Learning the Parameters of Global Constraints for Medical Scheduling

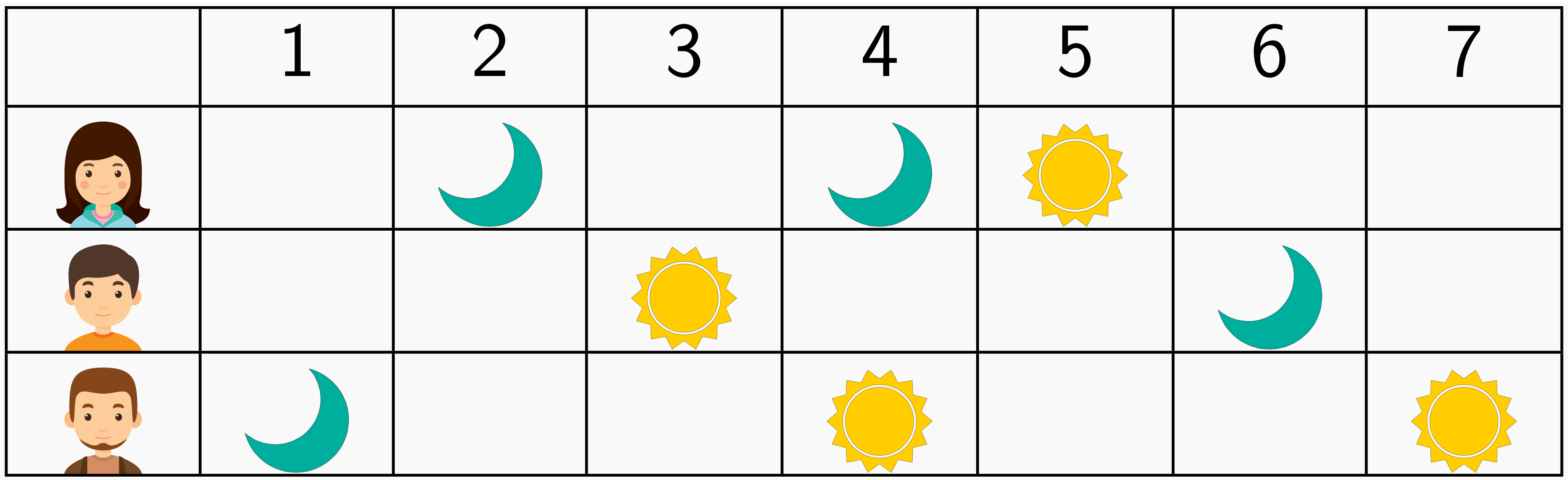
Émilie Picard-Cantin, Mathieu Bouchard,
Claude-Guy Quimper, and Jason Sweeney

