# The Asynchronous $t$-Step Approximation for Scheduling Batch Flow Systems

David Grimsman

June 16, 2016

Information & Decision Algorithms Laboratories

# Example: A Chemical Manufacturer



- Assume we have a request for
  - 1 batch of mixture A
  - 3 batches of mixture B
  - 1 batch of mixture C
- Each mixture recipe requires a certain amount of time at each workstation: 1-3.
- We want to complete the request as soon as possible.

How can we order the chemical processes to minimize time?

# Block Model

Recipe:

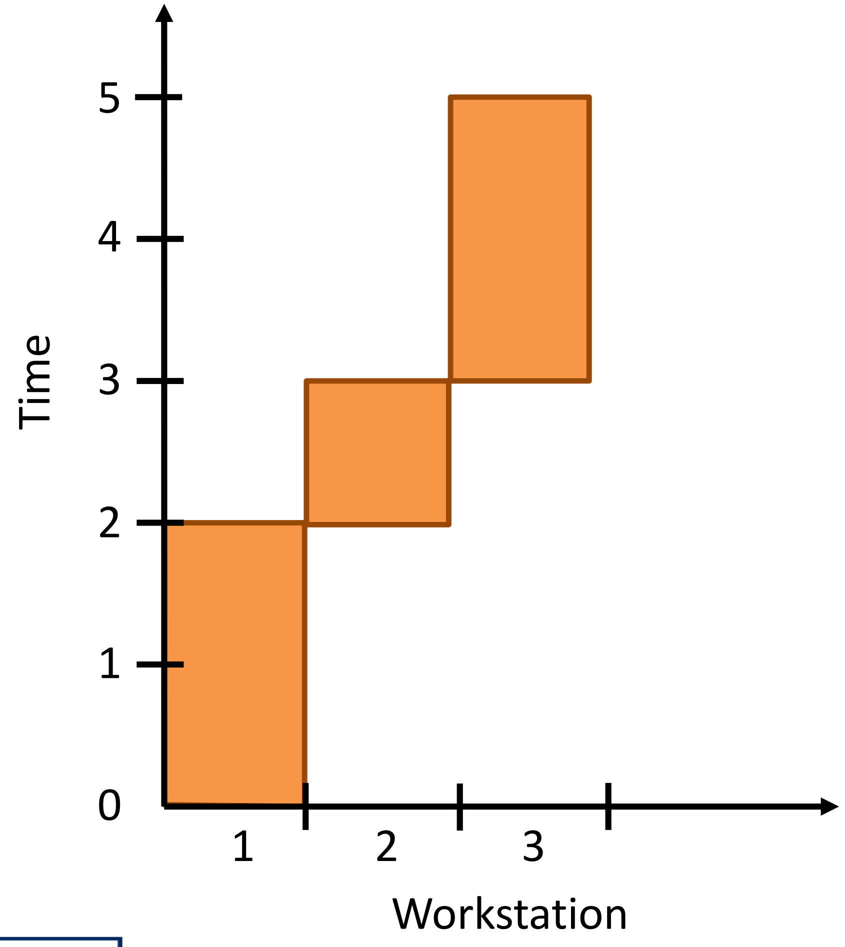
2 hours in workstation 1

1 hour in workstation 2

