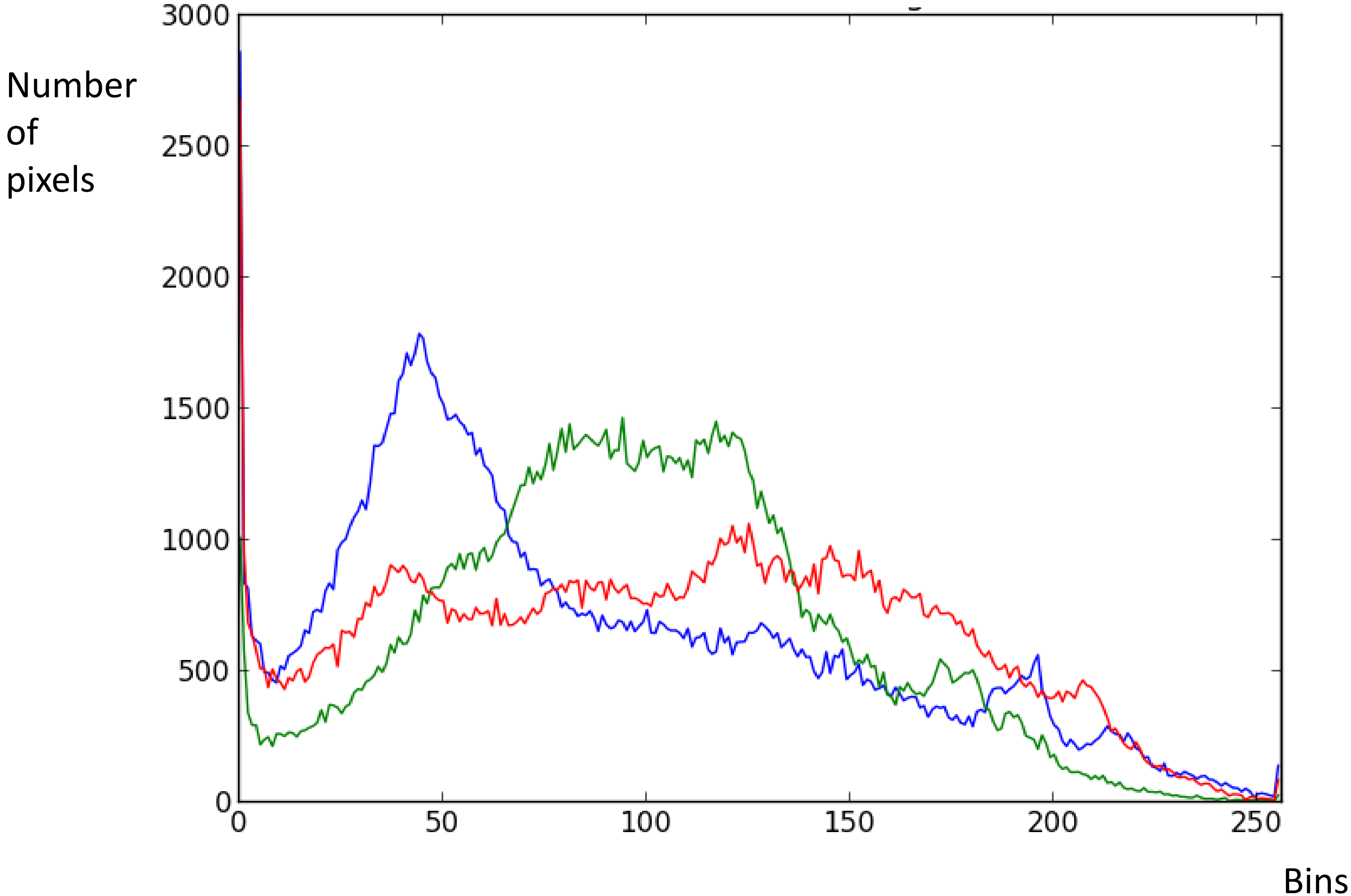
https://challenge2018.isic-archive.com/task3/

- What properties can be used to distiguish benign from malignant?
  - Color
  - Shape
  - Texture

Image Features



Color Histograms



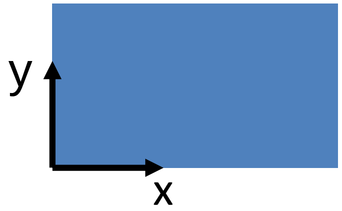
Moments

- General spatial moments:

$$M_{pq} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x^p\, y^q f(x,y)dxdy$$

- Analogy in Mechanics:

$$\bar{x} = \frac{M_{10}}{M_{00}} = \frac{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x^1\, \rho(x,y)dxdy}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \rho(x,y)dxdy}$$



- Higher order moments capture distribution of „mass“

Hue Moments

- Central moments:

$$\mu_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - \bar{x})^p \, (y - \bar{y})^q f(x, y) dx dy$$

- For digital images:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q$$

- Seven Hue moments [1] are calculated from combinations of central moments

e.g. $(\mu_{20} - \mu_{02})^2 + {\mu_{11}}^2$

[1] M. K. Hu, "Visual Pattern Recognition by Moment Invariants", IRE Trans. Info. Theory, vol. IT-8, pp.179–187, 1962

Hue Moments (2)

- Hue moments are invaraint to:
  - Image scale
  - Translations
  - Rotations
  - Reflections (partially)

Other feature descriptors

- Texture, e.g. Haarlick
- Histogram of Oriented Gradients (HOG)
- Make sure different features are on the same scale, normalize!
- Search for more feature descriptors and try out different combinations

How to get started?

- Literature research: read papers
- Load data and labels (how deal with unknown class)
- Image preprocessing: improve images?, colour charts, gel bubbles, scaling, normalization, …
- Classification methods
- Evaluation

Python

- Libraries (install with pip3, anaconda,…)
- Load images: matplotlib, PIL, numpy
- Read labels: csv, pandas
- Image processing: skimage, numpy, scipy, opencv

Organisation

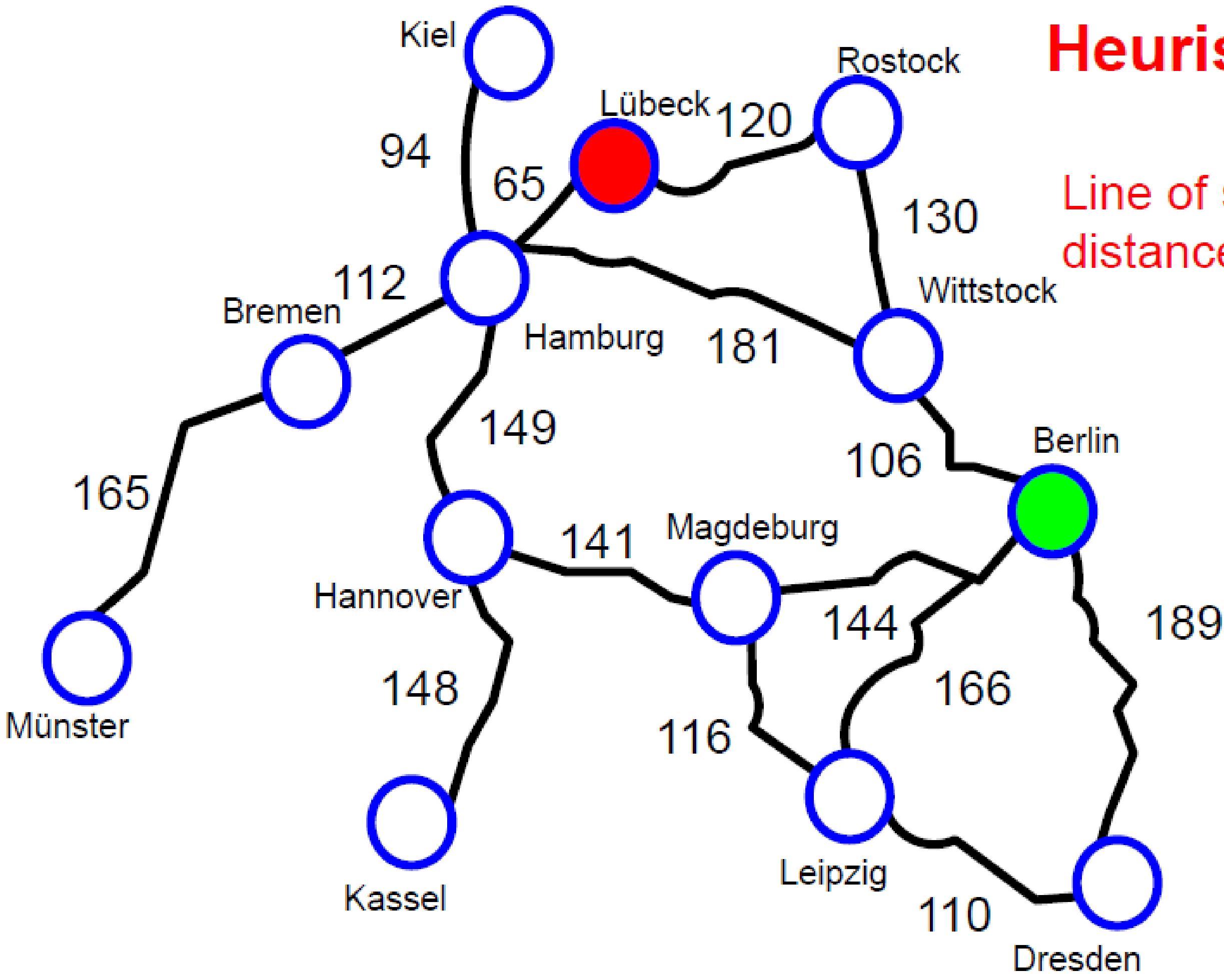
- Check ISM schedule for overview:
- 25.11.2020: Deadline for submitting your time schedule
- Send time schedule to johanna.sprenger@tuhh.de and state if you want to discuss your schedule in a zoom sessions with or if you want feedback via email

Groups

Breakout-Sessions

Exercise

Simple Graph Search



Heuristic

Line of sight distance

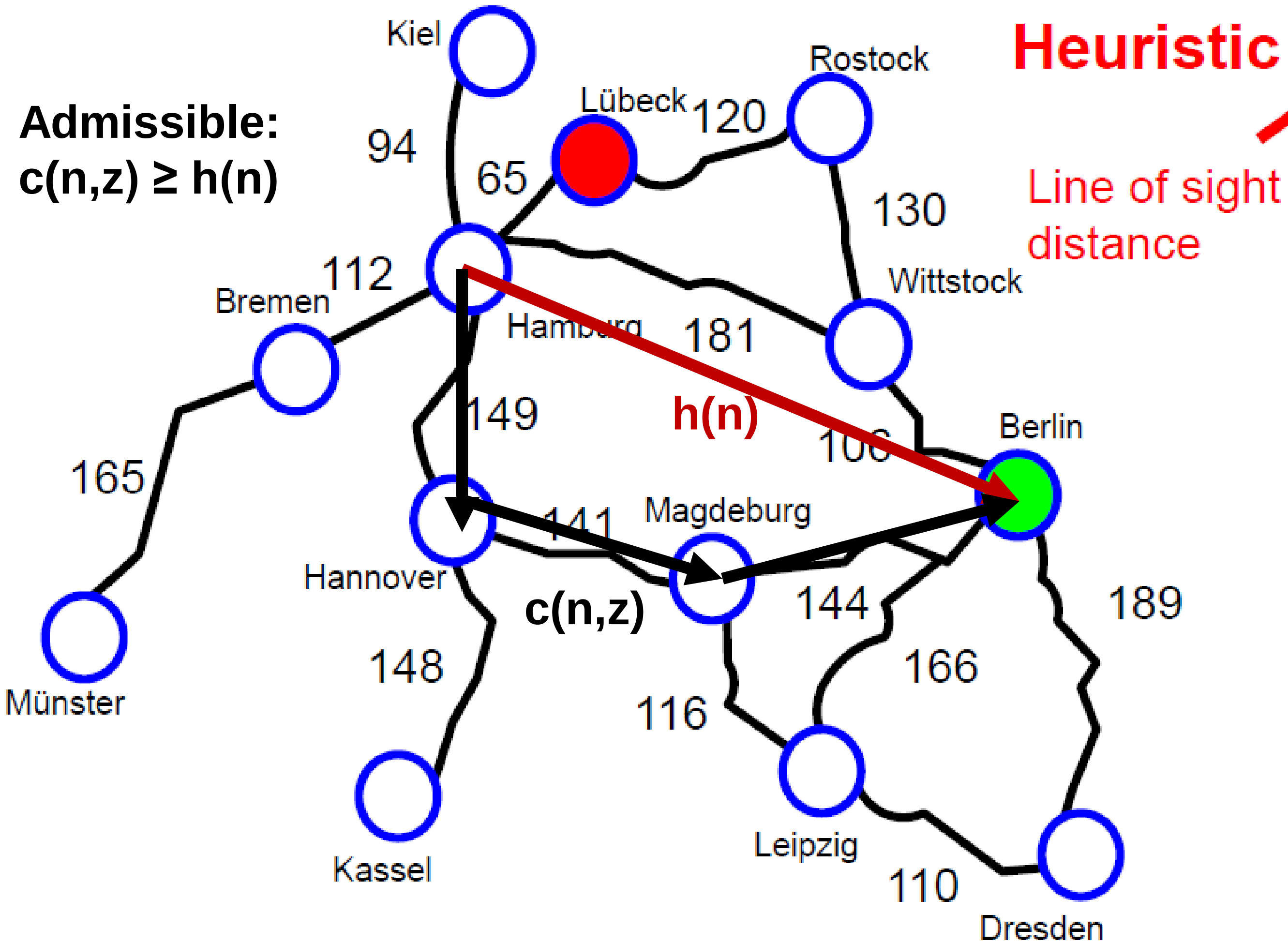
	$d_{LL}(\cdot, \mathrm{B})$
B	0
DD	165
H	250
HB	315
HH	254
HL	234
HRO	193
KI	296
KS	298
L	146
MD	128
MS	397
WK	93