Émilie Picard-Cantin

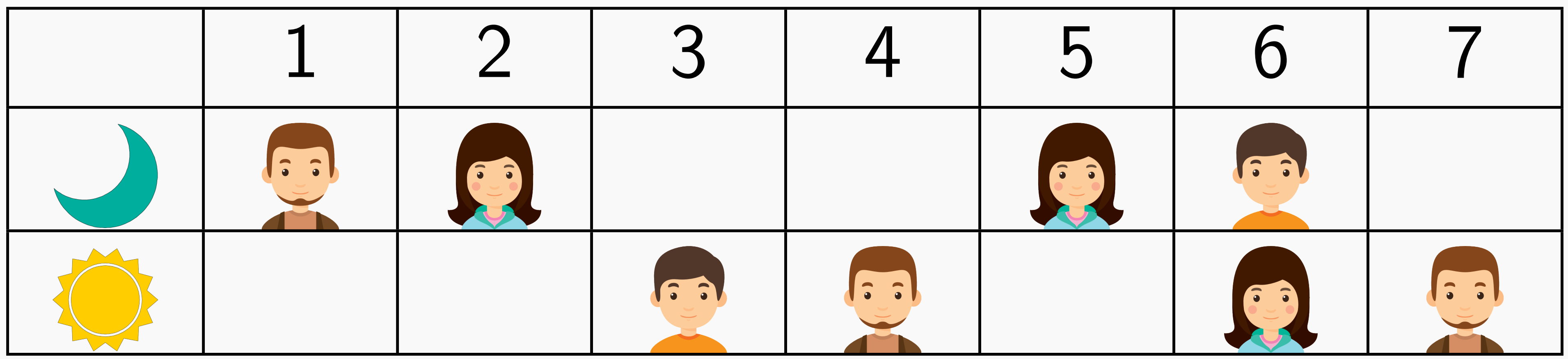
PetalMD

Medical Schedule

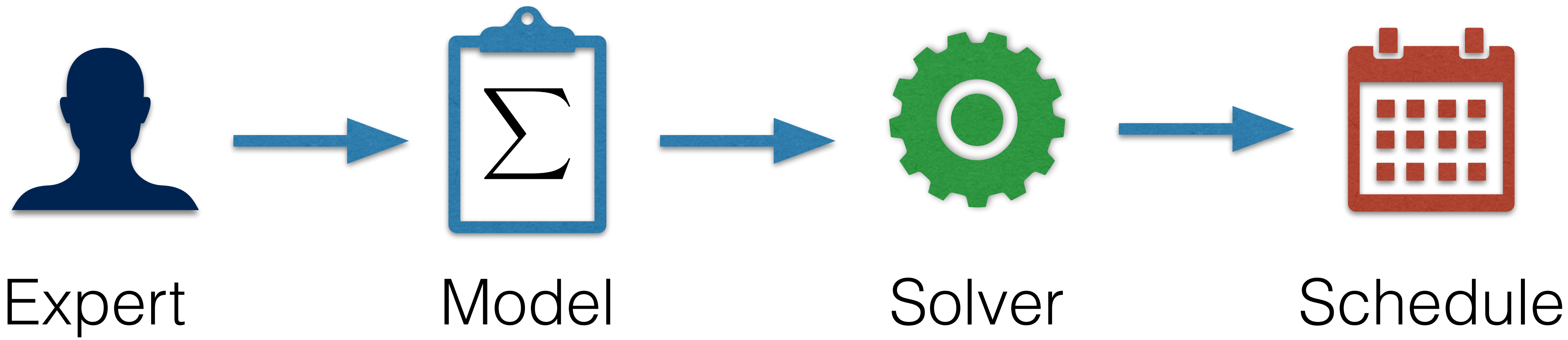
People oriented



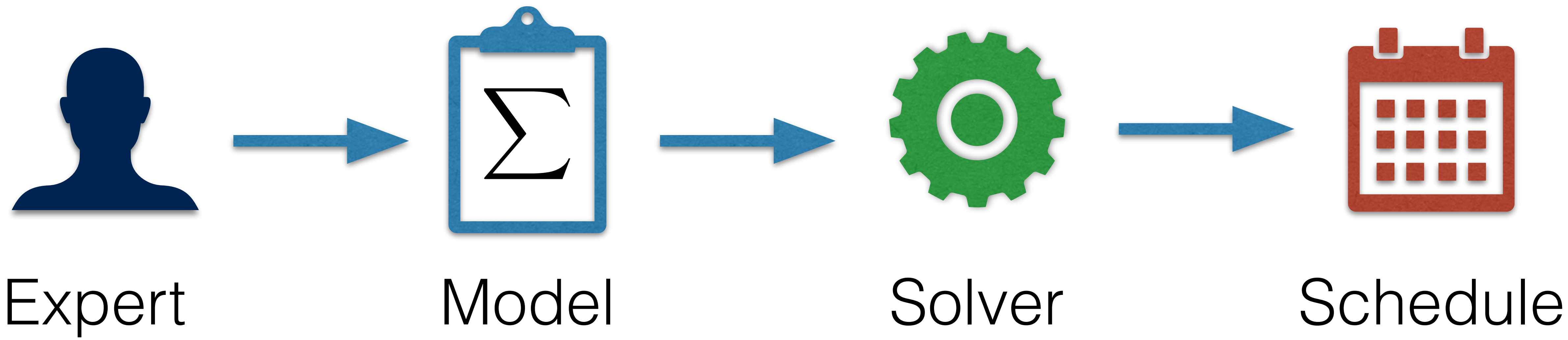
Shift / Task oriented

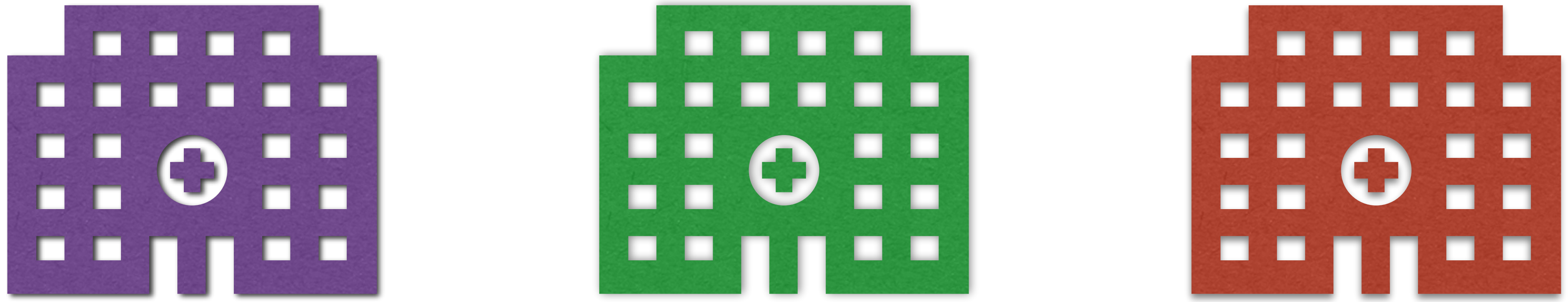


Scheduling Process

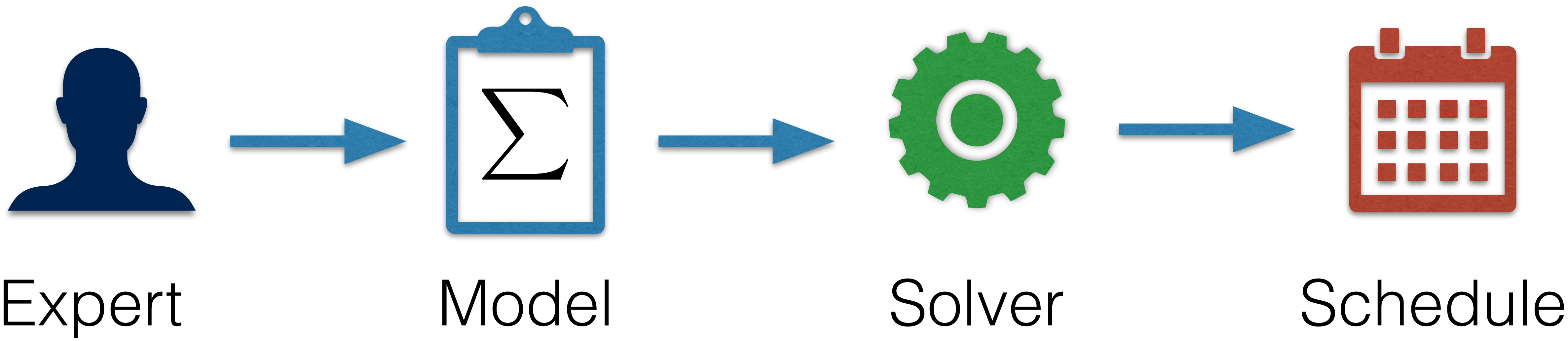


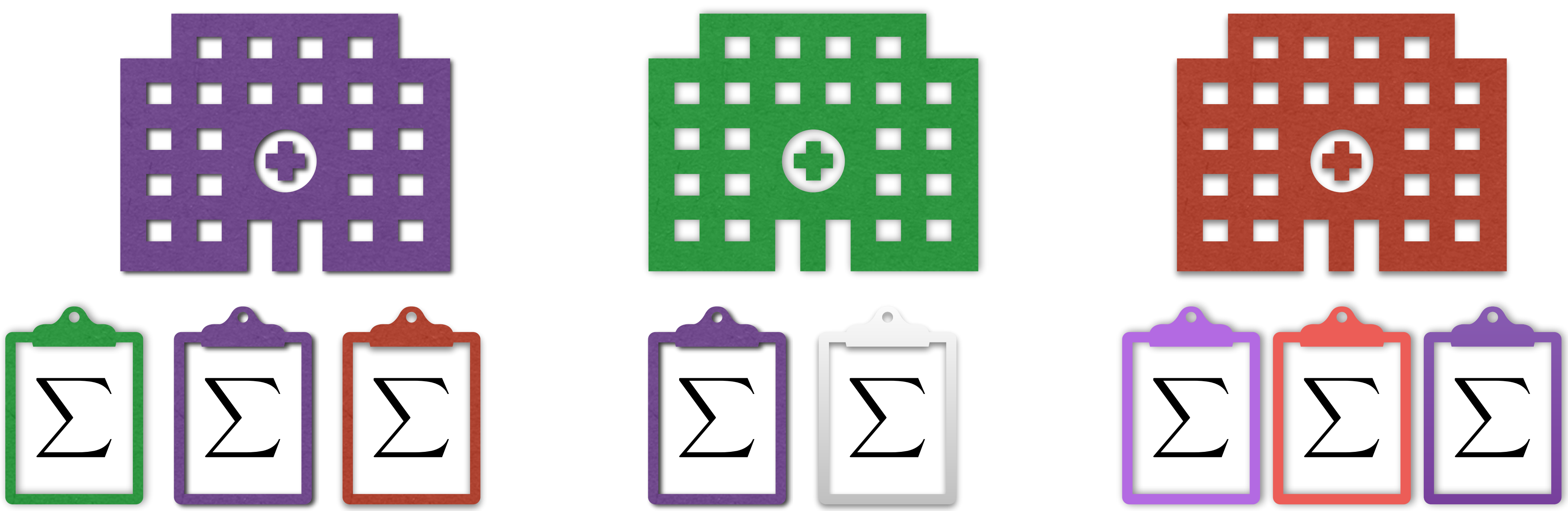
Scheduling Process





Scheduling Process





Speeding up the modelling process

Speeding up the modelling process

- **Challenge**: Models differ from one medical team to another

Speeding up the modelling process

- **Challenge**: Models differ from one medical team to another

- **Observation**: There are few differences between each model.