2 hours in workstation 3

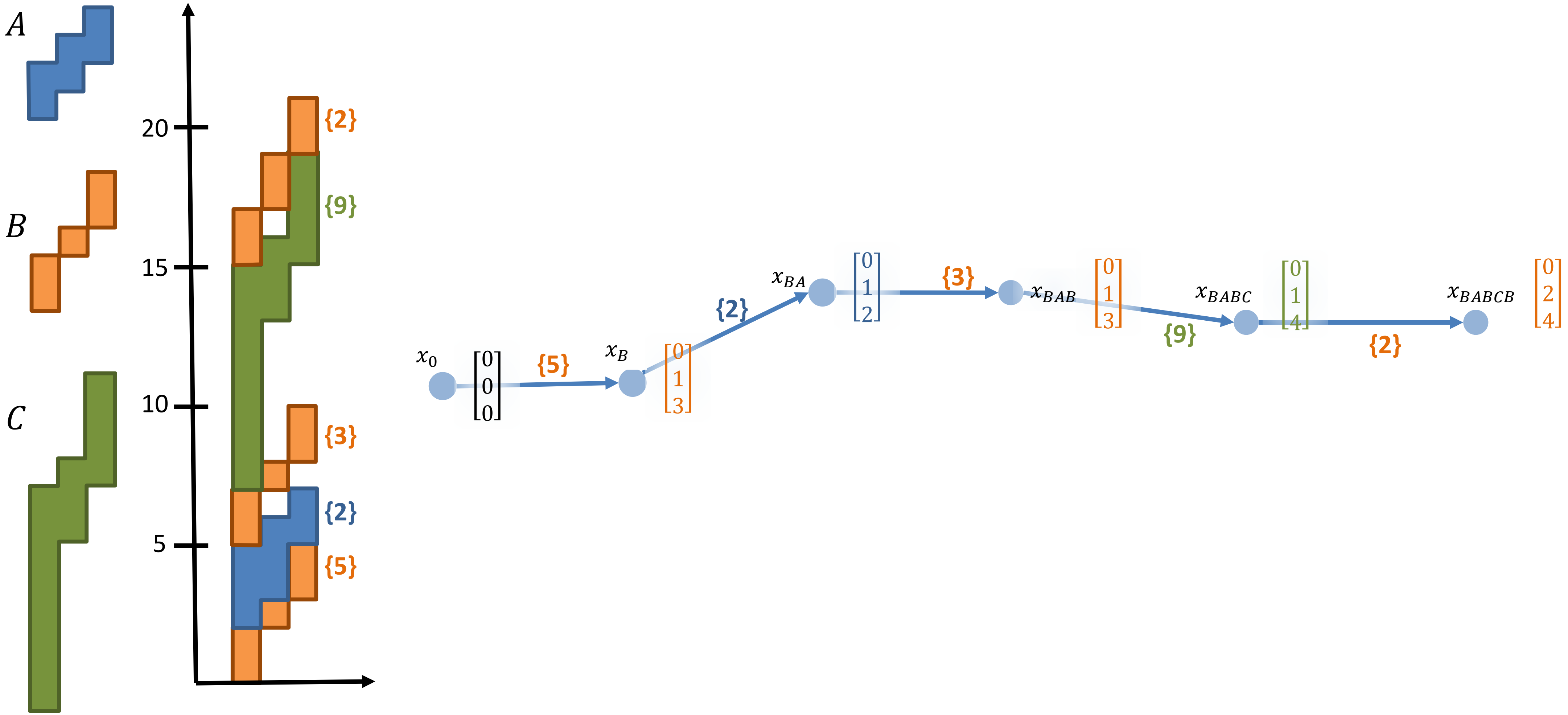
Each recipe defines a unique block

Upper contour: $\begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix}$

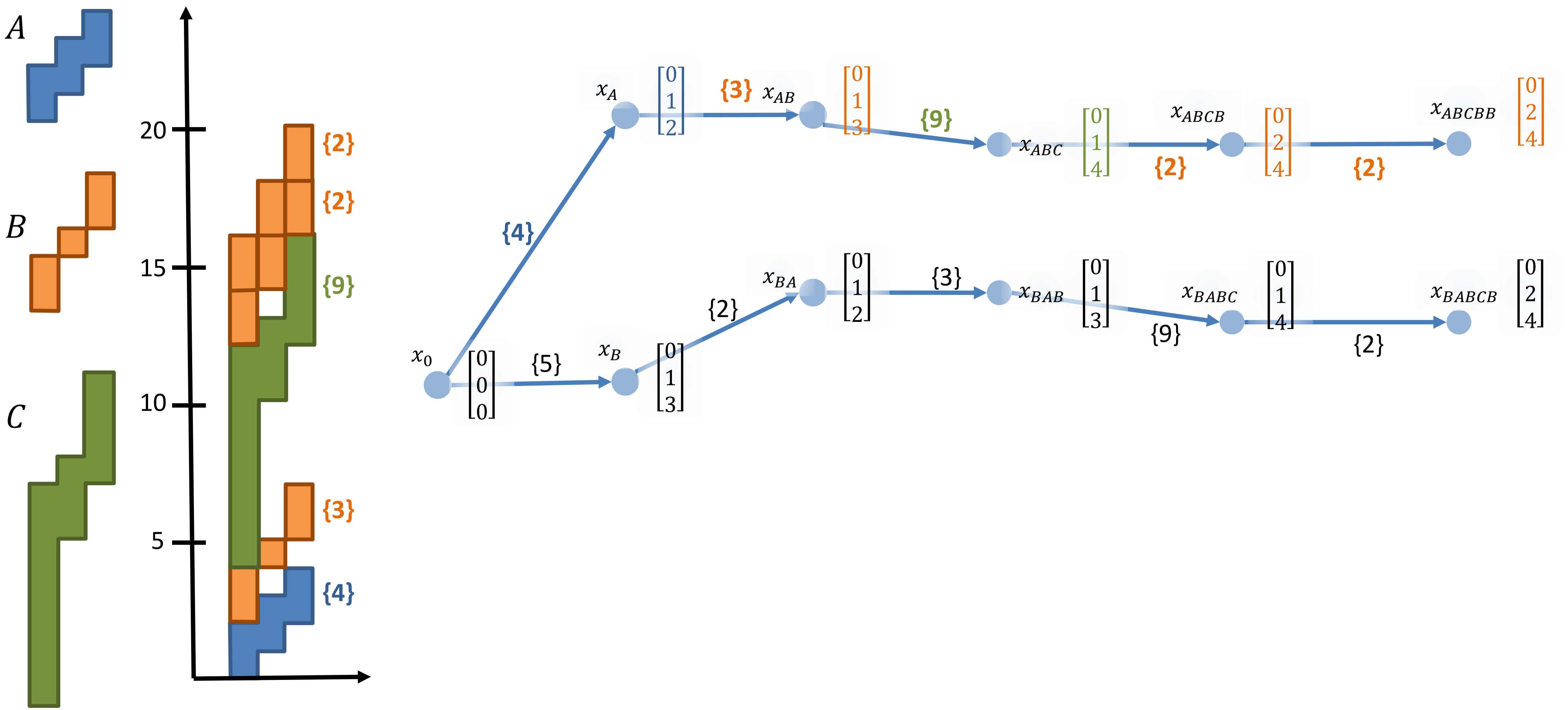


# Block Model



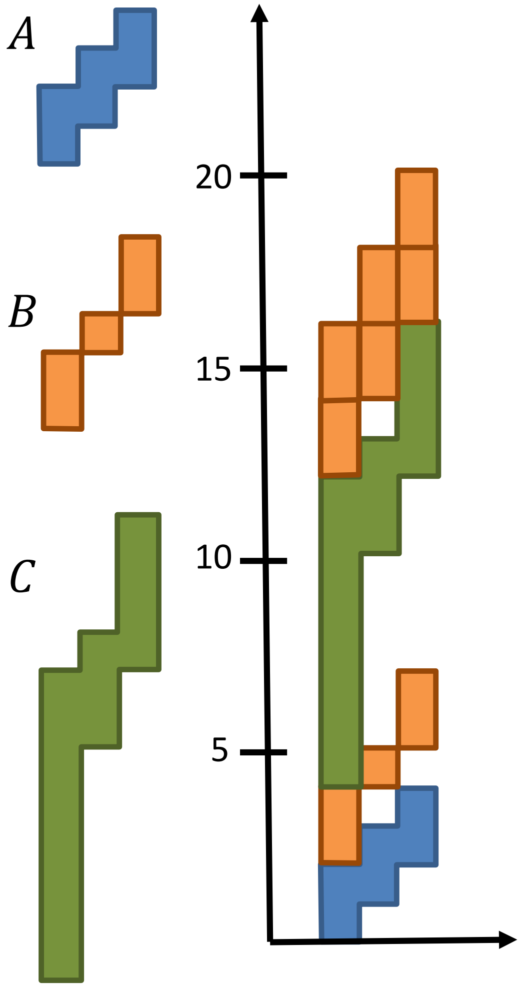

A
B
C
20
15
10
5
{2}
{9}
{3}
{2}
{5}
$x_0$
{5}
$x_B$
{2}
$x_{BA}$
{3}
$x_{BAB}$
{9}
$x_{BABC}$
{2}
$x_{BABCB}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\3\end{bmatrix}$
$\begin{bmatrix}0\\1\\2\end{bmatrix}$
$\begin{bmatrix}0\\1\\3\end{bmatrix}$
$\begin{bmatrix}0\\1\\4\end{bmatrix}$
$\begin{bmatrix}0\\2\\4\end{bmatrix}$