Simple Graph Search

- A* search: combine actual cost and heuristic
- **Admissible**: **$c(n,z) \geq h(n)$,** i.e., any actual path from n to goal z is longer or equal to the heuristic function value for n
- **Consistent: $h(n) \leq c(n,n') + h(n')$**, i.e., for any successor n‘ of n the cost of getting to n‘ and the estimate from n‘ are in total larger then the estimate from n

Simple Graph Search

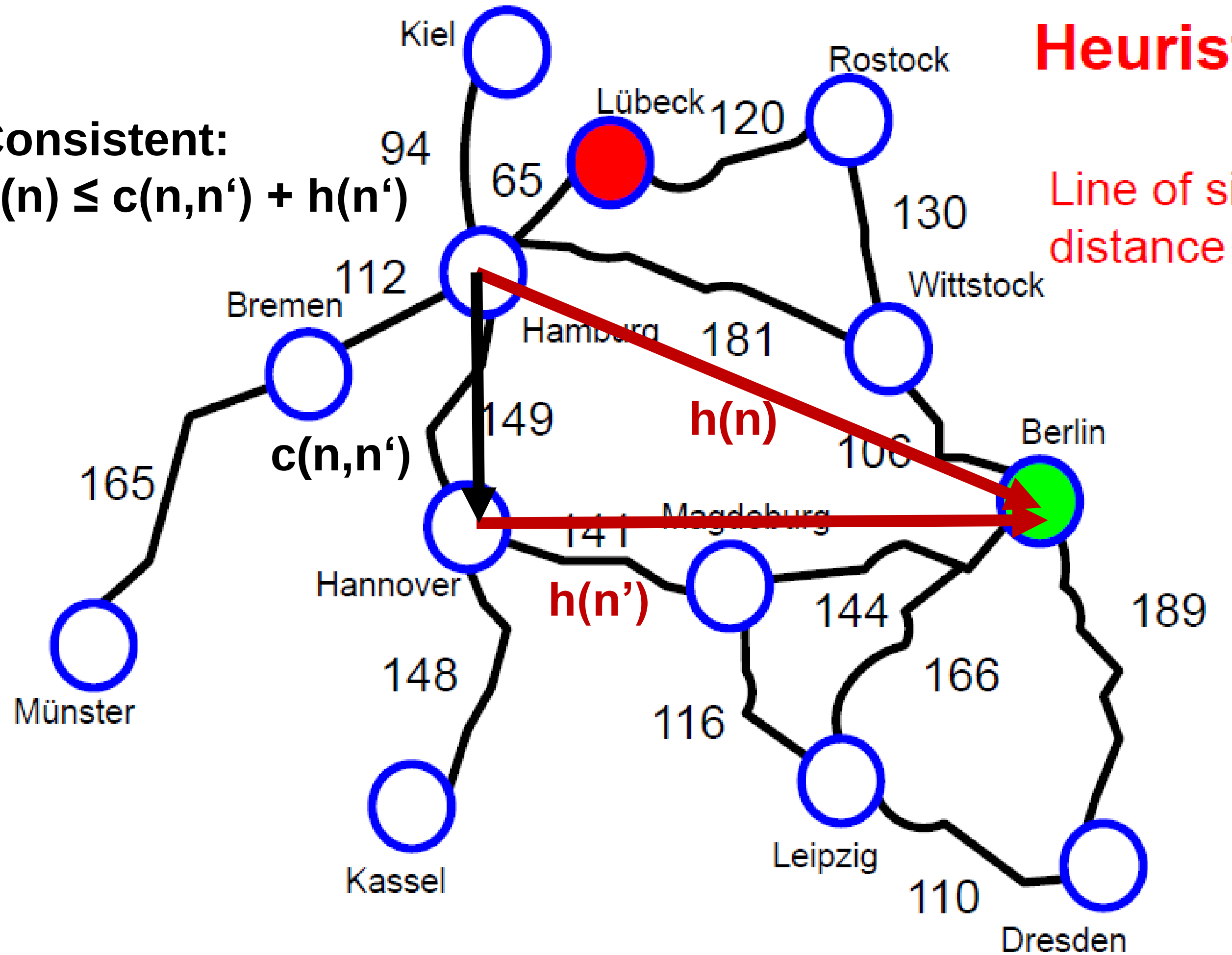
Admissible:
c(n,z) ≥ h(n)



	$d_{LL}(\cdot, B)$
B	0
DD	165
H	250
HB	315
HH	254
HL	234
HRO	193
KI	296
KS	298
L	146
MD	128
MS	397
WK	93

Simple Graph Search

Consistent:
h(n) ≤ c(n,n') + h(n')



Heuristic

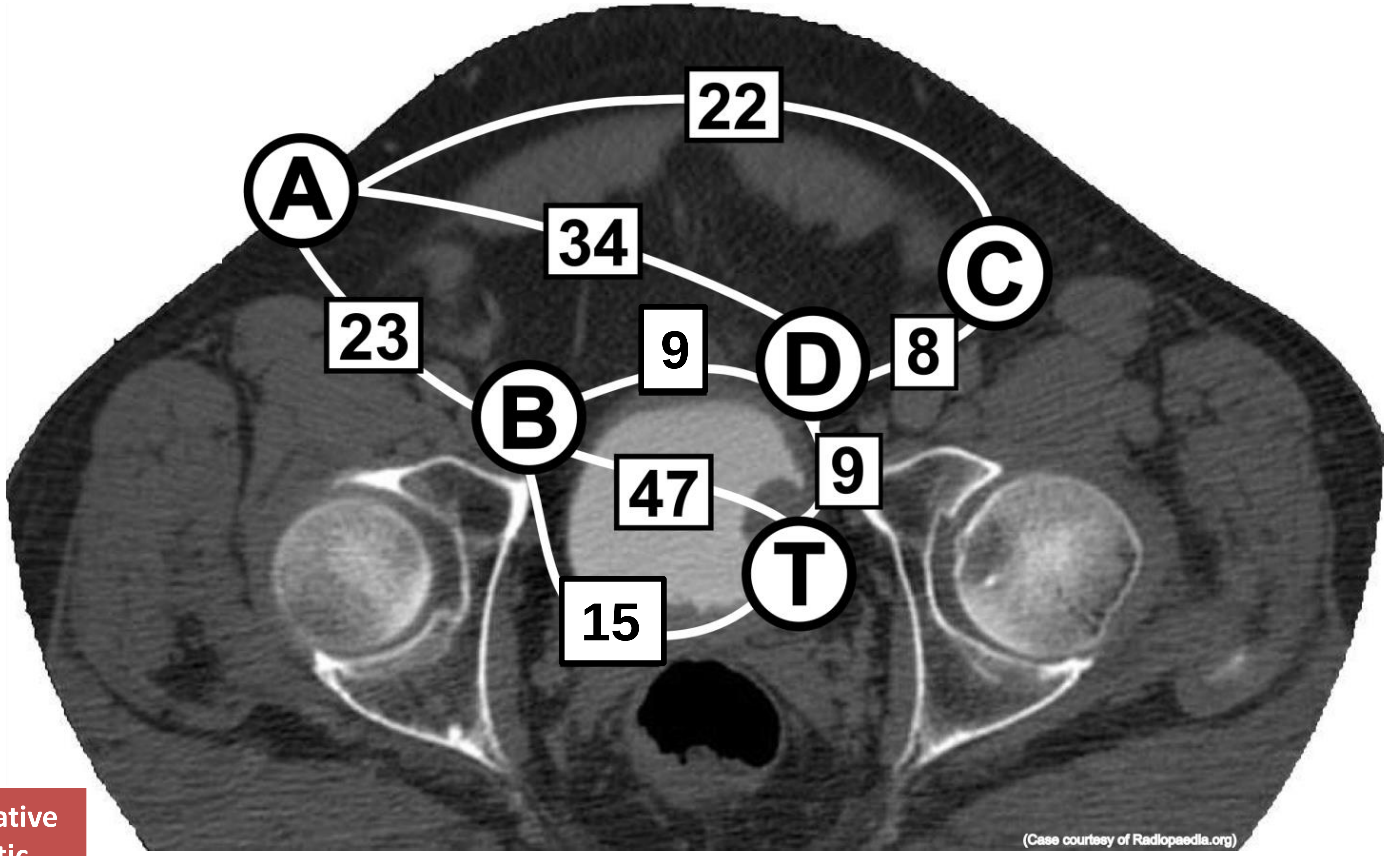
Line of sight distance

	$d_{LL}(\cdot, B)$
B	0
DD	165
H	250
HB	315
HH	254
HL	234
HRO	193
KI	296
KS	298
L	146
MD	128
MS	397
WK	93

Task 1: A*-Search

- Can you use the proposed heuristic or do you need to use the alternative heuristic?
- Perform an A*-Search from **A** to **T**!