Speeding up the modelling process

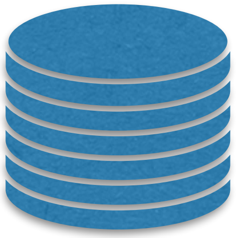
- **Challenge**: Models differ from one medical team to another
- **Observation**: There are few differences between each model.
- **Opportunity**: For legal reasons, hospitals keep a history of their schedules.

Speeding up the modelling process

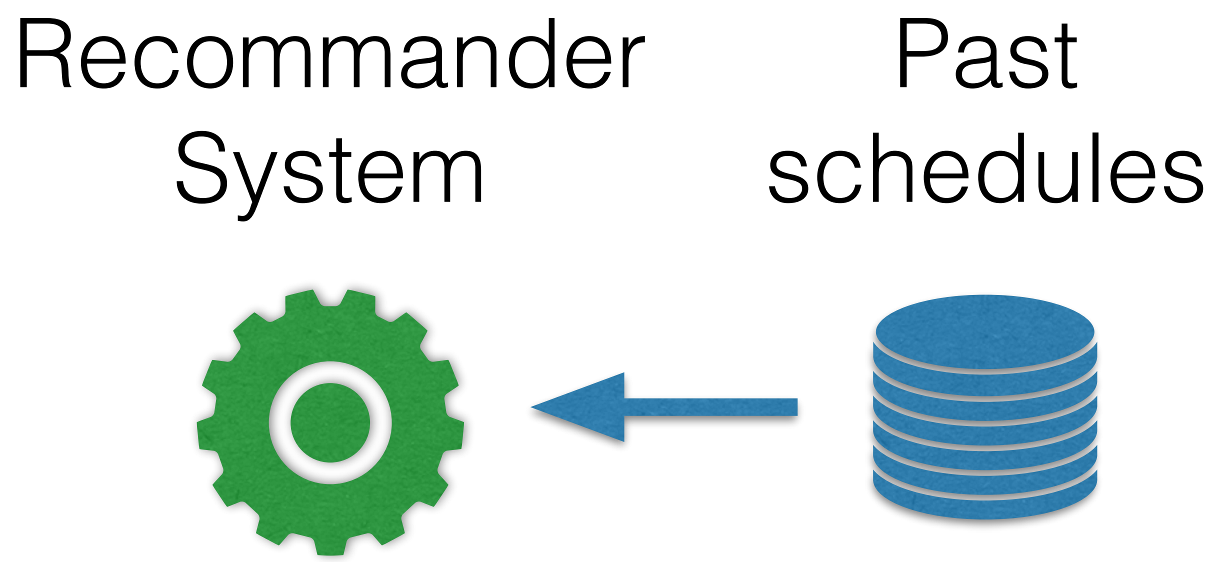
- **Challenge**: Models differ from one medical team to another
- **Observation**: There are few differences between each model.
- **Opportunity**: For legal reasons, hospitals keep a history of their schedules.
- **Goal**: To learn the models from historical data.