# Block Model

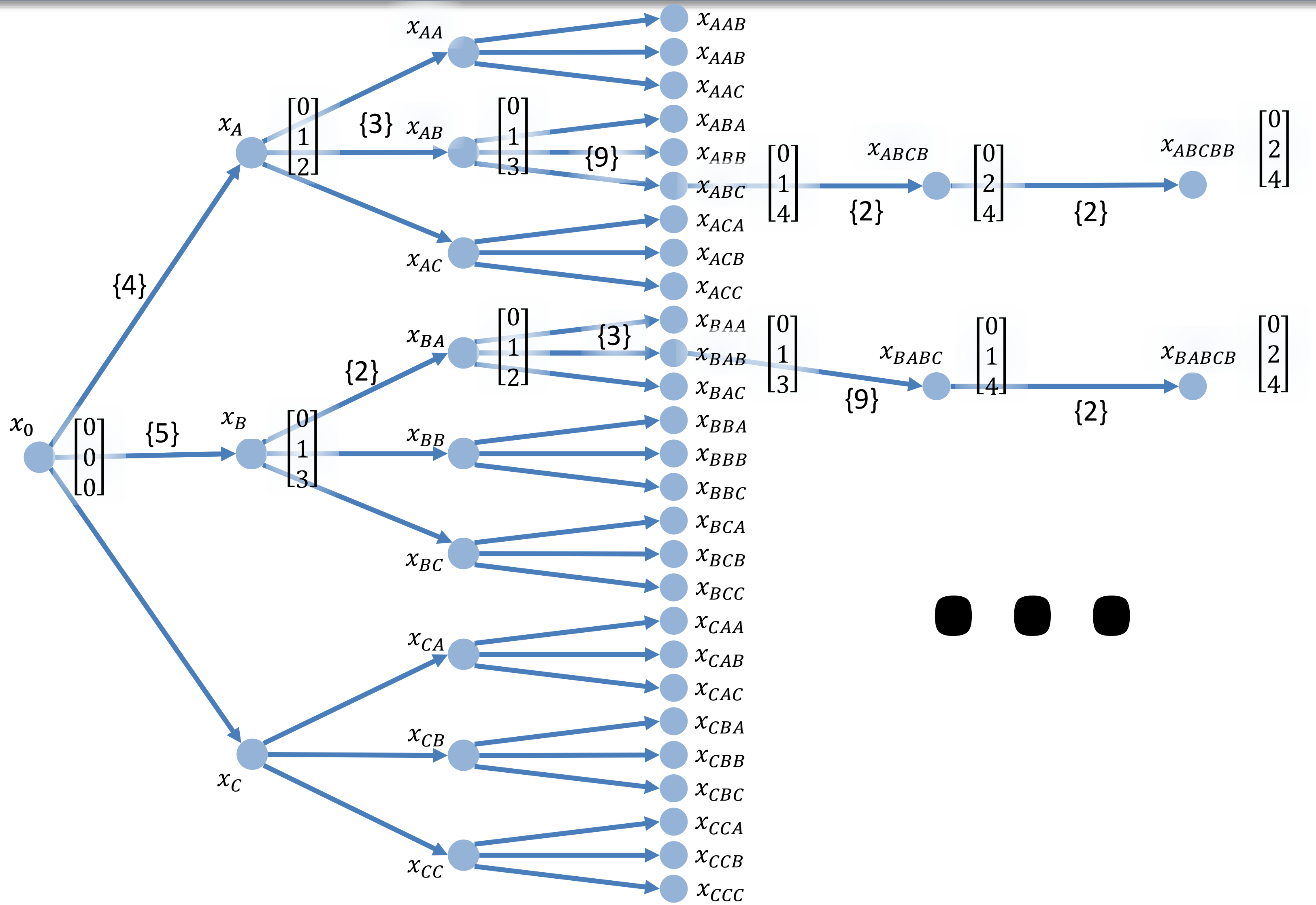

A
B
C
20
15
10
5
{2}
{2}
{9}
{3}
{4}
$x_0$
$x_A$
$x_B$
$x_{AB}$
$x_{BA}$
$x_{ABC}$
$x_{BAB}$
$x_{ABCB}$
$x_{BABC}$
$x_{ABCBB}$
$x_{BABCB}$
{4}
{5}
{3}
{2}
{9}
{3}
{2}
{9}
{2}
{2}
[0 0 0]
[0 1 2]
[0 1 3]
[0 1 3]
[0 1 2]
[0 1 4]
[0 1 3]
[0 2 4]
[0 1 4]
[0 2 4]
[0 2 4]

# Block Model





A
B
C
20
15
10
5

$x_0$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$
{4}
{5}
$x_A$
$\begin{bmatrix}0\\1\\2\end{bmatrix}$
$x_B$
$\begin{bmatrix}0\\1\\3\end{bmatrix}$
$x_C$
{3}
{2}
$x_{AA}$
$x_{AB}$
$\begin{bmatrix}0\\1\\3\end{bmatrix}$
{9}
$x_{AC}$
$x_{BA}$
$\begin{bmatrix}0\\1\\2\end{bmatrix}$
{3}
$x_{BB}$
$x_{BC}$
$x_{CA}$
$x_{CB}$
$x_{CC}$
$x_{AAB}$
$x_{AAB}$
$x_{AAC}$
$x_{ABA}$
$x_{ABB}$
$x_{ABC}$
$x_{ACA}$
$x_{ACB}$
$x_{ACC}$
$x_{BAA}$
$x_{BAB}$
$x_{BAC}$
$x_{BBA}$
$x_{BBB}$
$x_{BBC}$
$x_{BCA}$
$x_{BCB}$
$x_{BCC}$
$x_{CAA}$
$x_{CAB}$
$x_{CAC}$
$x_{CBA}$
$x_{CBB}$
$x_{CBC}$
$x_{CCA}$
$x_{CCB}$
$x_{CCC}$
$\begin{bmatrix}0\\1\\4\end{bmatrix}$
{2}
$x_{ABCB}$
$\begin{bmatrix}0\\2\\4\end{bmatrix}$
{2}
$x_{ABCBB}$
$\begin{bmatrix}0\\2\\4\end{bmatrix}$
$\begin{bmatrix}0\\1\\3\end{bmatrix}$
{9}
$x_{BABC}$
$\begin{bmatrix}0\\1\\4\end{bmatrix}$
{2}
$x_{BABCB}$
$\begin{bmatrix}0\\2\\4\end{bmatrix}$