Simple Graph Search

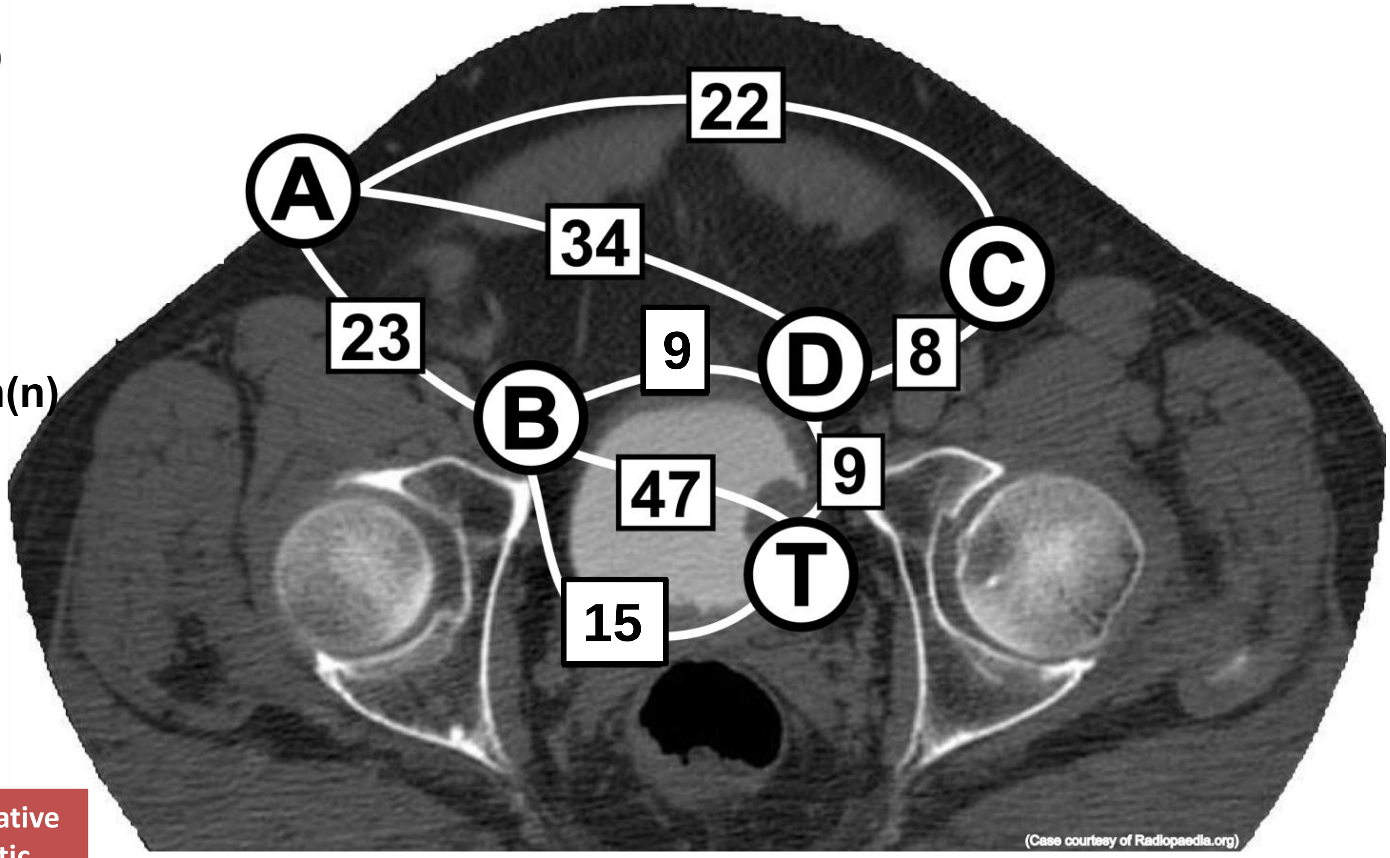


node	proposed heuristic	alternative heuristic
A	33	29
B	15	14
C	16	7
D	7	6
T	0	0

Simple Graph Search

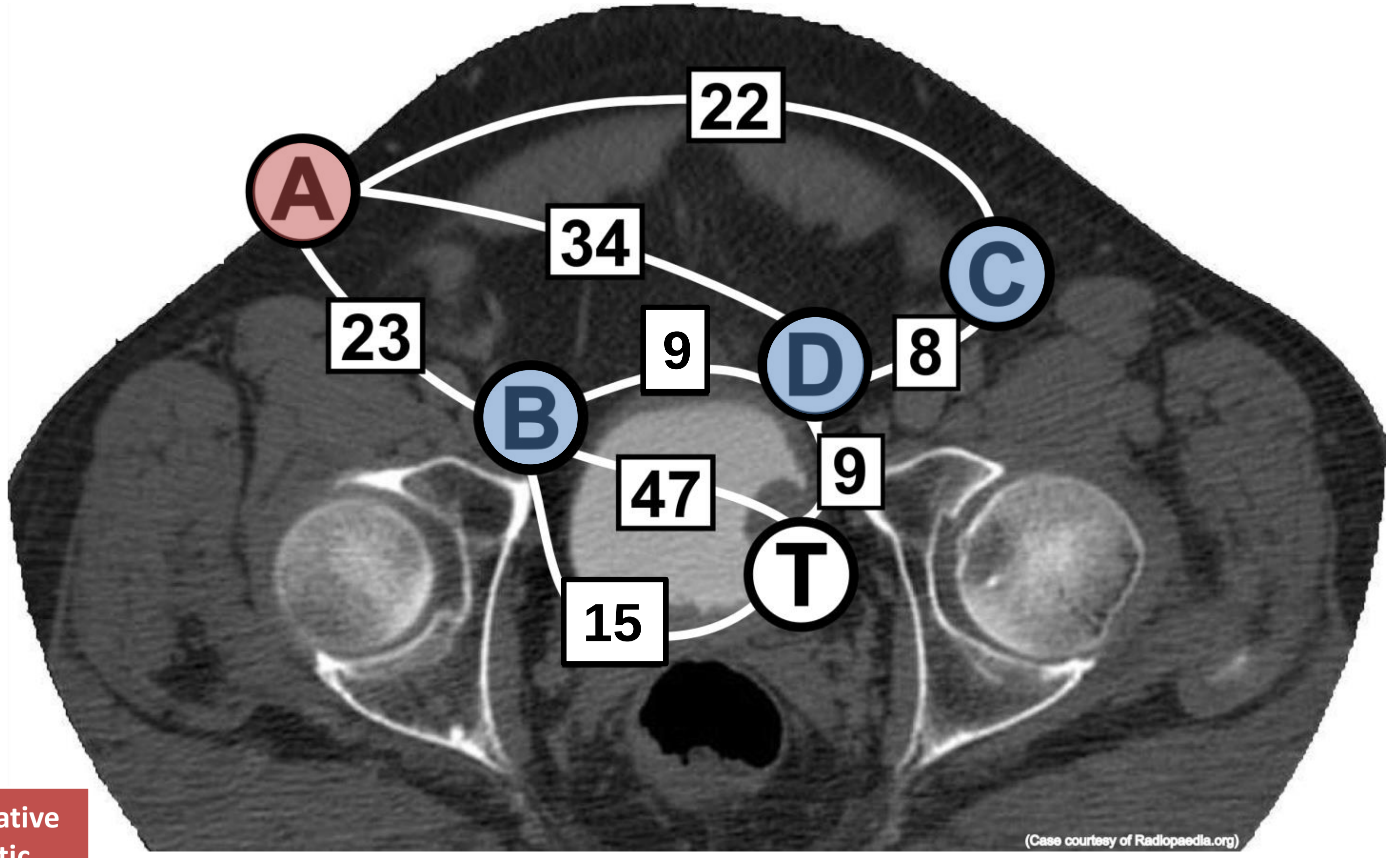
Consistent:
$h(n) \leq c(n,n') + h(n')$

admissible but not consistent:
$h(C) = 16$
$c(C,D) = 8$
$h(D) = 7$
→ $c(n,n') + h(n') < h(n)$



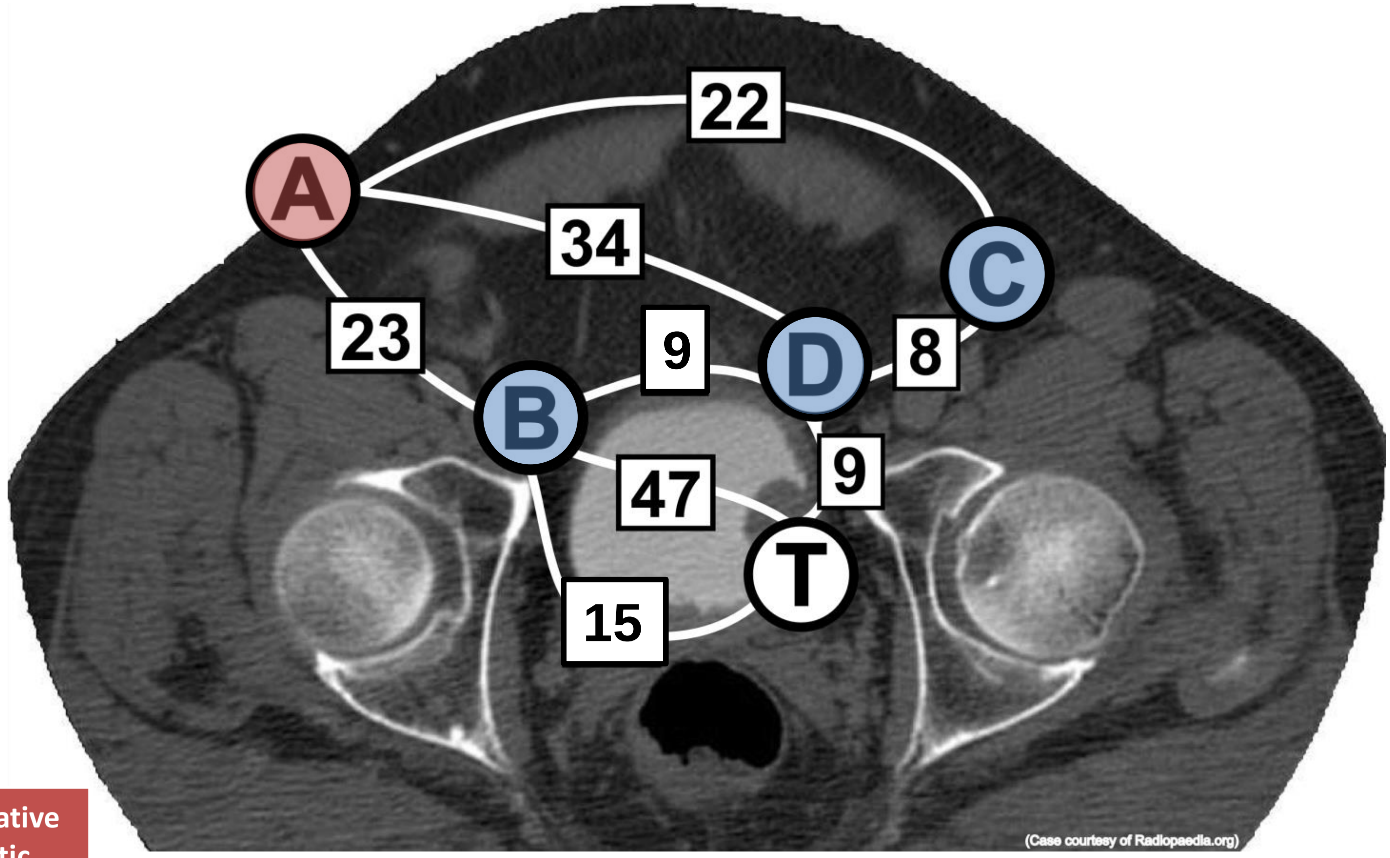
node	proposed heuristic	alternative heuristic
A	33	**29**
B	15	**14**
C	16	**7**
D	7	**6**
T	0	**0**

Simple Graph Search



node	proposed heuristic	alternative heuristic
A	33	**29**
B	15	**14**
C	16	**7**
D	7	**6**
T	0	**0**

Simple Graph Search

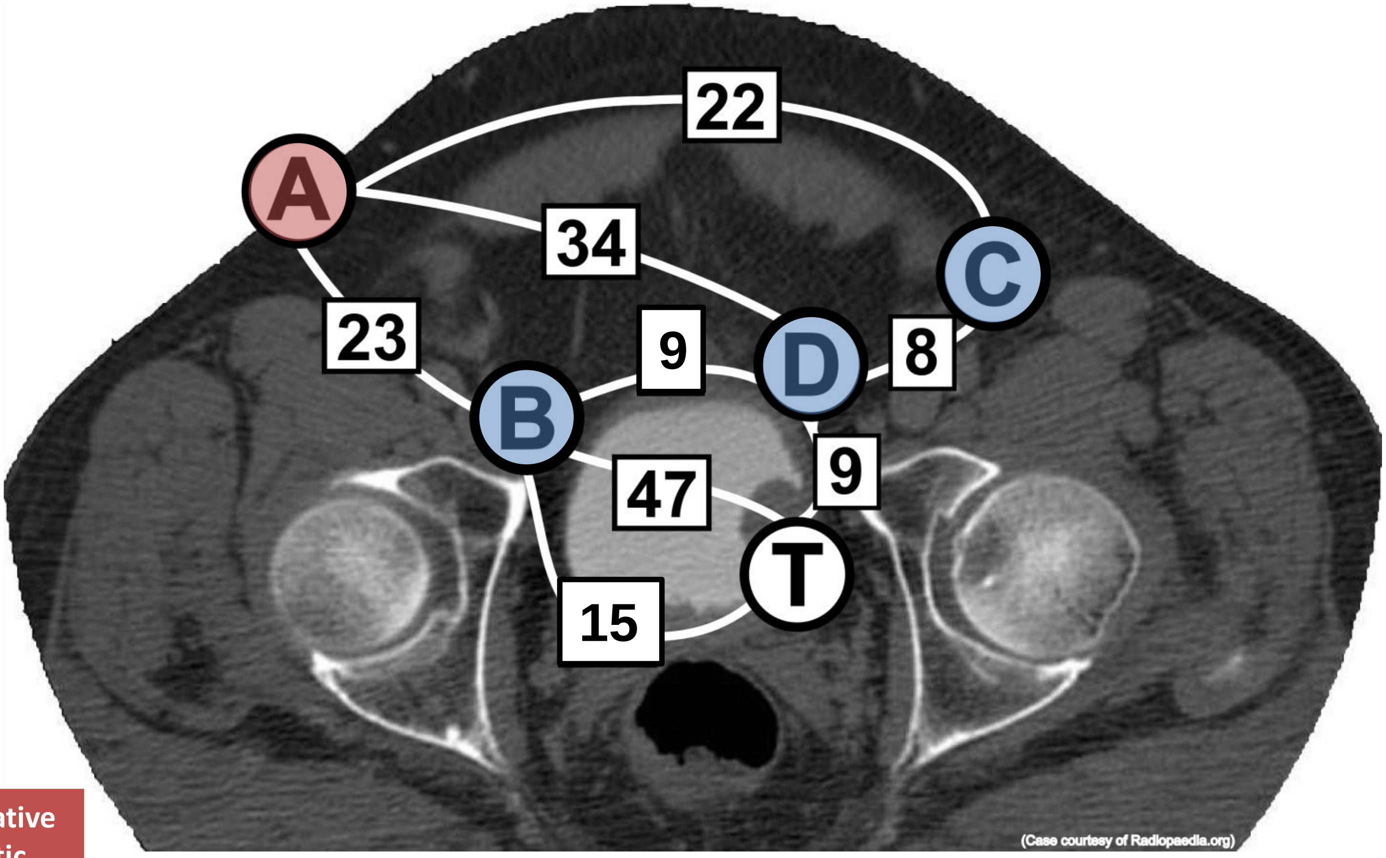


node	proposed heuristic	alternative heuristic
A	33	**29**
B	15	**14**
C	16	**7**
D	7	**6**
T	0	**0**

Simple Graph Search

Open list:
C(29,A)
D(40,A)
B(37,A)

Closed list:
A(29,-)

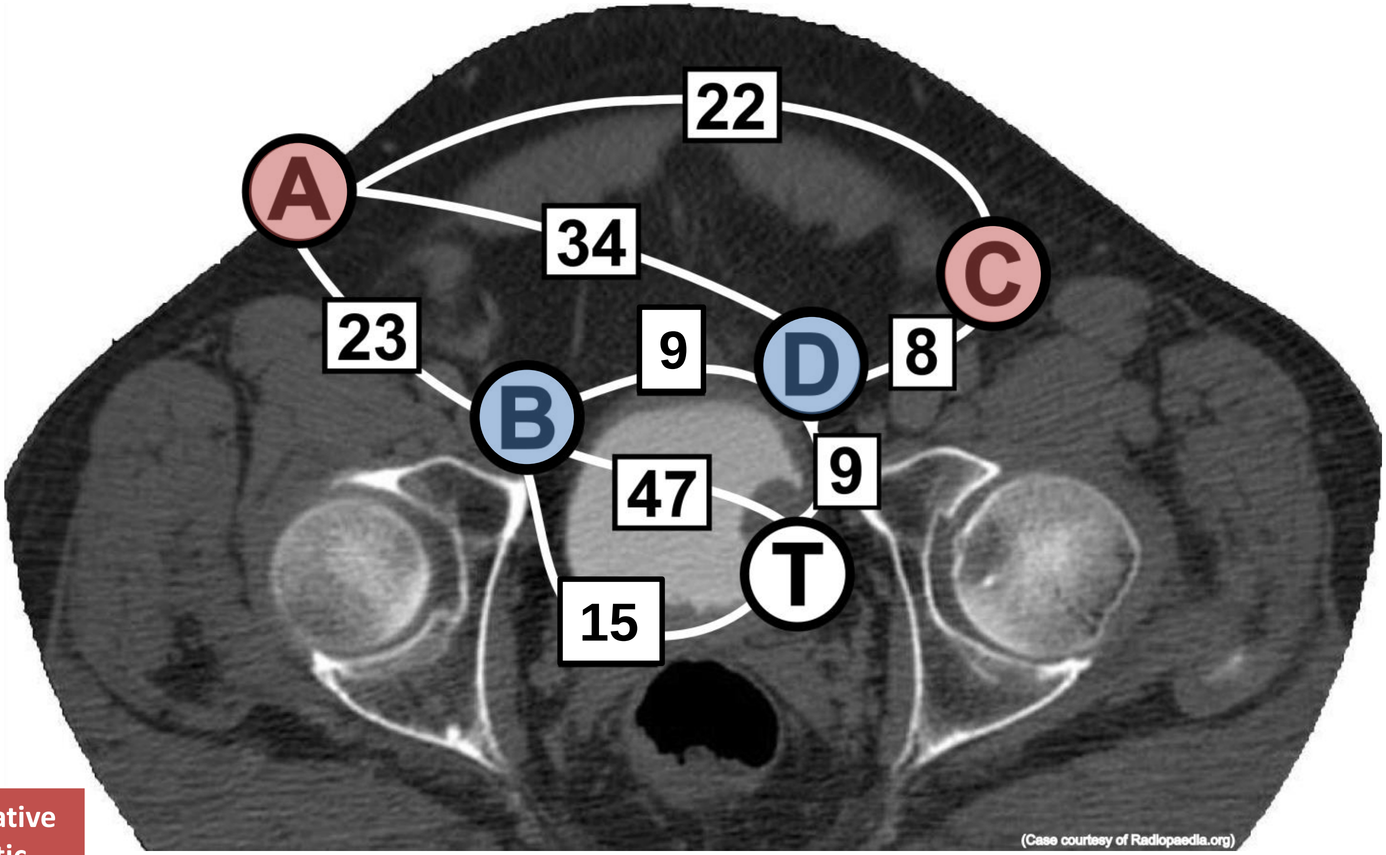


node	proposed heuristic	alternative heuristic
A	33	**29**
B	15	**14**
C	16	**7**
D	7	**6**
T	0	**0**