Recommander System

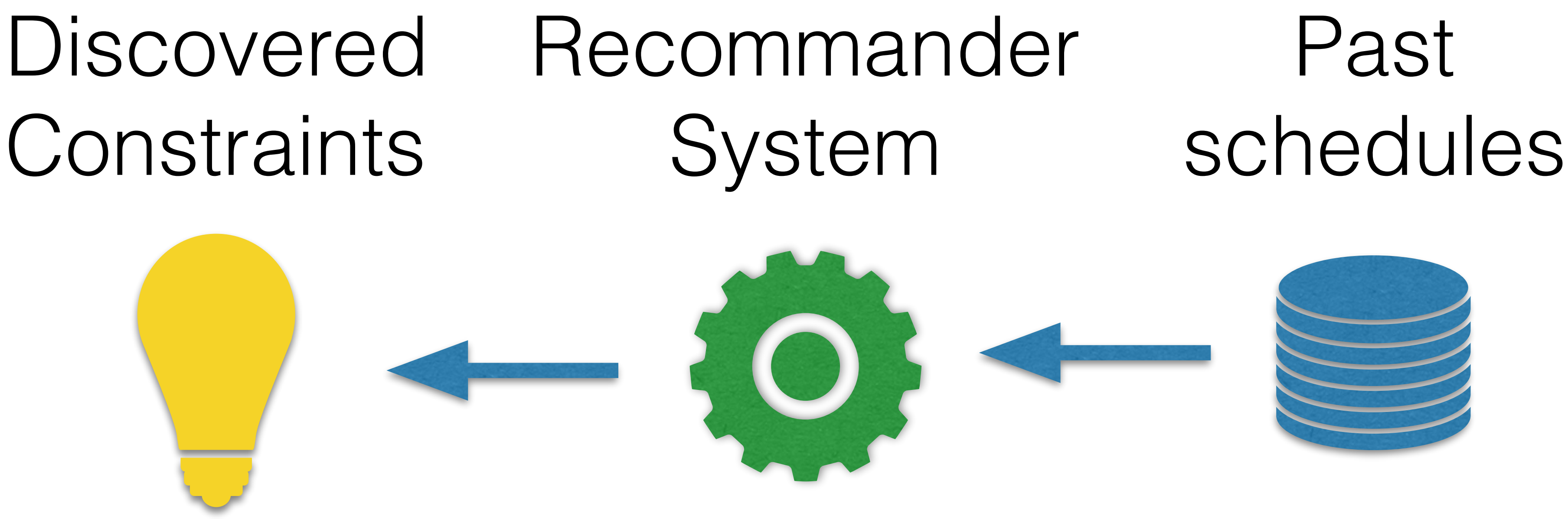
Past
schedules



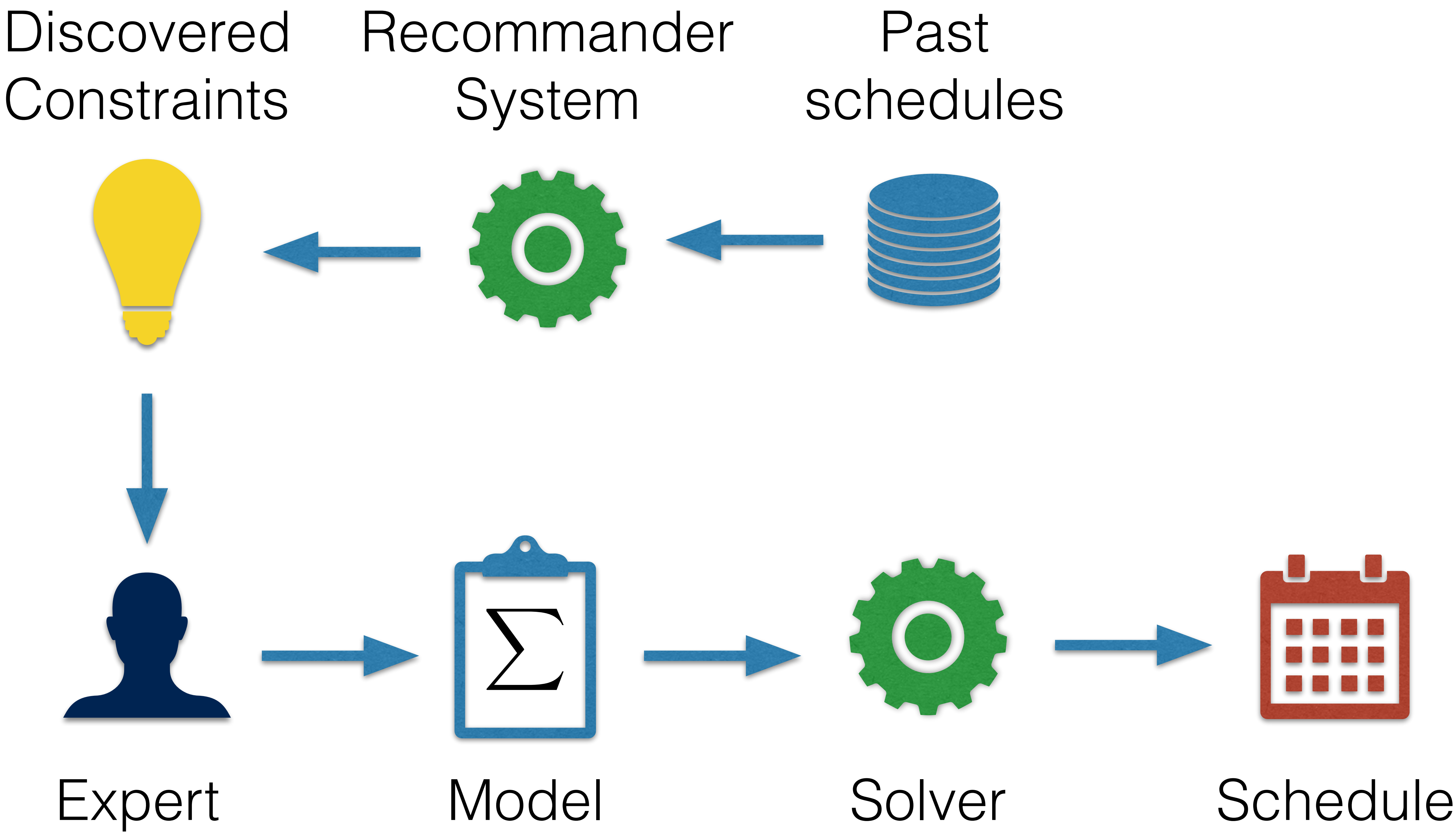
Recommander System



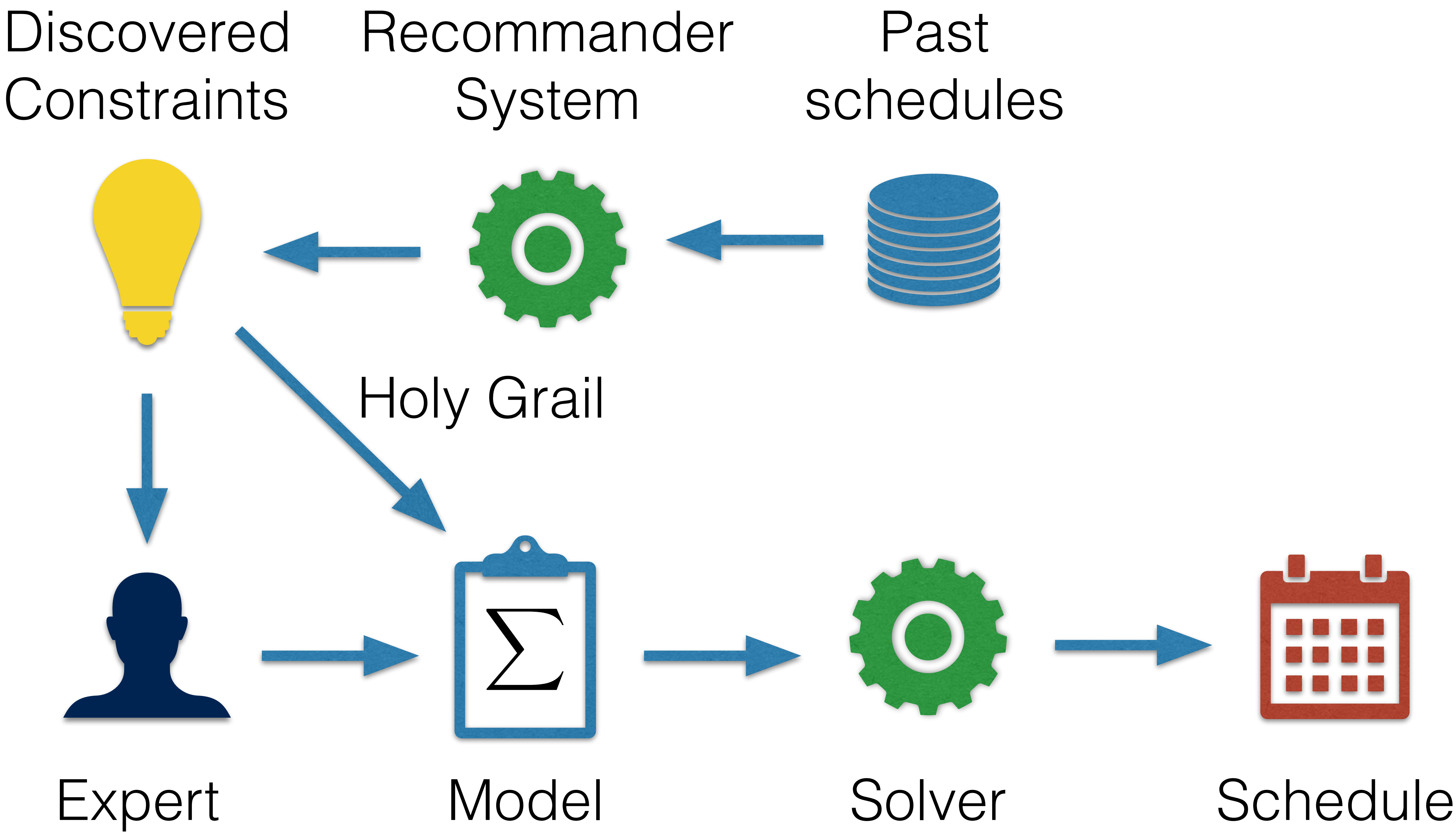
Recommander System



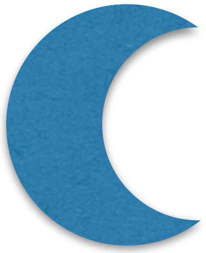
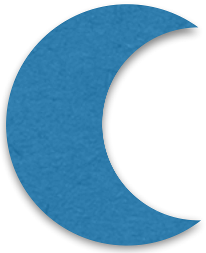
Recommander System