# Problem 1

Using the following definitions

- **Items**: $n = \{n_1, \ldots, n_N\}$ be a set of blocks
- **Sequencing Rule**: A sequence $s$ of length $K$ is a stack of blocks created by stacking block $n_{i_1}$, followed by $n_{i_2}$, etc.
- **Sequence Measurement**: the measurement $H(s, x_0)$ is the height of stack $s$ with initial condition $x_0$

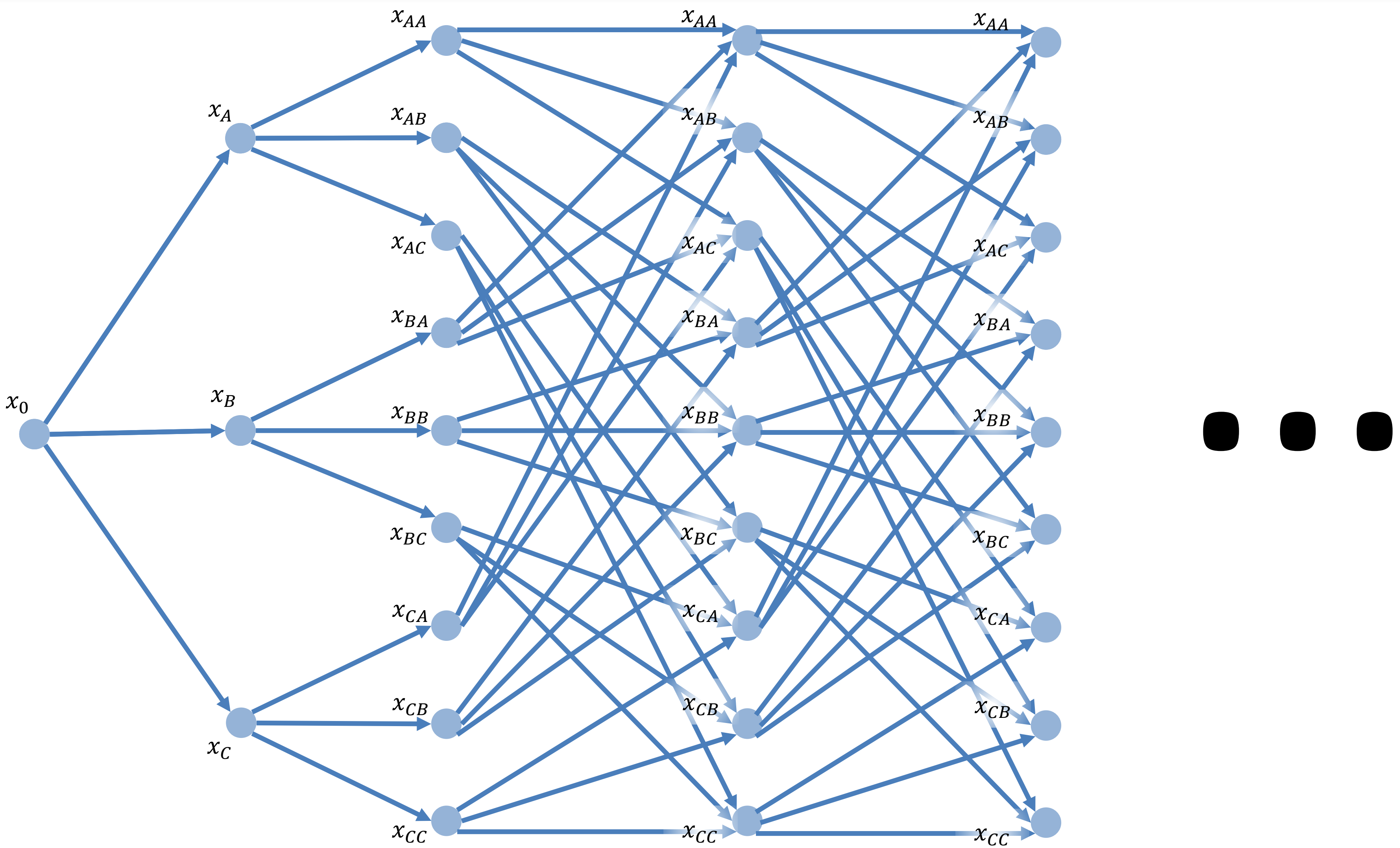
Let a quota $q = [q_1 \cdots q_N]$, where each $q_i$ represents how much of $n_i$ to make. Also, let the set $S_q = \{s | s \text{ is a sequence that satisfies } q\}$. Find

$$s^* = \arg\min_{s \in S_q} H(s, x_0)$$

# Symmetric $t$-Step Approximation (STA)



- A way to approximate systems in order to make Problem 1 less complex
- Given a $t$, assumes finding the current state is only a function of $u_{k-1}, \dots, u_{k-t}$
- As $t$ increases, error bound decreases monotonically



# Contributions of this work

- Determine which blocks are *rigid*
- Characterize sequences that never forget initial condition
- Formally present the ATA
- Prove that the ATA is always at least as good as the STA for the same complexity constraints
- Prove that the optimal path on the ATA has error bounds tighter than those for the optimal path on the STA

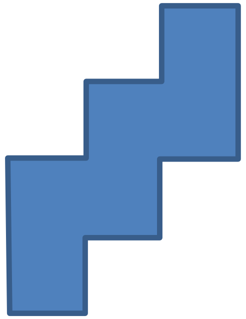
# Max Plus: Another Way to Model the System



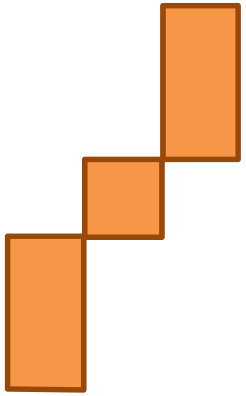
- Rules for max plus algebra
  - Plus: $a \oplus b = \max\{a, b\}$
  - Multiplication: $a \otimes b = a + b$
  - Zero: $\varepsilon = -\infty$
  - One: $e = 0$
- Others have modeled batch processes this way
- Our model is different because we limit the types of shapes available