Simple Graph Search

Open list:
C(29,A)
D(40,A)
B(37,A)

Closed list:
A(29,-)

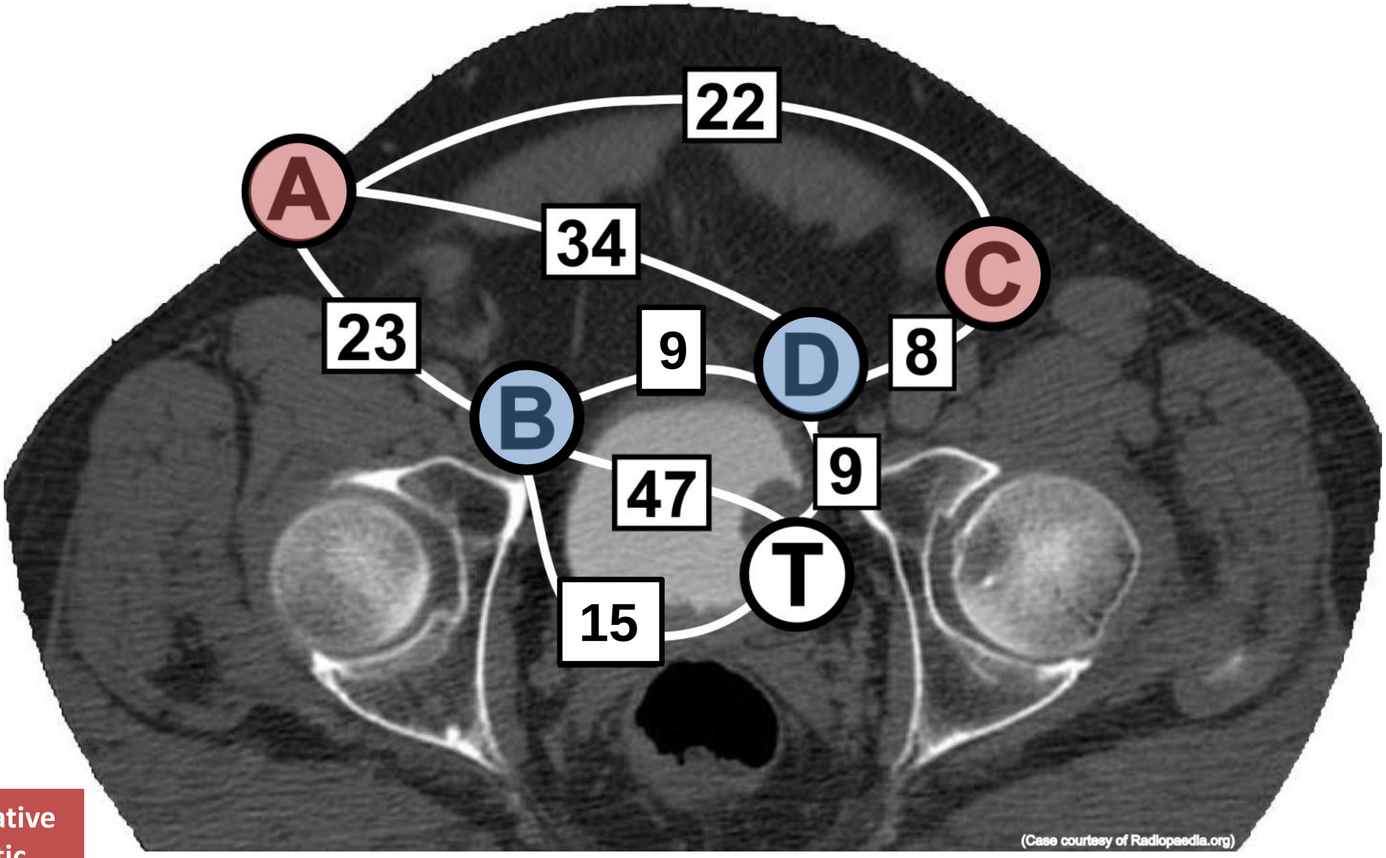


node	proposed heuristic	alternative heuristic
A	33	**29**
B	15	**14**
C	16	**7**
D	7	**6**
T	0	**0**

Simple Graph Search

Open list:
D(36,C)
B(37,A)

Closed list:
A(29,-)
C(29,A)

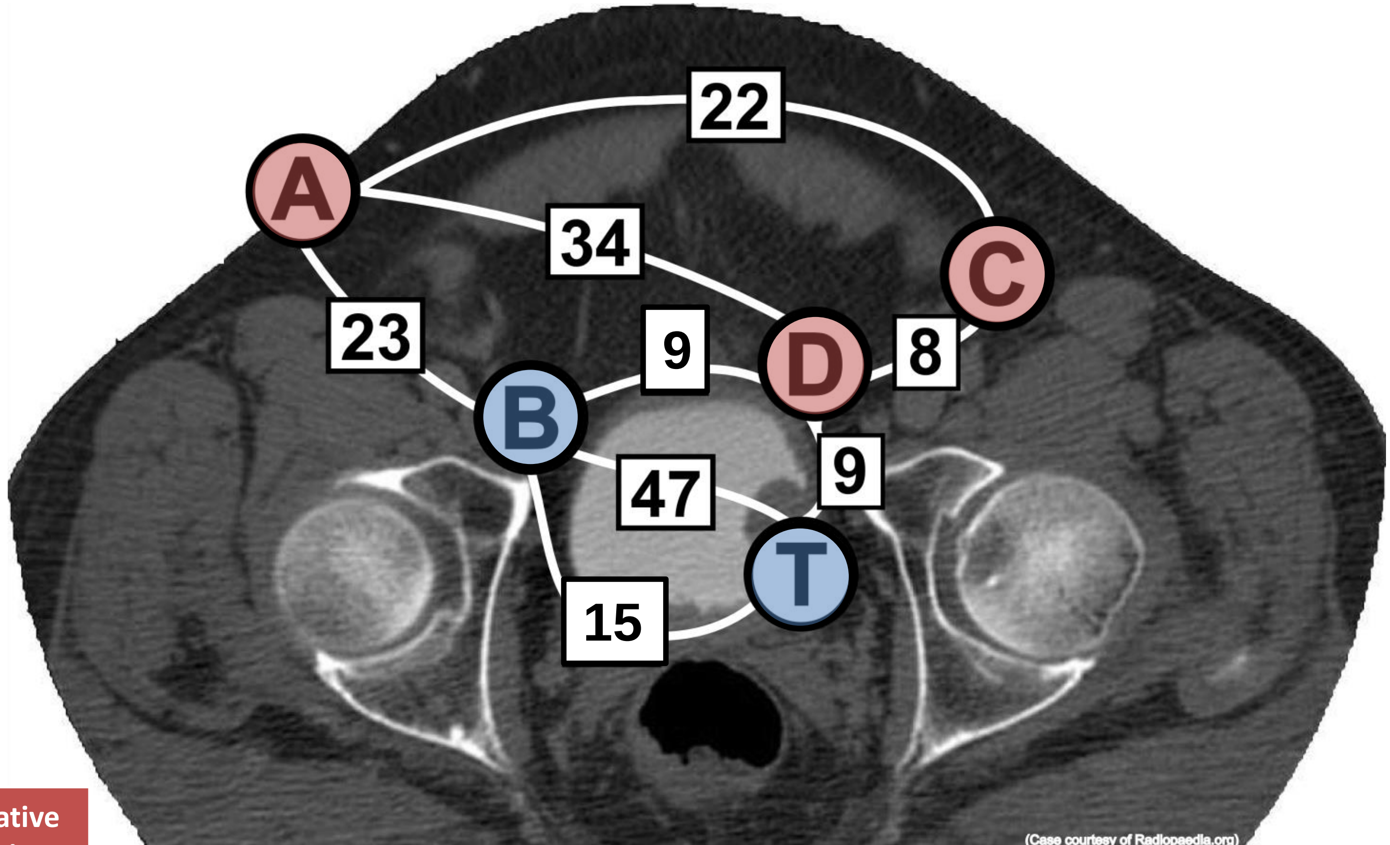


node	proposed heuristic	alternative heuristic
A	33	**29**
B	15	**14**
C	16	**7**
D	7	**6**
T	0	**0**

Simple Graph Search

Open list:
D(36,C)
B(37,A)

Closed list:
A(29,-)
C(29,A)

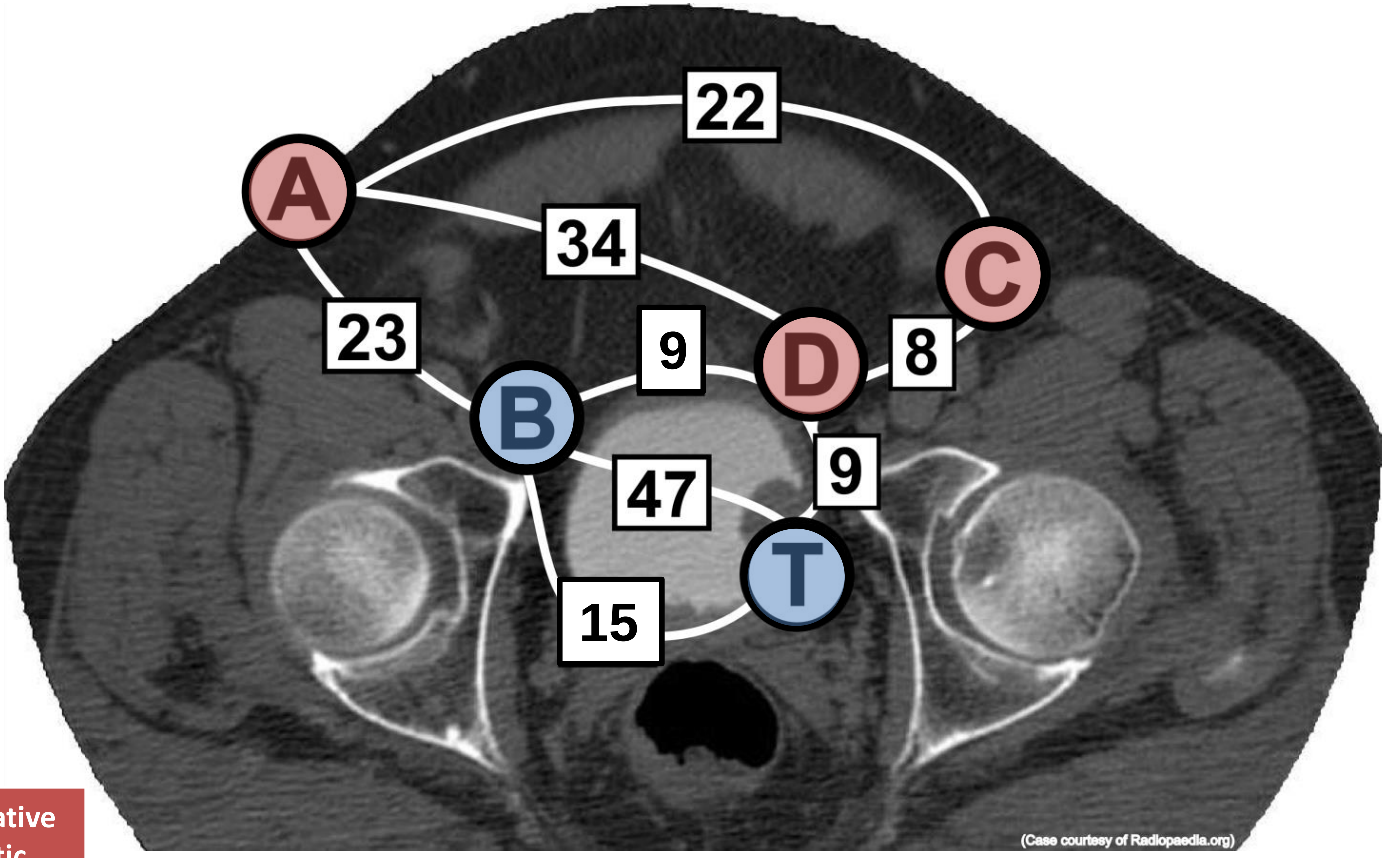


node	proposed heuristic	alternative heuristic
A	33	**29**
B	15	**14**
C	16	**7**
D	7	**6**
T	0	**0**

Simple Graph Search

Open list:
B(37,A)
T(39,D)

Closed list:
A(29,-)
C(29,A)
D(36,C)