Recommander System

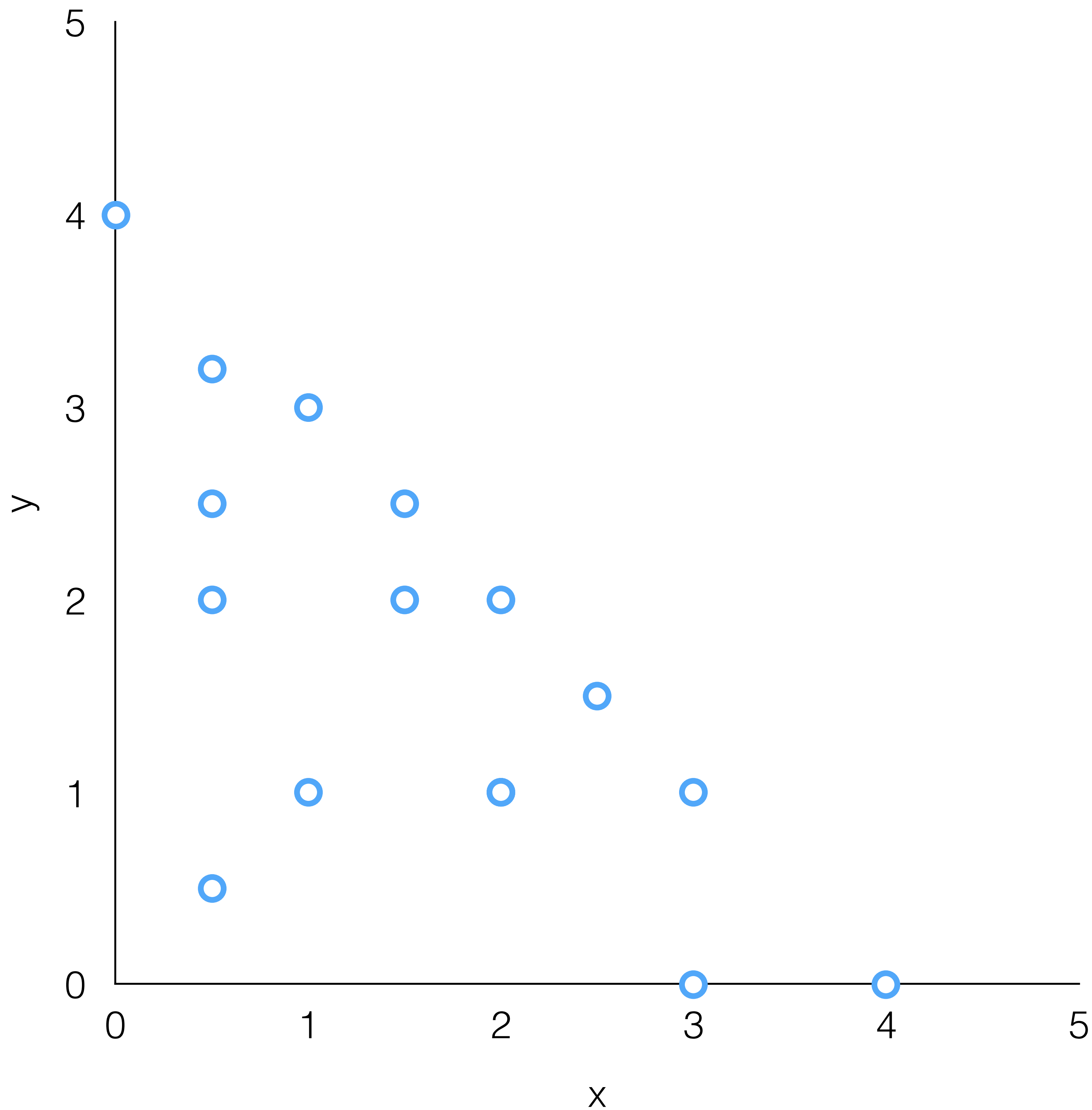


How to learn a constraint?

1	2	3	4	5	6	7
						

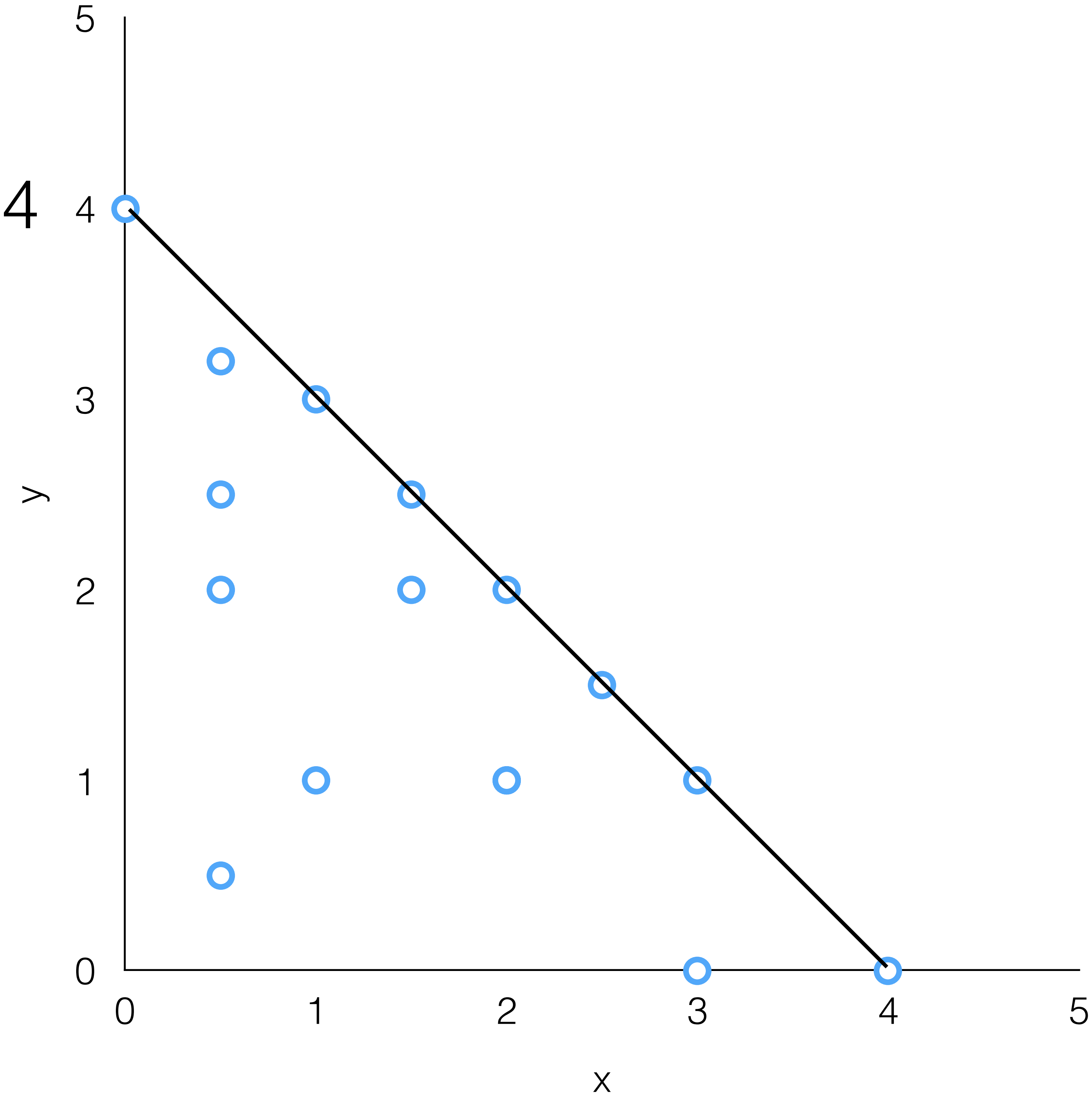
- Do we have a limit of:
  - 2 night-shifts per week?
  - 1 night-shift every 3 days?

Which constraint was imposed?



Which constraint was imposed?