# Each Shape is A Unique Max-Plus Matrix





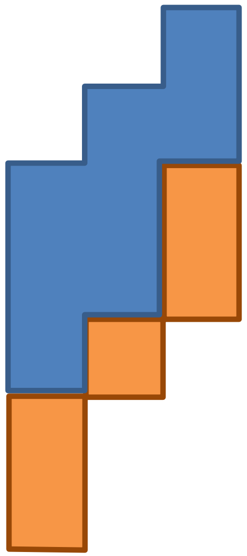
$$\Rightarrow u_A = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}, l_A = \begin{bmatrix} e \\ 1 \\ 2 \end{bmatrix} \Rightarrow M_A = \begin{bmatrix} 2 & 1 & e \\ 3 & 2 & 1 \\ 4 & 3 & 2 \end{bmatrix}$$



$$\Rightarrow u_B = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}, l_B = \begin{bmatrix} e \\ 2 \\ 3 \end{bmatrix} \Rightarrow M_B = \begin{bmatrix} 2 & e & \varepsilon \\ 3 & 1 & e \\ 5 & 3 & 2 \end{bmatrix}$$

# Each Shape is A Unique Max-Plus Matrix





$$x(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$x(1) = M_B \otimes x(0) = \begin{bmatrix} 2 & e & \varepsilon \\ 3 & 1 & e \\ 5 & 3 & 2 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$$

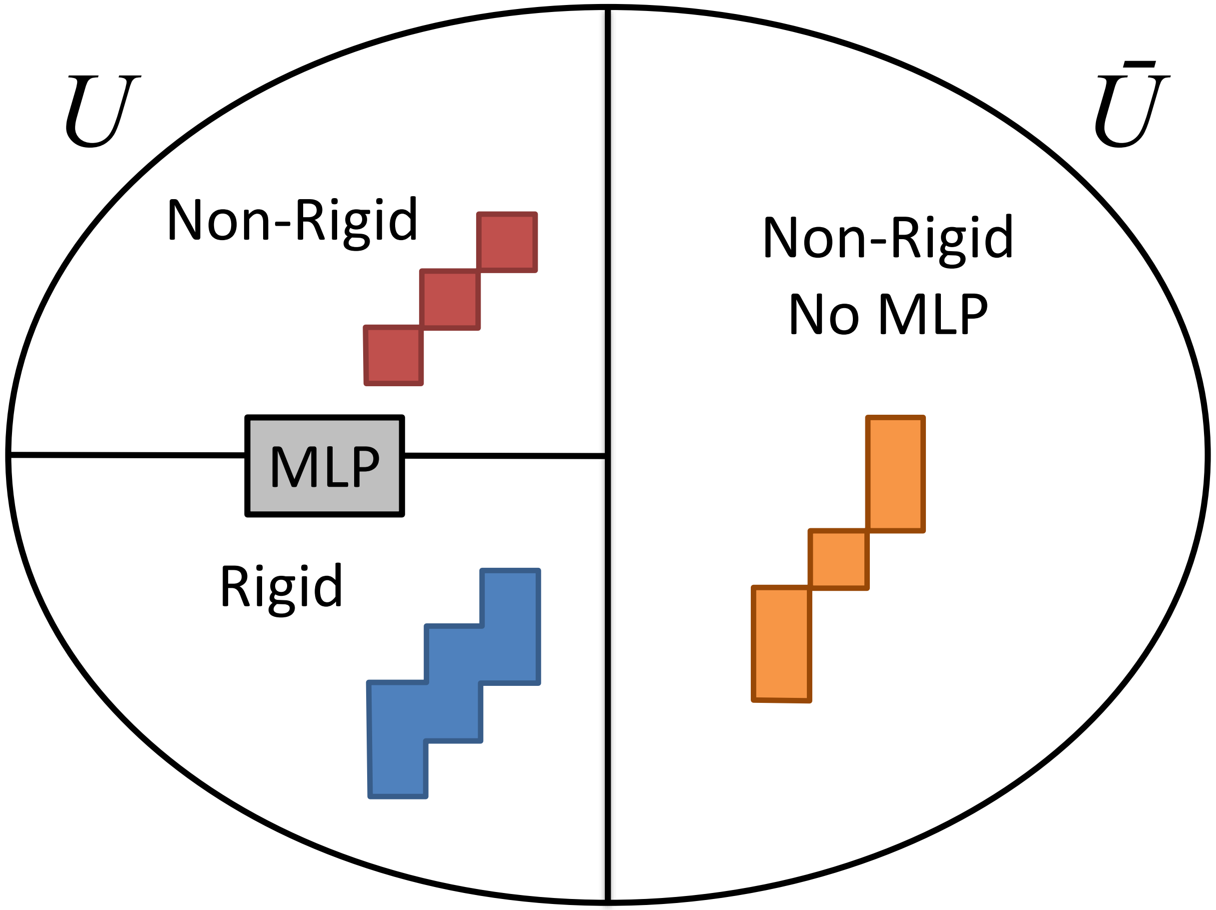
$$x(2) = M_A \otimes x(1) = \begin{bmatrix} 2 & 1 & e \\ 3 & 2 & 1 \\ 4 & 3 & 2 \end{bmatrix} \otimes \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \\ 7 \end{bmatrix}$$