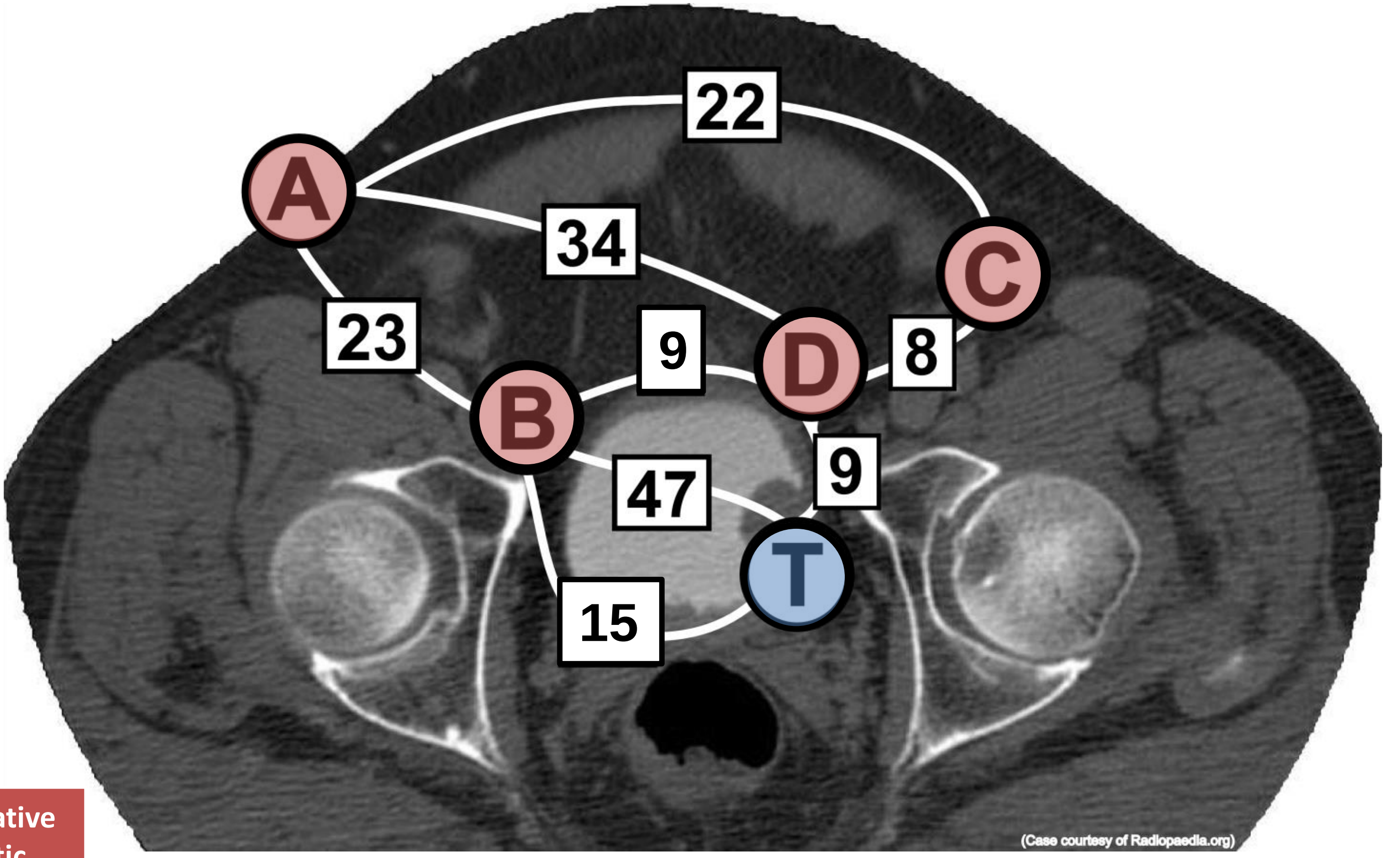


node	proposed heuristic	alternative heuristic
A	33	**29**
B	15	**14**
C	16	**7**
D	7	**6**
T	0	**0**

Simple Graph Search

Open list:
B(37,A)
T(39,D)

Closed list:
A(29,-)
C(29,A)
D(36,C)

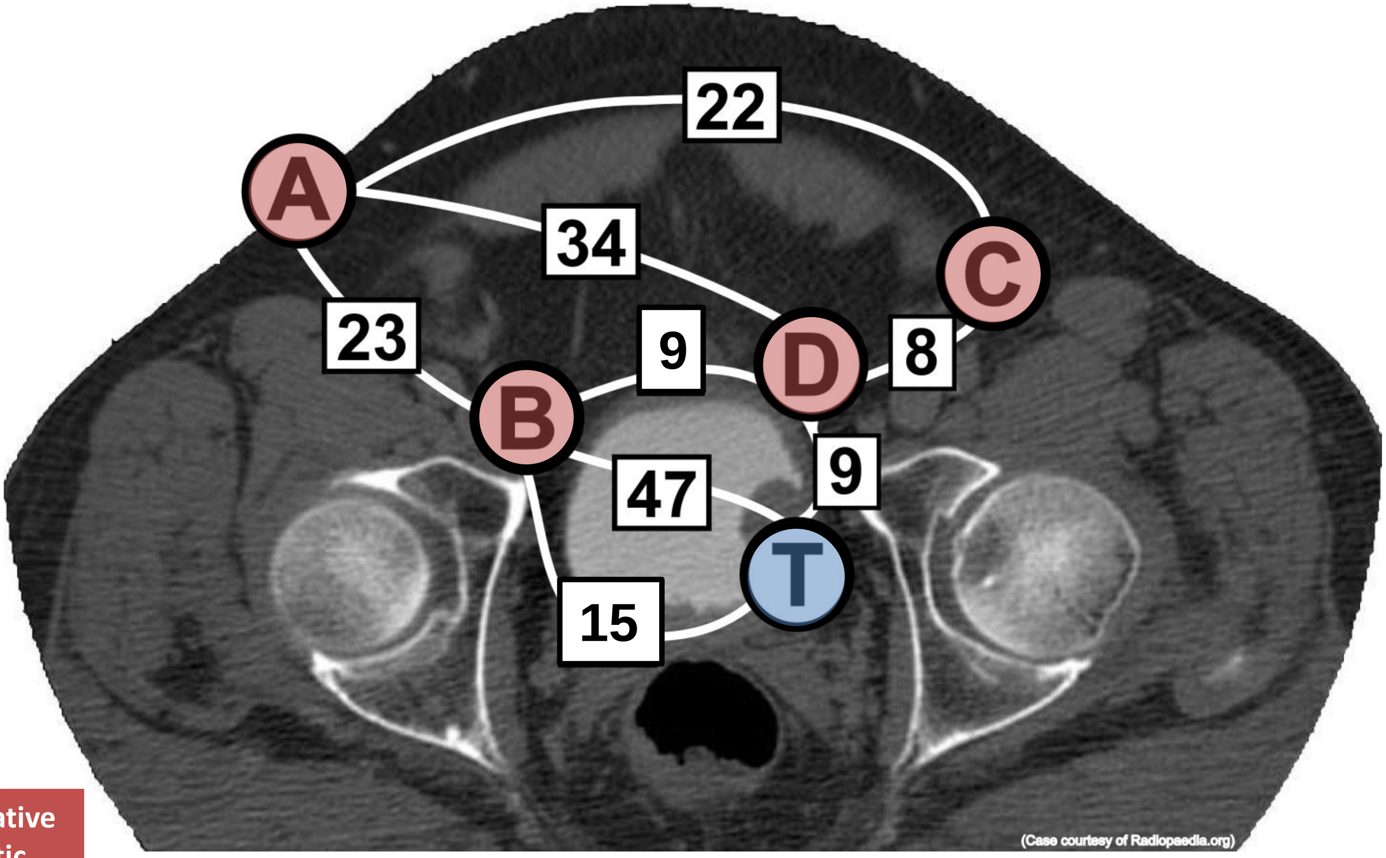


node	proposed heuristic	alternative heuristic
A	33	**29**
B	15	**14**
C	16	**7**
D	7	**6**
T	0	**0**

Simple Graph Search

Open list:
T(38,B)

Closed list:
A(29,-)
C(29,A)
D(36,C)
B(37,A)

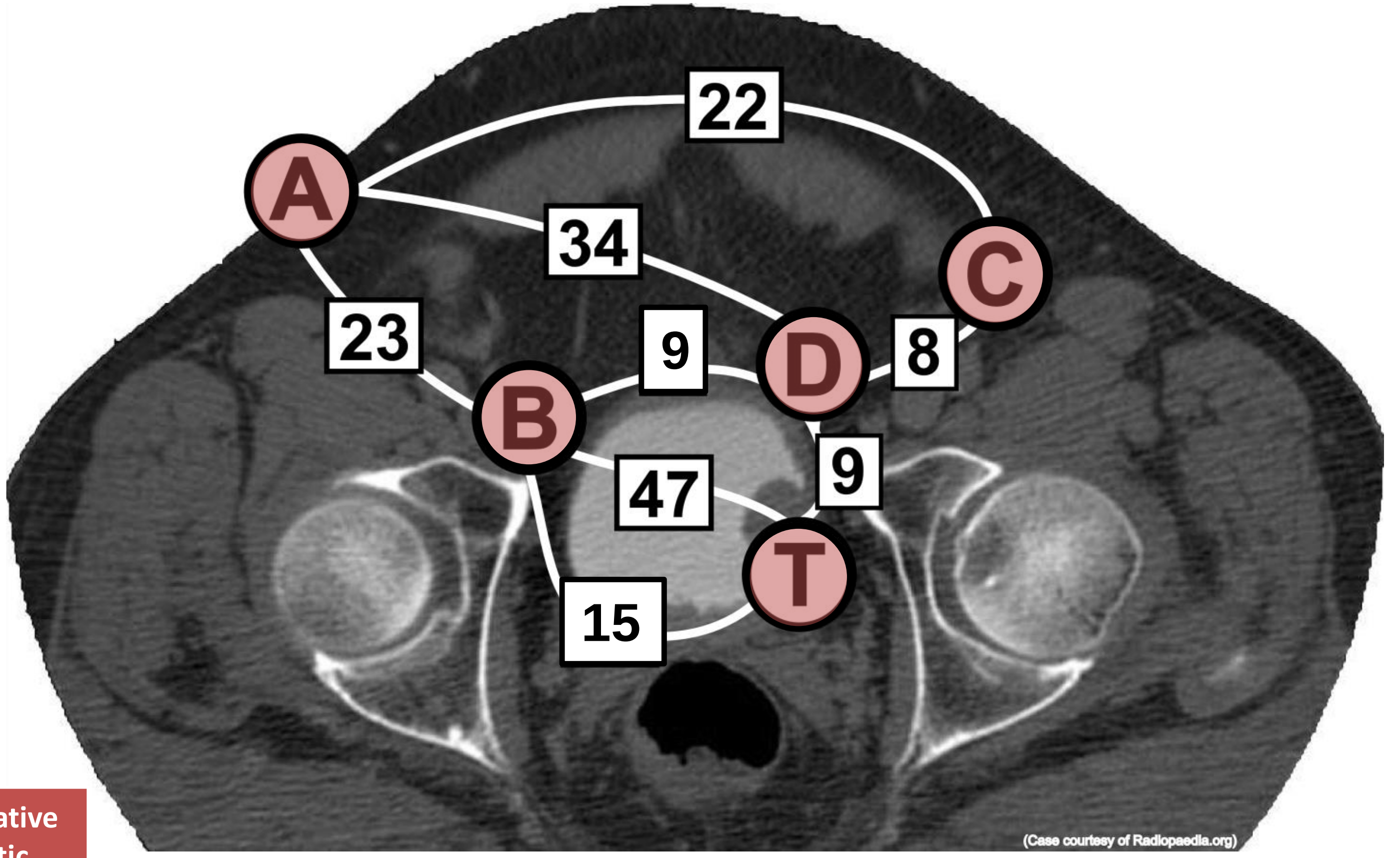


node	proposed heuristic	alternative heuristic
A	33	**29**
B	15	**14**
C	16	**7**
D	7	**6**
T	0	**0**

Simple Graph Search

Open list:
T(38,B)

Closed list:
A(29,-)
C(29,A)
D(36,C)
B(37,A)

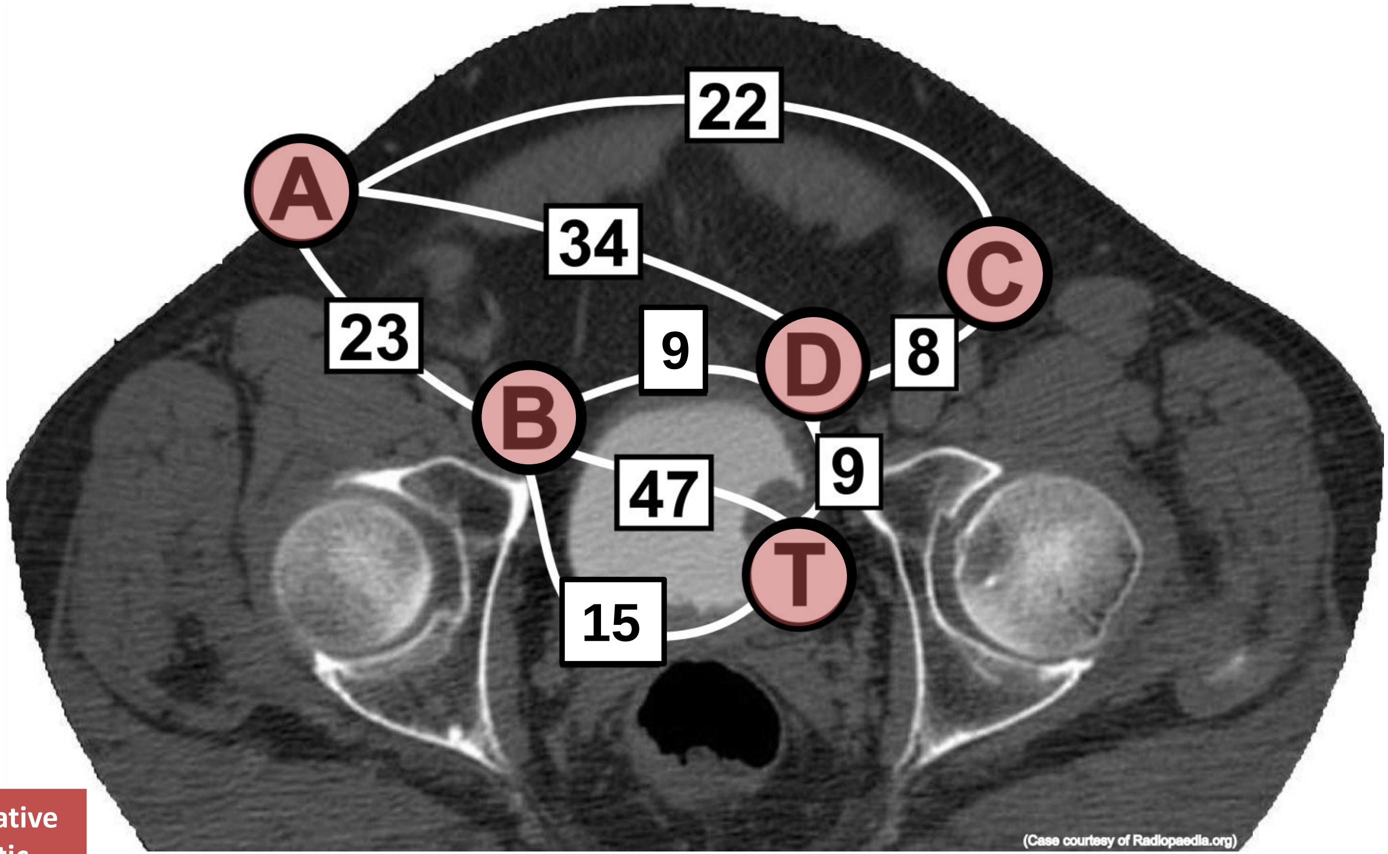


node	proposed heuristic	alternative heuristic
A	33	**29**
B	15	**14**
C	16	**7**
D	7	**6**
T	0	**0**