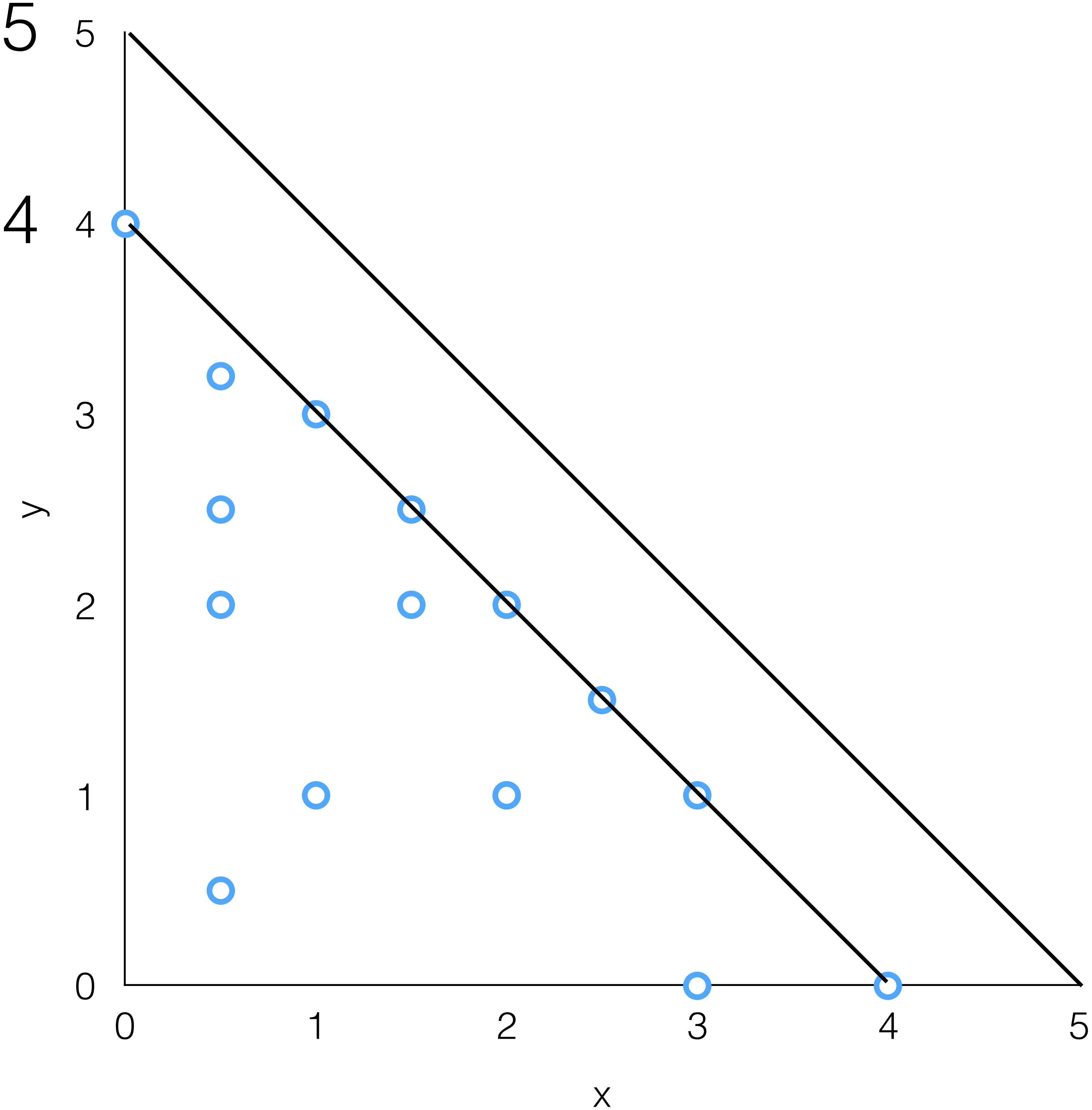
$x + y \leq 4$



Which constraint was imposed?

$x + y \leq 5$

$x + y \leq 4$



Problem Definition

- Consider a random assignment $\vec{X}$ with $P[X_i = v] = p_v$

Problem Definition

- Consider a random assignment $\vec{X}$ with $P[X_i = v] = p_v$
- The probability of observing $\vec{X}$ is $P(\vec{X}) = \Pi_{i=1}^{n} p_{X_i}$

Problem Definition

- Consider a random assignment $\vec{X}$ with $P[X_i = v] = p_v$
- The probability of observing $\vec{X}$ is $P(\vec{X}) = \Pi_{i=1}^{n} p_{X_i}$
- Consider the constraint $C([X_1, \ldots, X_n],[\alpha_1, \ldots, \alpha_m])$

Problem Definition

- Consider a random assignment $\vec{X}$ with $P[X_i = v] = p_v$
- The probability of observing $\vec{X}$ is $P(\vec{X}) = \Pi_{i=1}^{n} p_{X_i}$
- Consider the constraint $C([X_1, \ldots, X_n],[\alpha_1, \ldots, \alpha_m])$
- The probability that a random assignment satisfies C is

$$G_C(\vec{\alpha}) = \sum_{\vec{X} | C(\vec{X},\vec{\alpha})} P(\vec{X})$$

Problem Definition

- Consider a random assignment $\vec{X}$ with $P[X_i = v] = p_v$
- The probability of observing $\vec{X}$ is $P(\vec{X}) = \Pi_{i=1}^{n} p_{X_i}$
- Consider the constraint $C([X_1, \ldots, X_n],[\alpha_1, \ldots, \alpha_m])$
- The probability that a random assignment satisfies C is

$$G_C(\vec{\alpha}) = \sum_{\vec{X} | C(\vec{X}, \vec{\alpha})} P(\vec{X})$$

Problem Definition

- Consider a random assignment $\vec{X}$ with $P[X_i = v] = p_v$
- The probability of observing $\vec{X}$ is $P(\vec{X}) = \Pi_{i=1}^{n} p_{X_i}$
- Consider the constraint $C([X_1, \ldots, X_n],[\alpha_1, \ldots, \alpha_m])$
- The probability that a random assignment satisfies C is

$$G_C(\vec{\alpha}) = \sum_{\vec{X} | C(\vec{X}, \vec{\alpha})} P(\vec{X})$$

- Finding $\vec{\alpha}$ consists in solving:

$$\min_{\vec{\alpha}} G_C(\vec{\alpha})$$

$$C(\vec{X}, \vec{\alpha}) \qquad \forall \vec{X} \in \text{Examples}$$

How to compute $G_C(\vec{\alpha})$?

- Enumerating and summing the probability of all solutions of a constraint is slow.
- We mainly developed two techniques to compute or bound this probability
  - Using Markov chains
  - Using dynamic programming

Markov Chains

- Some constraints can naturally be encoded with an automaton.