Height of the stack

$$C\big(x(2)\big) = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

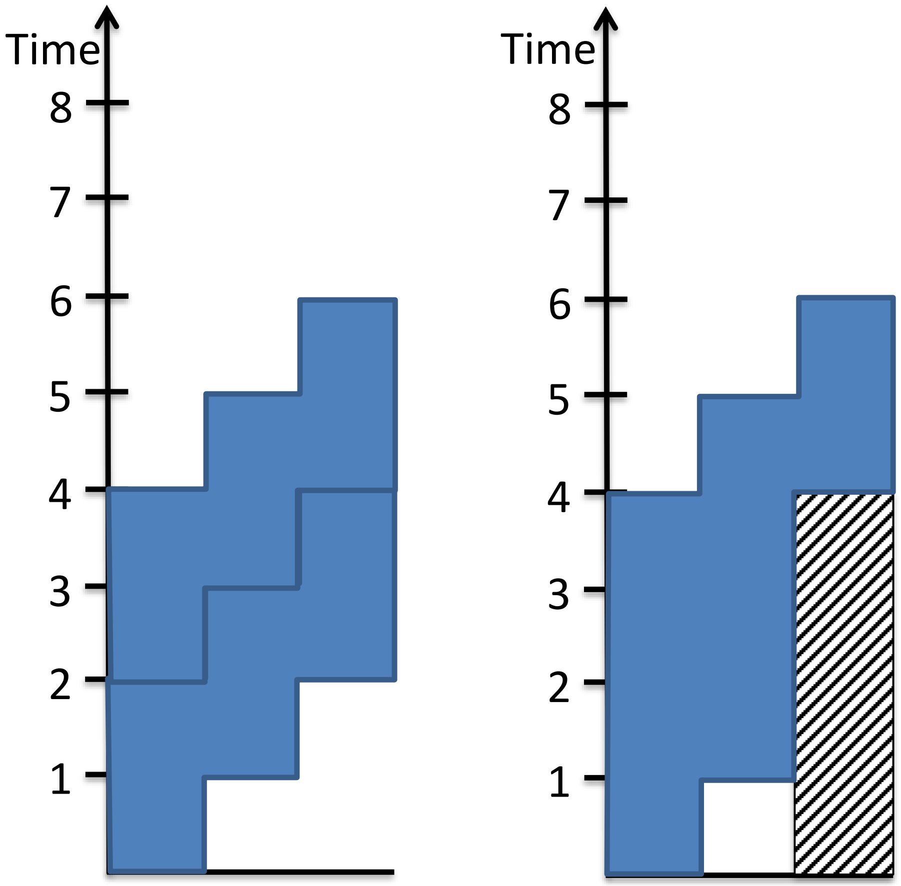
# Sets of Matrices





# Rigid Matrices

- Definition: $C(A \otimes x) = x_A \; \forall \, x.$
- Theorem: A matrix $A$ is rigid if and only if $A$ is rank one, in the max-plus sense
- No matter the initial condition, the upper contour is the same



# Rigid Matrices



**Proof**

$A$ is rigid $\Leftrightarrow$

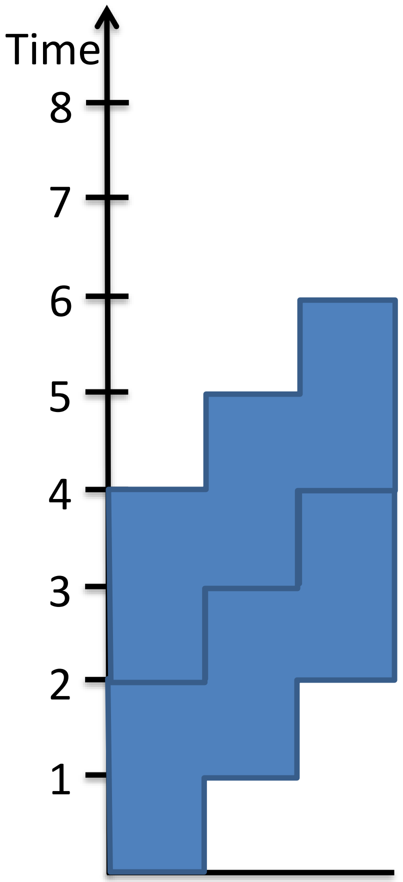
$C(A \otimes x) = x_A \; \forall \, x \;\; \Leftrightarrow$

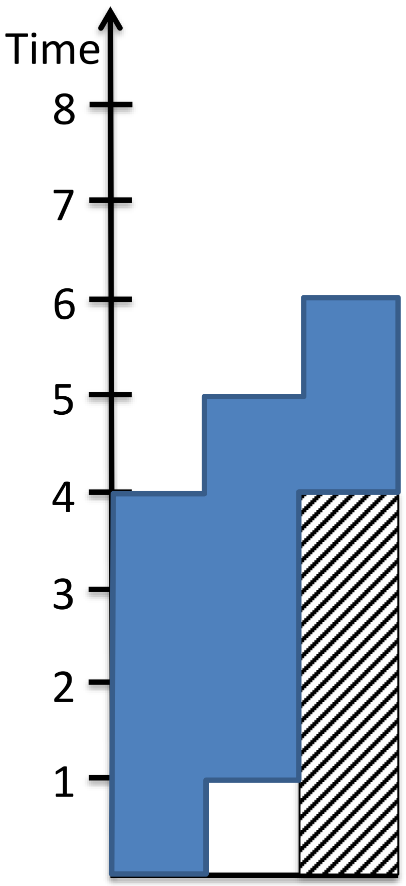
for every $x$ there exists a constant $c$ such that