Simple Graph Search

Closed list:
A(29,-)
C(29,A)
D(36,C)
B(37,A)
T(38,B)

→ A-B-T



node	proposed heuristic	alternative heuristic
A	33	**29**
B	15	**14**
C	16	**7**
D	7	**6**
T	0	**0**

Solution A*-Search

Point	Open list	Closed list
A	C(22+7, A) D(34+6, A) B(23+14, A)	A(29, A)
C	D(30+6, C) B(23+14, A)	A(29, A) C(29, A)
D	T(39, D) B(37, A)	A(29, A) C(29, A) D(36, C)
B	T(38, B)	A(29, A) C(29, A) D(36, C) B(37, A)
T	-	A(29, A) C(29, A) D(36, C) B(37, A) T(38, B)

Task2: Constrained Optimization

You are solving a treatment planning problem resulting in the following linear program optimizing the beam-on times.

$$
\begin{aligned}
\max \quad & f = 3x_1 + 5x_2 \\
\text{s.t.} \quad & 4x_1 + 6x_2 \geq 24 \\
& 5x_1 + 2x_2 \leq 20 \\
& 2x_1 + 4x_2 \leq 20
\end{aligned}
$$

Additional safety constraints require all beams to have a beam-on time of at most 4min. To balance the dose, each beam should be active for at least 2min.

- Add all necessary constraints.
- Remove redundant constraints.

Constrained Optimization

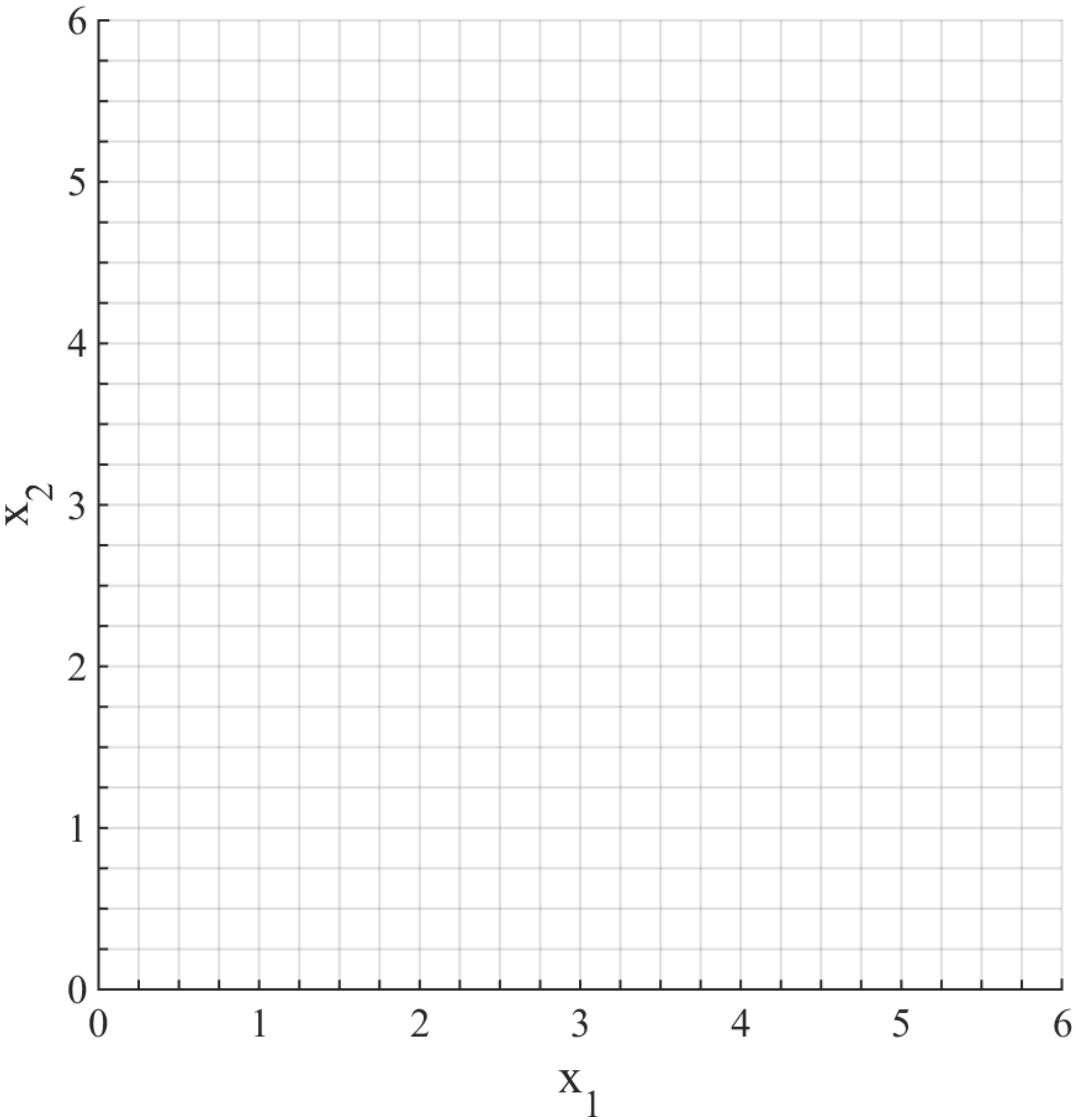
You are solving a treatment planning problem resulting in the following linear program optimizing the beam-on times.

$$
\begin{aligned}
\max \quad & f = 3x_1 + 5x_2 \\
\text{s.t.} \quad & 4x_1 + 6x_2 \geq 24 \\
& 5x_1 + 2x_2 \leq 20 \\
& 2x_1 + 4x_2 \leq 20 \\
& x_1,\ x_2 \geq 2 \\
& x_1,\ x_2 \leq 4
\end{aligned}
$$

Additional safety constraints require all beams to have a beam-on time of at most 4min. To balance the dose, each beam should be active for at least 2min.

- Add all necessary constraints.
- Remove redundant constraints.

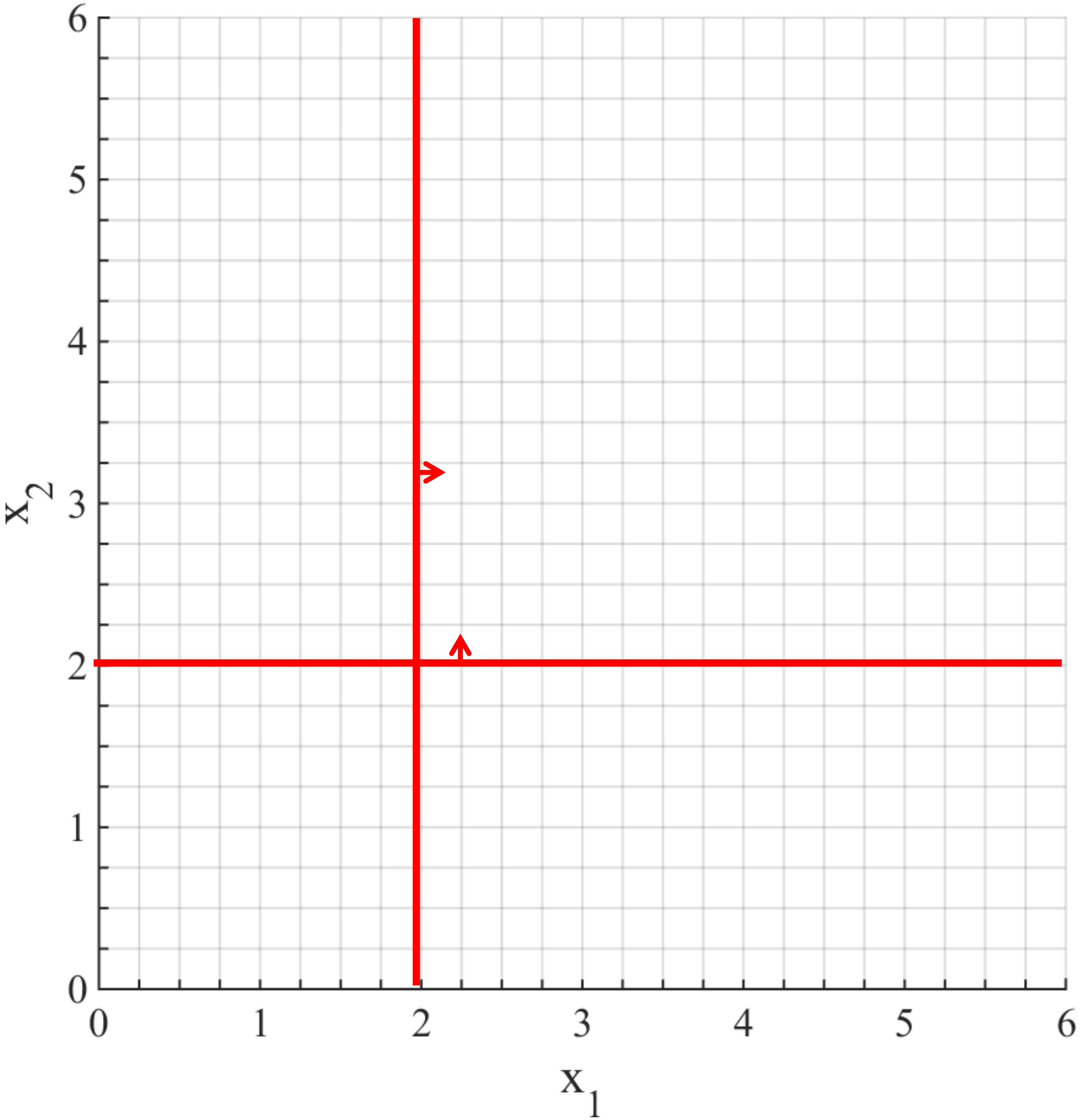
Constrained Optimization



$$\max f = 3x_1 + 5x_2$$
$$4x_1 + 6x_2 \geq 24$$
$$5x_1 + 2x_2 \leq 20$$
$$2x_1 + 4x_2 \leq 20$$
$$x_1, x_2 \geq 2$$
$$x_1, x_2 \leq 4$$

- Redundant constraints?
- Solution?