1 = a night shift

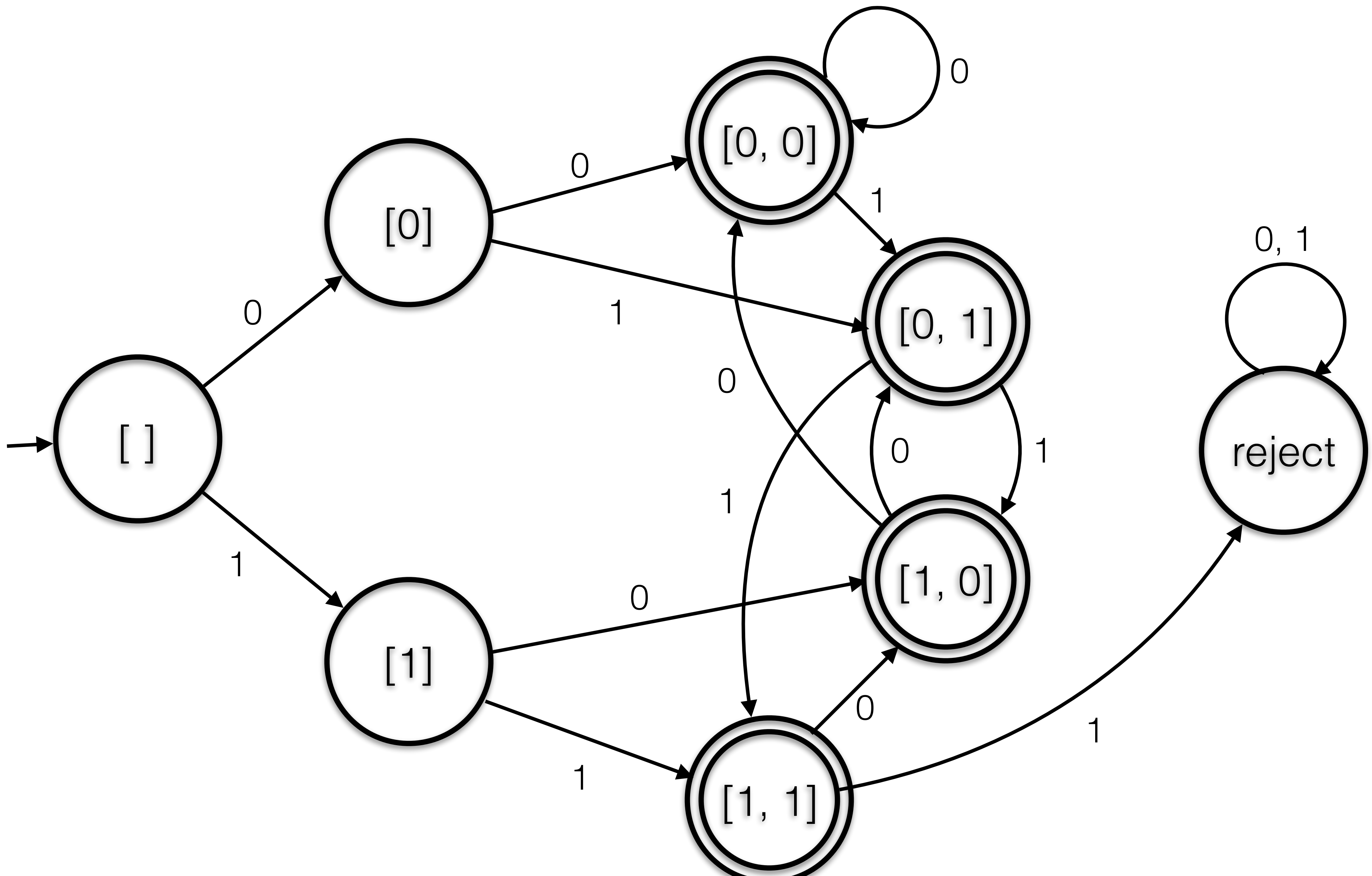
$$\textsc{Sequence}([X_1, \ldots, X_n], \{1\}, 0, 2, 3)$$

at least 0

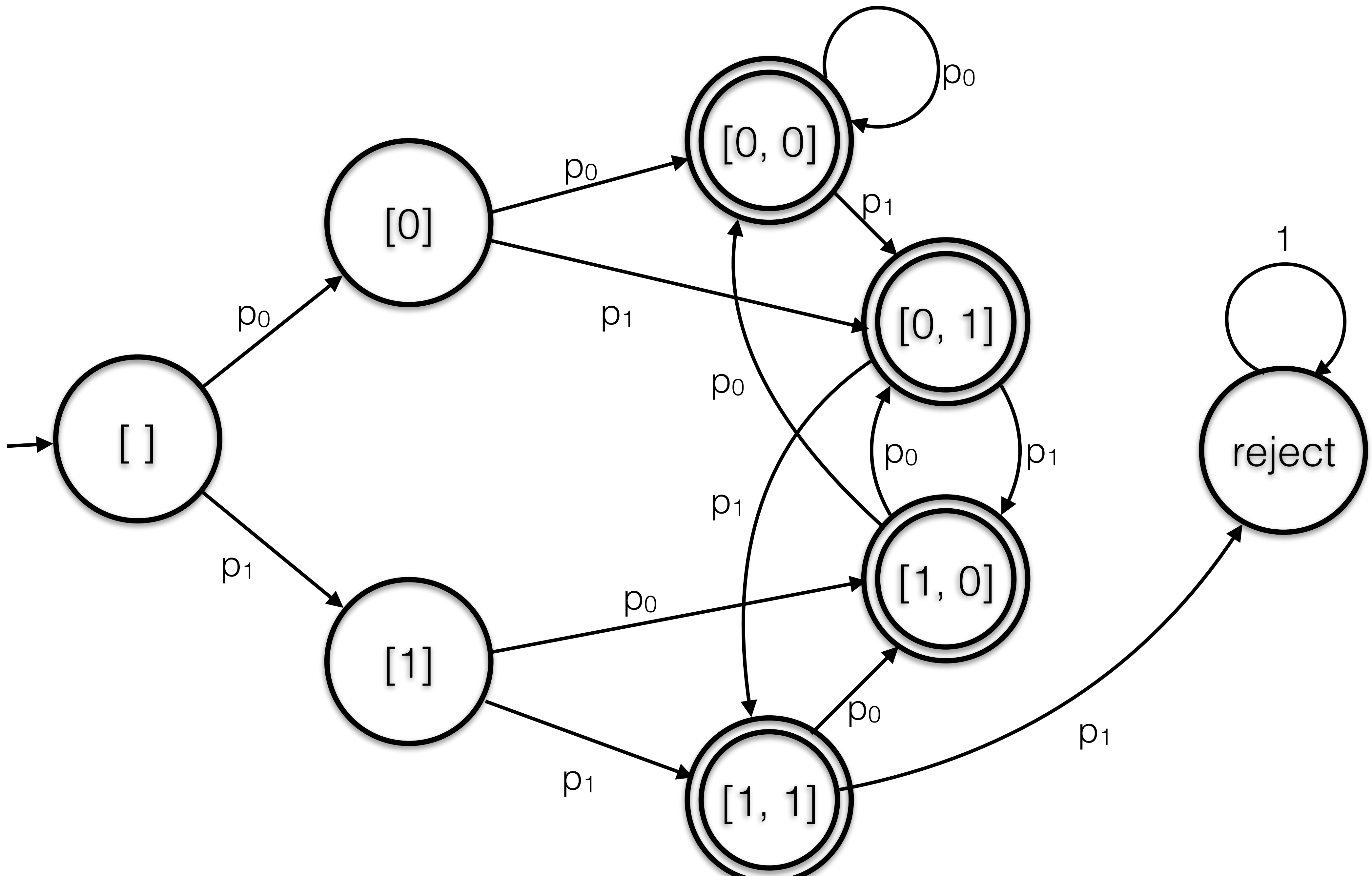
at most 2

every 3 days

Sequence Automaton



Sequence Markov Chain



Computing $G_{\textsc{Sequence}}(\vec{\alpha})$

- Let $M_{\vec{\alpha}}$ be the transition matrix of the Markov chain for the constraint with parameters $\vec{\alpha}$.