$\mathrm{A} \otimes x = c \otimes x_A \; \forall \, x \Leftrightarrow$

$range(A)$ is dimension 1 $\Leftrightarrow$

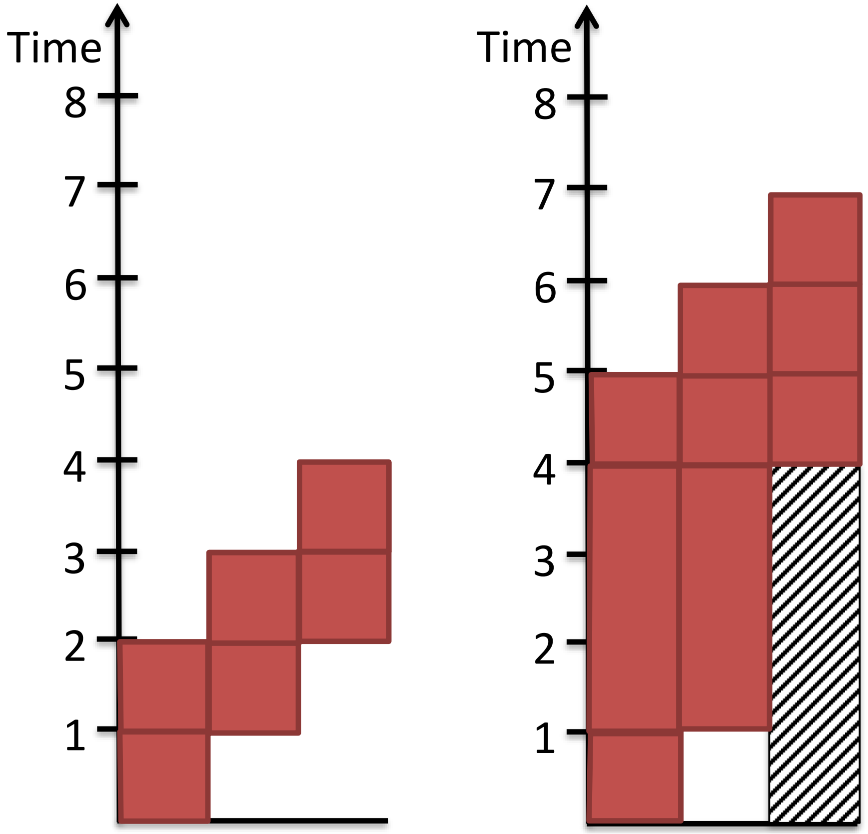
$A$ is rank 1



# Non-Rigid Matrices with the MLP

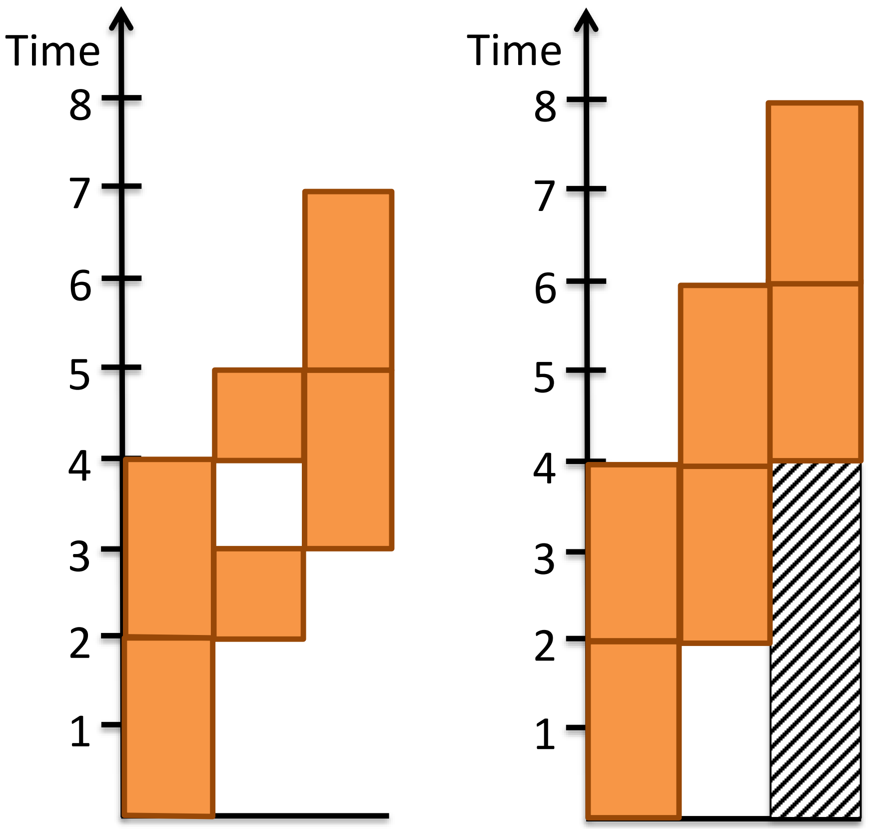





- Definition: $A$ is not rigid, but $A^{\otimes c}$ is a rigid matrix for some positive integer $c > 1$.
- A sequence of $A$ repeated $c$ times “forgets” its initial condition
- These matrices are dense in the space of all admissible matrices

# Matrices without the MLP

- Definition: $A^{\otimes c}$ is not rigid for all $c \geq 1$
- Theorem: $A$ does not have the MLP if and only if the dimension of the eigenspace of $A$ is greater than 1
- This can be determined by testing whether the communication graph of $A$ has more than one *maximally strongly connected subgraph*
- The sequence of $A$ repeated never "forgets" its initial condition



# Matrices without the MLP

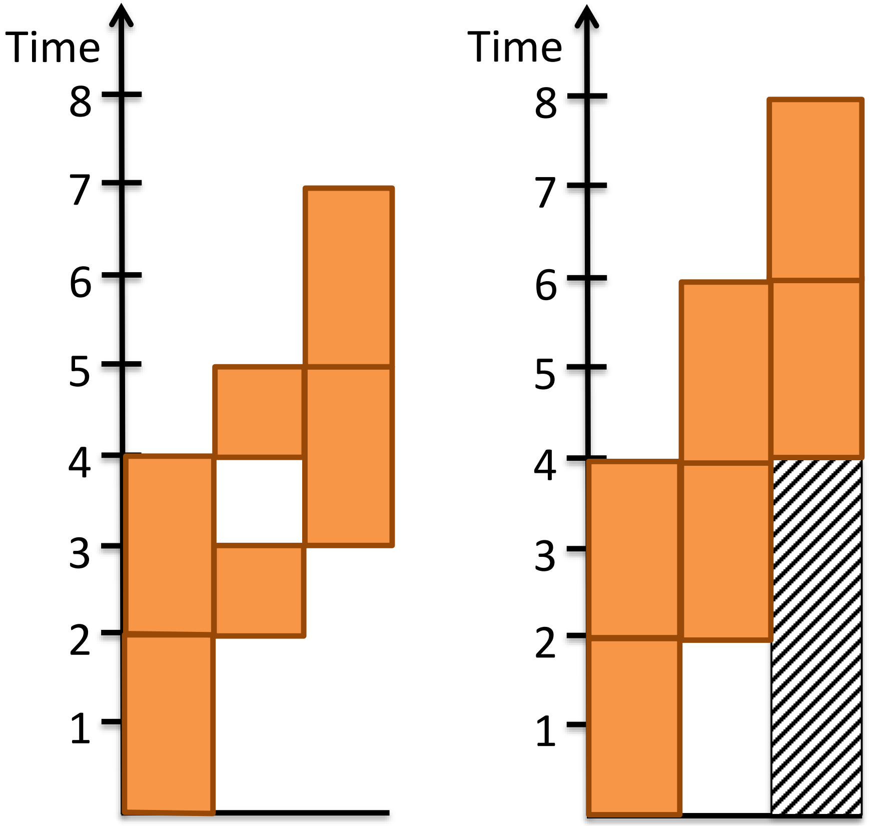


**Proof idea**

$A \otimes x = \lambda \otimes x \Rightarrow \lambda$ is an eigenvalue of $A$ and $x$ is an eigenvector of $A$.

When the system $x_{k+1} = A \otimes x_k$ becomes periodic, then $x_{k+1} = \lambda \otimes x_k$, and $x_k$ is an eigenvector of $A$.

If $x_k$ can have multiple values for a given $A$, then the system is always dependent on the initial condition.