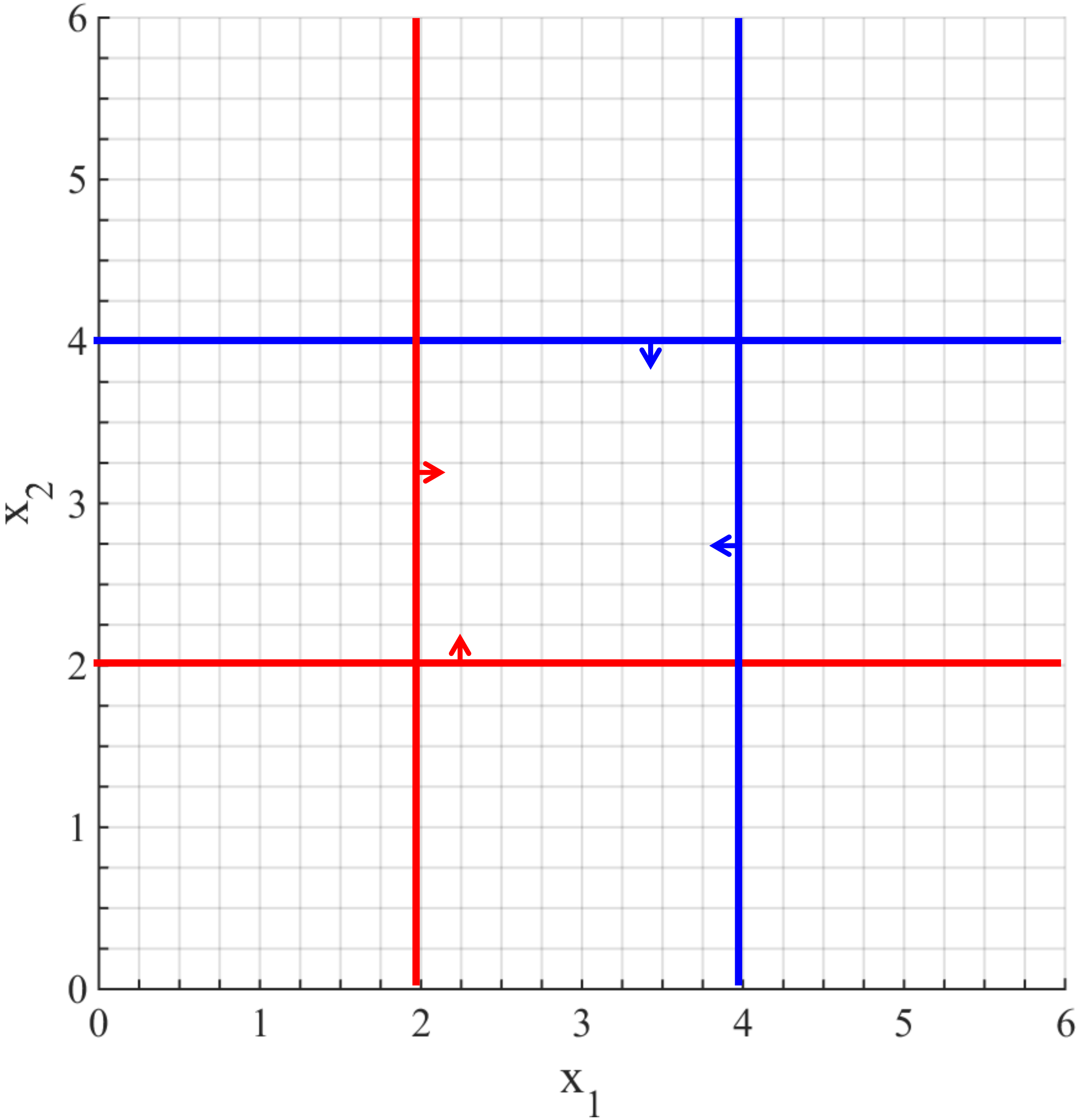
Constrained Optimization



$$\max f = 3x_1 + 5x_2$$
$$4x_1 + 6x_2 \geq 24$$
$$5x_1 + 2x_2 \leq 20$$
$$2x_1 + 4x_2 \leq 20$$
$$x_1, x_2 \geq 2$$
$$x_1, x_2 \leq 4$$

- Redundant constraints?
- Solution?

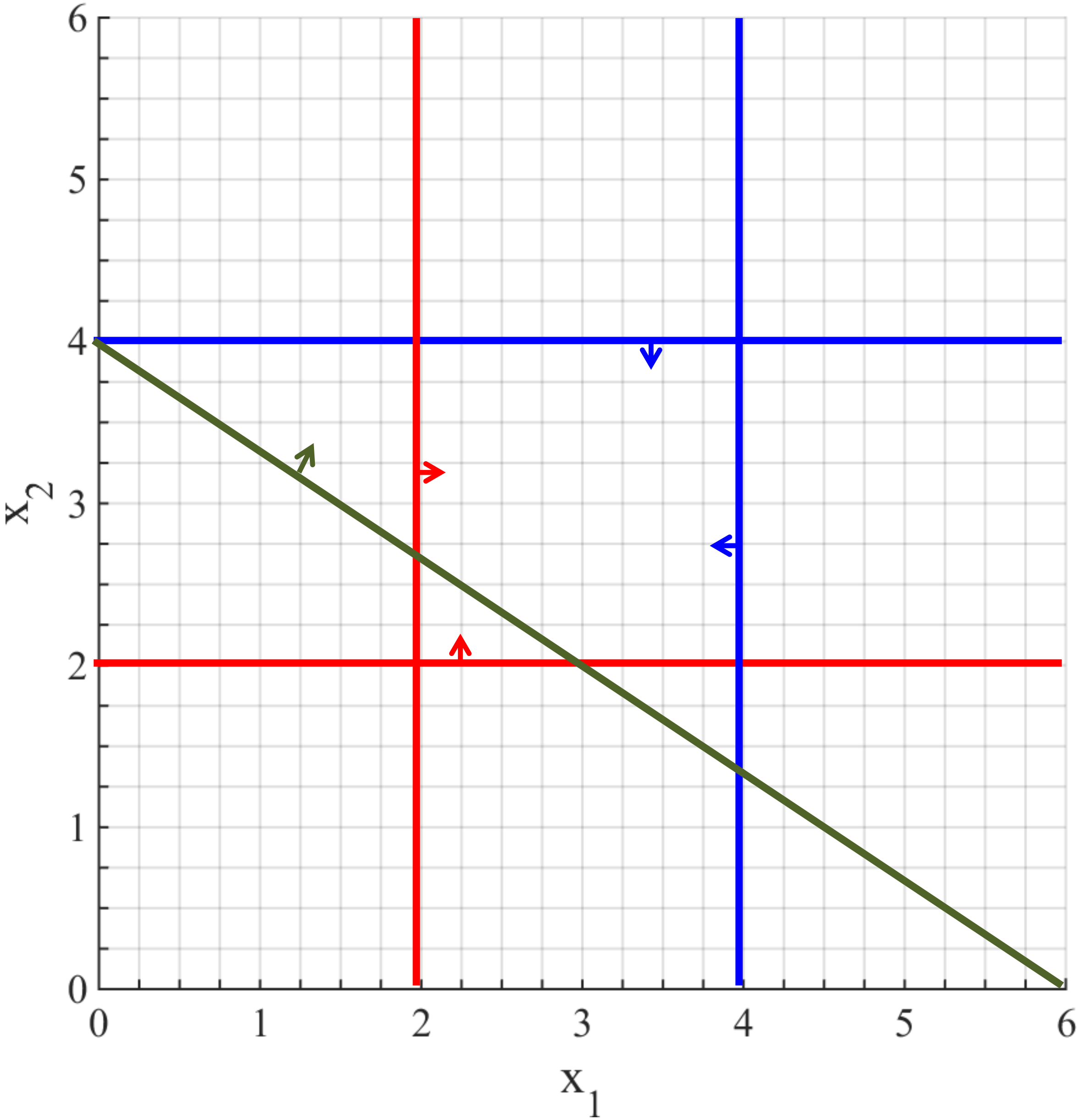
Constrained Optimization



$$\max f = 3x_1 + 5x_2$$
$$4x_1 + 6x_2 \geq 24$$
$$5x_1 + 2x_2 \leq 20$$
$$2x_1 + 4x_2 \leq 20$$
$$x_1,\ x_2 \geq 2$$
$$x_1,\ x_2 \leq 4$$

- Redundant constraints?
- Solution?

Constrained Optimization



$\max f = 3x_1 + 5x_2$

$4x_1 + 6x_2 \geq 24$

$5x_1 + 2x_2 \leq 20$

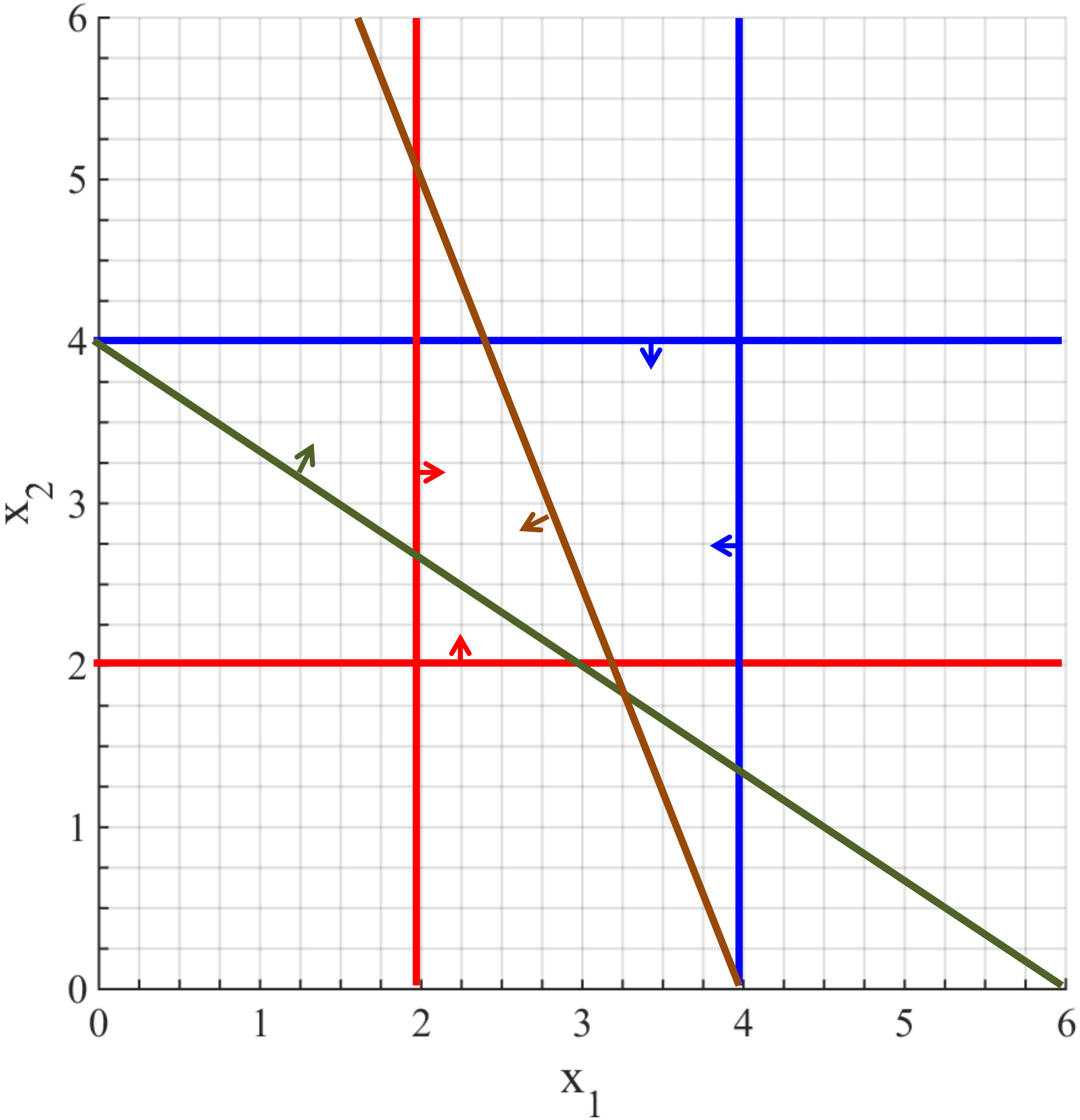
$2x_1 + 4x_2 \leq 20$

$x_1, x_2 \geq 2$

$x_1, x_2 \leq 4$

- Redundant constraints?
- Solution?

Constrained Optimization



$\max f = 3x_1 + 5x_2$

$4x_1 + 6x_2 \geq 24$

$5x_1 + 2x_2 \leq 20$

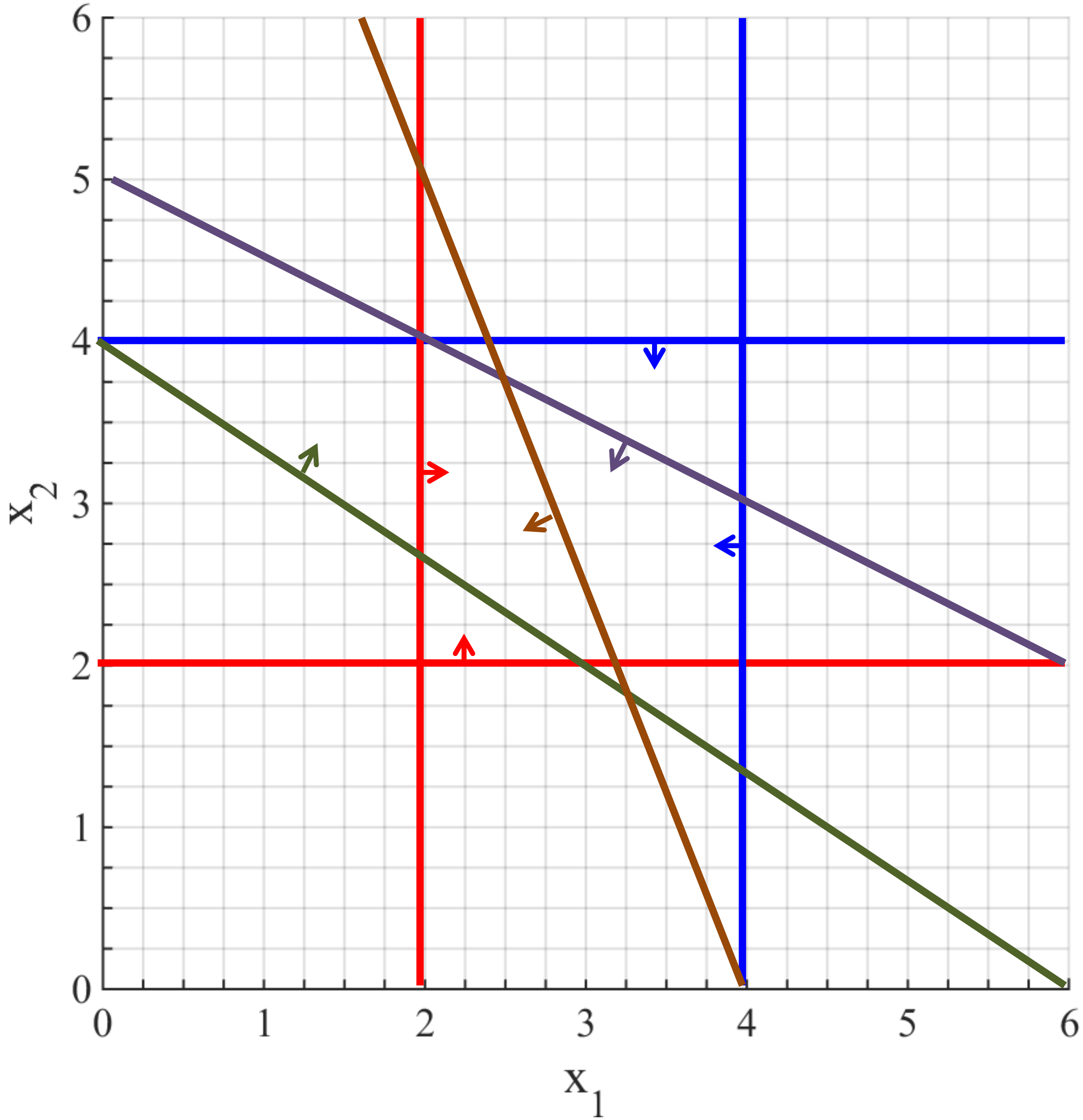
$2x_1 + 4x_2 \leq 20$

$x_1, x_2 \geq 2$

$x_1, x_2 \leq 4$

- Redundant constraints?
- Solution?