- One can compute the probability of reaching the reject state after reading *n* characters by computing $M_{\vec{\alpha}}^n$.

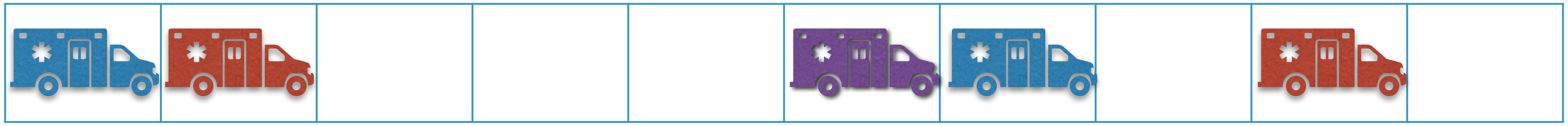
- For every combination of $\vec{\alpha}$, compute $M_{\vec{\alpha}}^n$ and evaluate $G_C(\vec{\alpha})$.

- Keep $\vec{\alpha}$ that minimizes $G_C(\vec{\alpha})$.

When parameters are sets

- If the parameter contains a set, there is an exponential number of combinations to explore.

$$\textsc{Among}([X_1, \ldots, X_n], l, u, \vec{z})$$
$$\textsc{Sequence}([X_1, \ldots, X_n], l, u, w, \vec{z})$$
$$\textsc{SubSetFocus}([X_1, \ldots, X_n], l, m, \vec{z})$$



Branch & Bound

$$\min_{\vec{\alpha}} G_C(\vec{\alpha})$$

$$C(\vec{X}, \vec{\alpha}) \qquad \forall \vec{X} \in \text{Examples}$$

- We use this strategy to compute a bound on $G_C(\vec{\alpha})$

$$p = \sum_v z_v \cdot p_v$$

$$\begin{aligned} G_C(\vec{\alpha}) &\geq G_C(ext(\vec{\alpha})) \\ &\geq \sum_{k=0}^{n} D[k, ext(\vec{\alpha})] \left(\min_p p^k (1-p)^{n-k} \right) \end{aligned}$$

Branch & Bound

$$\min_{\vec{\alpha}} G_C(\vec{\alpha})$$

$$\forall \vec{X} \in \text{Examples}$$

Probability that a value belongs to the set.

- We use this … ompute a bound on $G_C(\vec{\alpha})$

$$p = \sum_{v} z_v \cdot p_v$$

$$G_C(\vec{\alpha}) \geq G_C(ext(\vec{\alpha}))$$

$$\geq \sum_{k=0}^{n} D[k, ext(\vec{\alpha})] \left(\min_{p} p^k (1-p)^{n-k} \right)$$

Branch & Bound

$$\min_{\vec{\alpha}} G_C(\vec{\alpha})$$

$$C(\vec{X}, \vec{\alpha}) \qquad \forall \vec{X} \in \text{Examples}$$

- We use this strategy to compute a bound on $G_C(\vec{\alpha})$

$$p = \sum_v z_v \cdot p_v$$

$$G_C(\vec{\alpha}) \geq G_C(ext(\vec{\alpha}))$$

$$\geq \sum_{k=0}^{n} D[k, ext(\vec{\alpha})] \left(\min_p p^k (1-p)^{n-k} \right)$$

Branch & Bound

$$\min_{\vec{\alpha}} G_C(\vec{\alpha})$$

$$C(\vec{X}, \vec{\alpha})$$ xamples

Set parameters to
extreme values
(requires monotonicity)

- We use this strategy t nd on $G_C(\vec{\alpha})$

$$p = \sum_v z_v \cdot p_v$$

$$G_C(\vec{\alpha}) \geq G_C(ext(\vec{\alpha}))$$

$$\geq \sum_{k=0}^{n} D[k, ext(\vec{\alpha})] \left(\min_p p^k (1-p)^{n-k} \right)$$

Branch & Bound

$$\min_{\vec{\alpha}} G_C(\vec{\alpha})$$

$$C(\vec{X}, \vec{\alpha}) \qquad \forall \vec{X} \in \text{Examples}$$

- We use this strategy to compute a bound on $G_C(\vec{\alpha})$

$$p = \sum_v z_v \cdot p_v$$

$$\begin{aligned} G_C(\vec{\alpha}) &\geq G_C(ext(\vec{\alpha})) \\ &\geq \sum_{k=0}^{n} D[k, ext(\vec{\alpha})] \left(\min_p p^k (1-p)^{n-k} \right) \end{aligned}$$

Branch & Bound

$$\min_{\vec{\alpha}} G_C(\vec{\alpha})$$

$$C(\vec{X}, \vec{\alpha}) \qquad \forall \vec{X} \in \text{Examples}$$

- We use this strategy to compute a bound on $G_C(\vec{\alpha})$

Count how many solutions exist with exactly k values in the set

$$p_v$$

$$G_C(\ldots t(\vec{\alpha}))$$

$$\geq \sum_{k=0}^{n} D[k, ext(\vec{\alpha})] \left(\min_{p} p^k (1-p)^{n-k} \right)$$

Branch & Bound

$$\min_{\vec{\alpha}} G_C(\vec{\alpha})$$

$$C(\vec{X}, \vec{\alpha}) \qquad \forall \vec{X} \in \text{Examples}$$

- We use this strategy to compute a bound on $G_C(\vec{\alpha})$

Count how many solutions exist with exactly k values in the set

p_v

$G_C($ … $t(\vec{\alpha}))$

Probability of a random assignment with k values in the set

$$\geq \sum_{k=0}^{n} D[k, ext(\vec{\alpha})] \left(\min_p p^k (1-p)^{n-k} \right)$$

Branch & Bound

$$\min_{\vec{\alpha}} G_C(\vec{\alpha})$$

$$C(\vec{X}, \vec{\alpha}) \qquad \forall \vec{X} \in \text{Examples}$$

- We use this strategy to compute a bound on $G_C(\vec{\alpha})$

$$p = \sum_v z_v \cdot p_v$$

$$\begin{aligned} G_C(\vec{\alpha}) &\geq G_C(ext(\vec{\alpha})) \\ &\geq \sum_{k=0}^{n} D[k, ext(\vec{\alpha})] \left(\min_p p^k (1-p)^{n-k} \right) \end{aligned}$$

Studied Constraints

SubSetFocus

Sequence

Among

GCC

AtMostNValue

AtLeastNValue

AtMostBalance

AtLeastBalance

Experiments

Num. of examples	Rank of initial constraint 1	2	3	∞	Num. of instances
1	8	46	1	545	600
2	42	119	0	439	600
3	78	148	0	374	600
4	105	172	0	323	600
5	139	170	0	291	600
10	261	117	0	222	600

Table 1: Results for SubsetFocus. Number of instances for which the initial constraint was ranked first, second, third or was not found.

Conclusion

- We were able to make a recommander system that helps experts to determine the parameters of certain constraints.

Conclusion

- We were able to make a recommander system that helps experts to determine the parameters of certain constraints.
- The system is not used!

Conclusion

- We were able to make a recommander system that helps experts to determine the parameters of certain constraints.
- The system is not used!
- It could have saved hundreds of hours in expert time.