Constrained Optimization



$\max f = 3x_1 + 5x_2$

$4x_1 + 6x_2 \geq 24$

$5x_1 + 2x_2 \leq 20$

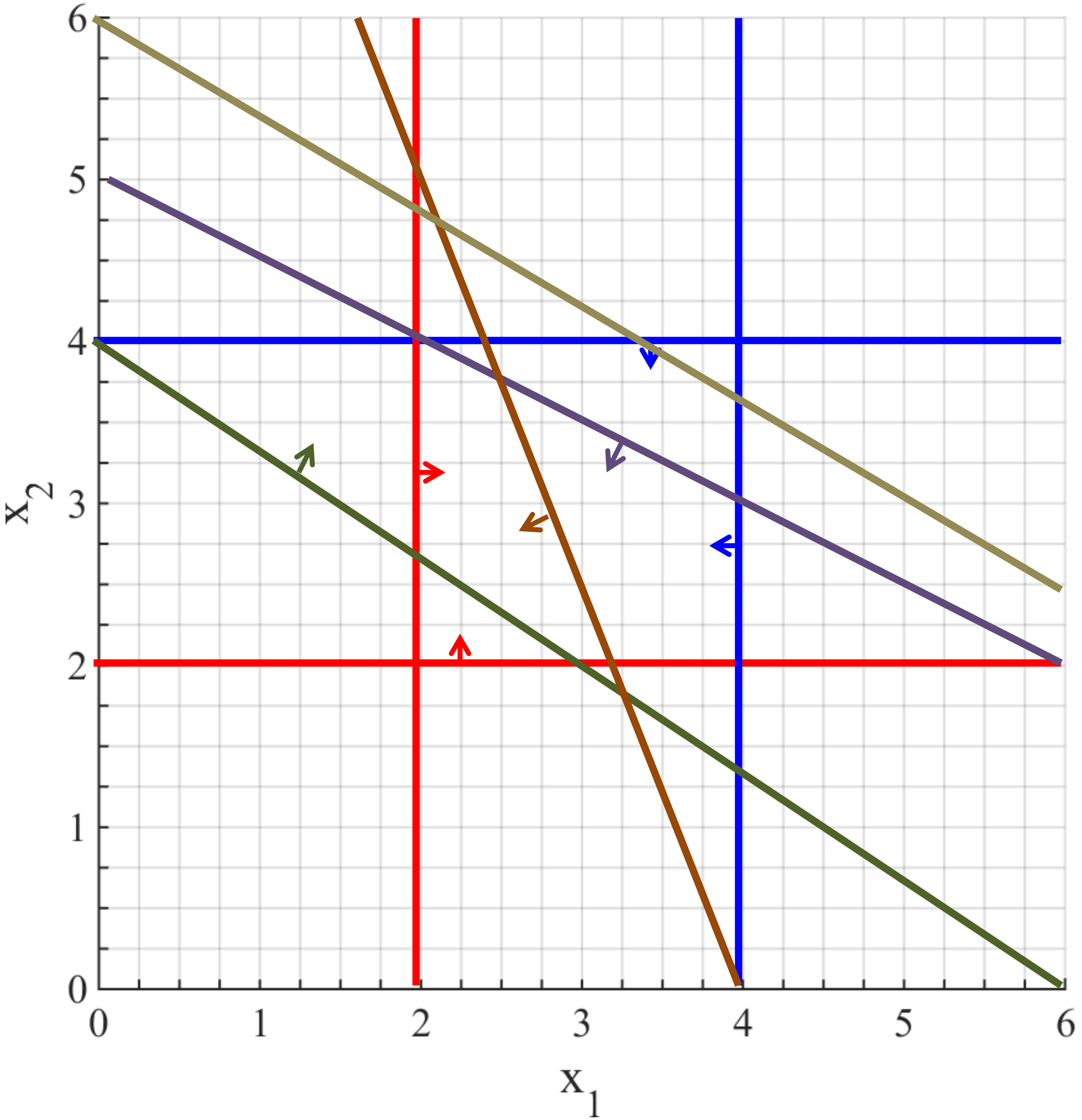
$2x_1 + 4x_2 \leq 20$

$x_1, x_2 \geq 2$

$x_1, x_2 \leq 4$

- Redundant constraints?
- Solution?

Constrained Optimization



$\max f = 3x_1 + 5x_2$

$4x_1 + 6x_2 \geq 24$

$5x_1 + 2x_2 \leq 20$

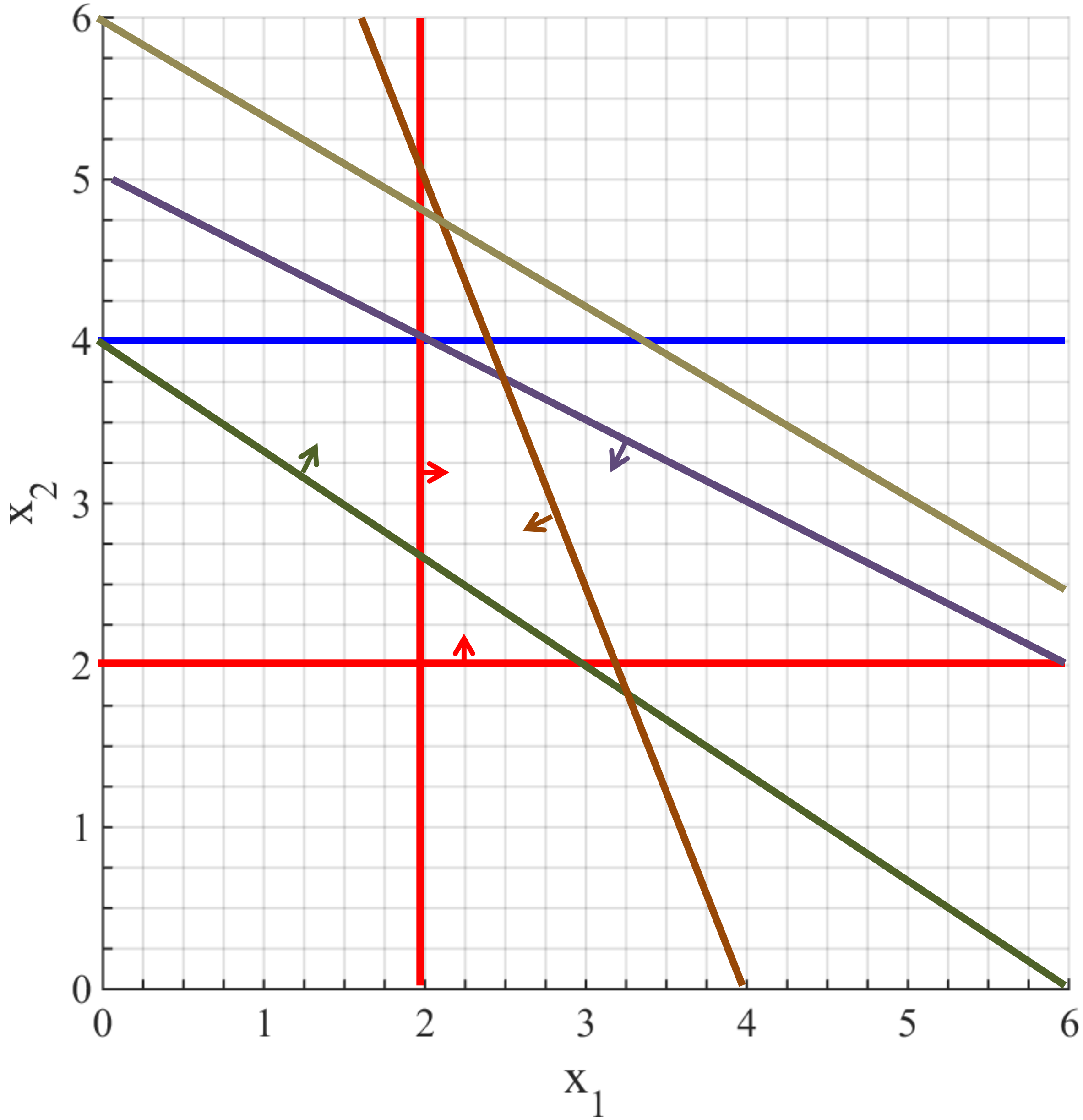
$2x_1 + 4x_2 \leq 20$

$x_1, x_2 \geq 2$

$x_1, x_2 \leq 4$

- Redundant constraints?
- Solution?

Constrained Optimization



$\max f = 3x_1 + 5x_2$

$4x_1 + 6x_2 \geq 24$

$5x_1 + 2x_2 \leq 20$

$2x_1 + 4x_2 \leq 20$

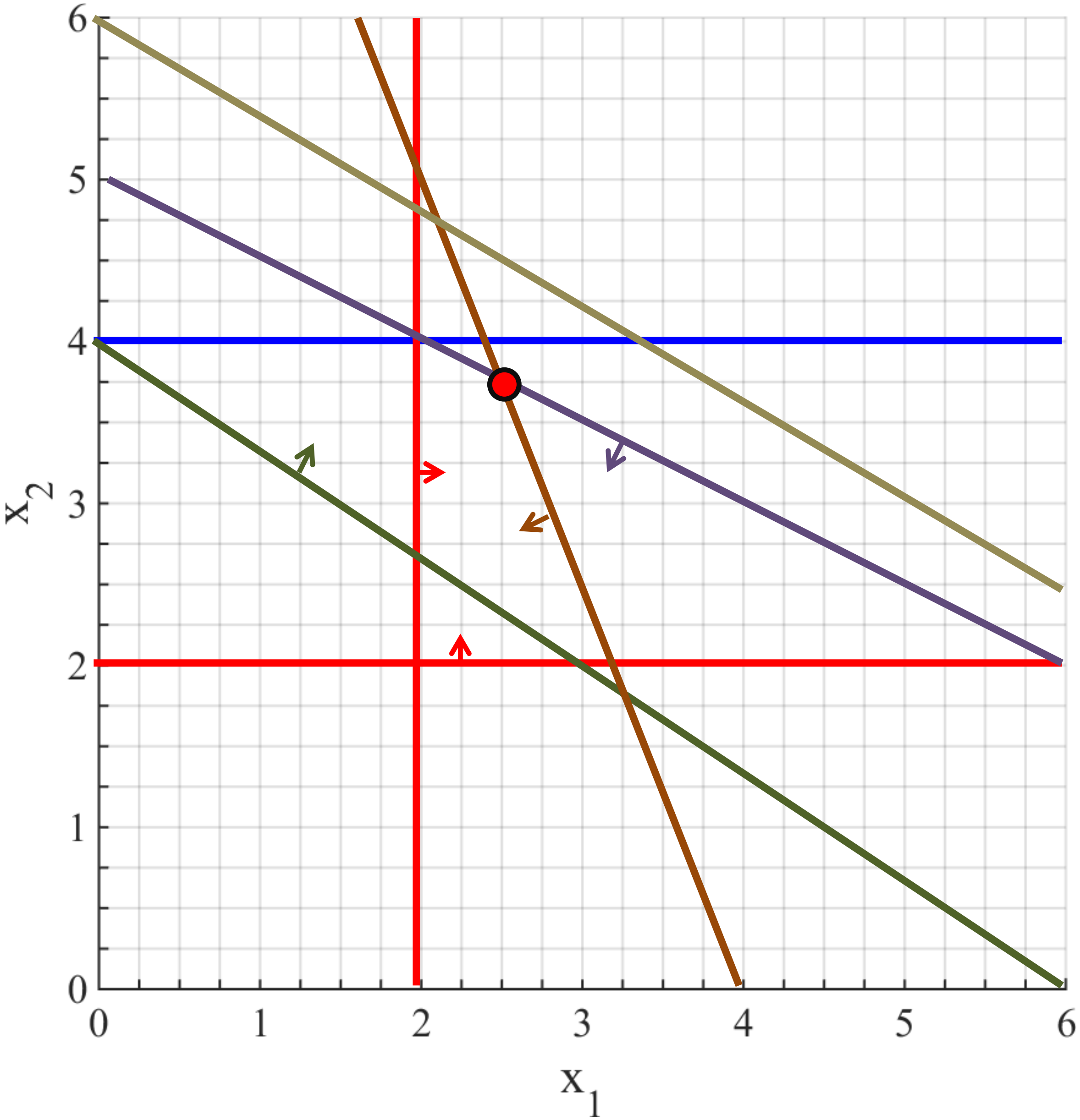
$x_1, x_2 \geq 2$

- Redundant constraints?

  $x_1, x_2 \leq 4$

- Solution?

Constrained Optimization



max $f = 3x_1 + 5x_2$

$4x_1 + 6x_2 \geq 24$

$5x_1 + 2x_2 \leq 20$

$2x_1 + 4x_2 \leq 20$

$x_1, x_2 \geq 2$

- Redundant constraints?

$x_1, x_2 \leq 4$

- Solution?

$5x_1 + 2x_2 = 2x_1 + 4x_2$

$= 20$

$\rightarrow x_1 = 2.5, x_2 = 3.75$

What to do for harder problems?

- Use heuristics!
- Simulated annealing
- Gradient descent/hill climbing
- → Neural networks and deep learning