# Takeaways



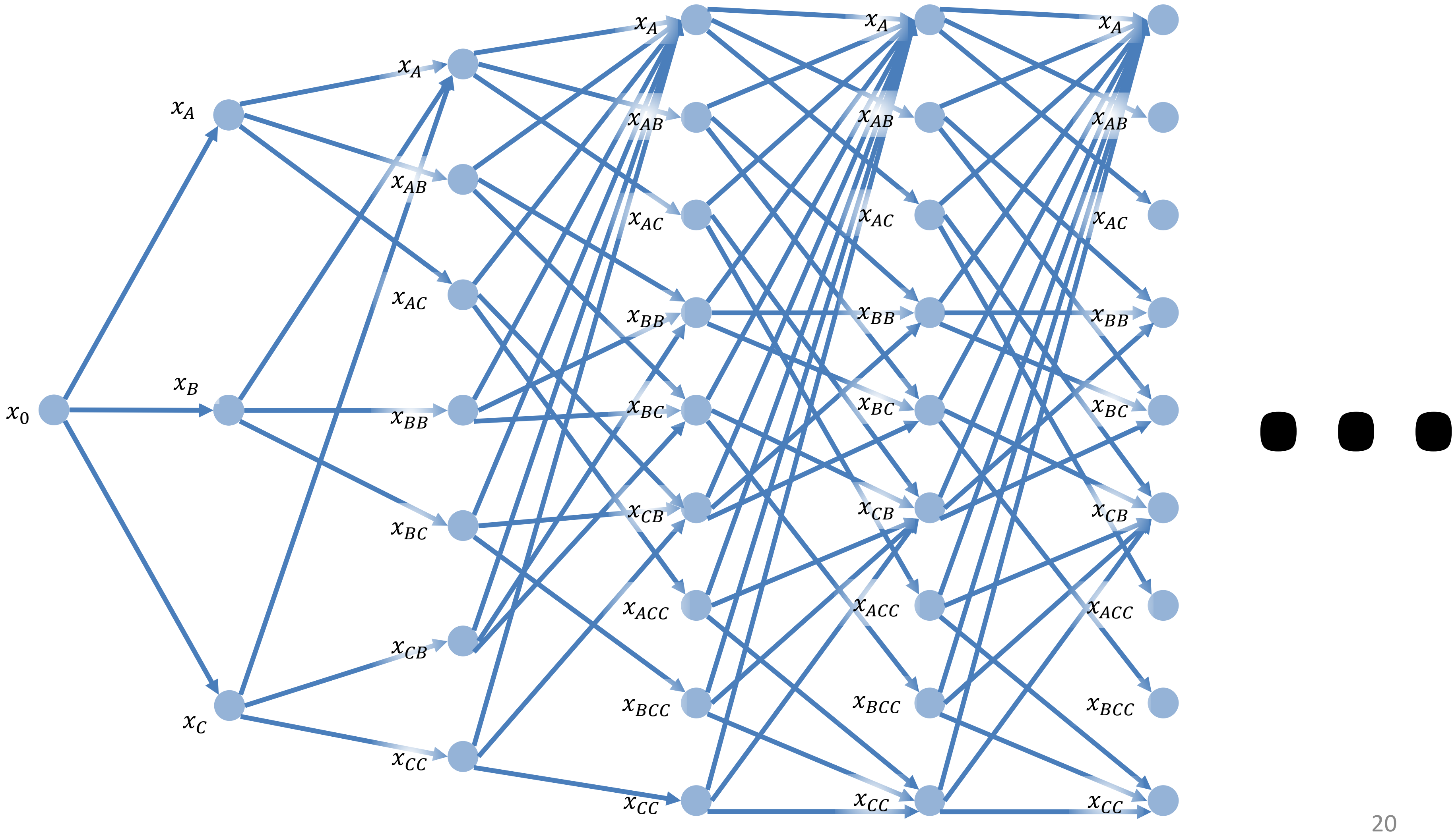
A fast test can be applied to evaluate whether a matrix is rigid

Sequences exist that never forget their initial condition

Both of these groups are sparse in the space of admissible matrices

# Asymmetric $t$-Step Approximation (ATA)

- Takes advantage of blocks or sequences of blocks that have memory loss
- With the same resources, an equal or better system approximation can be created
- Requires study of these systems and block properties



# Problem 2

Using the following definitions

- **Items**: $n = \{n_1, \dots, n_N\}$ be a set of blocks
- **Sequencing Rule**: A sequence $s$ of length $K$ is a stack of blocks created by stacking block $n_{i_1}$, followed by $n_{i_2}$, etc.
- **Sequence Measurement**: the measurement $\widehat{H}(s, x_0)$ is the height of stack $s$ estimated using the ATA

Let a quota $q = [q_1 \cdots q_N]$, where each $q_i$ represents how much of $n_i$ to make. Also, let the set $S_q = \{s | s \text{ is a sequence that satisfies } q\}$. Find

$$\hat{s}^* = \arg\min_{s \in S_q} \widehat{H}(s, x_0)$$

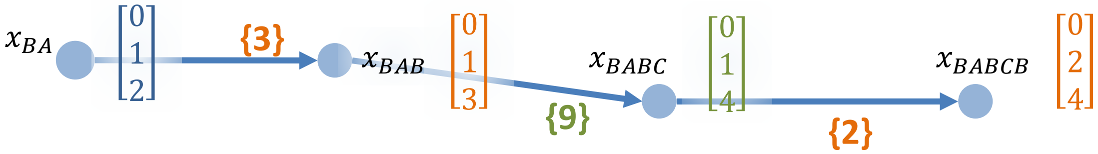
# System approximation

- Let
  - $\widehat{H}_{ATA}$ be the approximation to $H$ using the ATA
  - $\widehat{H}_{STA}$ be the approximation to $H$ using the STA
- Theorem: $H(s, x_0) \geq \widehat{H}_{ATA}(s, x_0) \geq \widehat{H}_{STA}(s, x_0) \; \forall \; s, x_0$
- This means the ATA is always a closer or equivalent approximation than STA

# Error Bounds

Let

- $s(t,k)$ be the subsequence of $s$ of length $t$ starting at time $k$
- $x_{min} = [\varepsilon \quad \cdots \quad \varepsilon \quad e]^T$
- $\gamma^t(s) = \max_x \{H(s(t,0),x) - H(s(t-1,0),x) - (H(s(t,0),x_{min}) - H(s(t-1,0),x_{min}))\}$
- $\Gamma^t = \max_{s \in S(t)} \gamma^t(s)$
- $\hat{t}_{STA}$ is the value of $t$ used in STA
- $\hat{t}_{ATA}$ is the maximum sequence length found in the ATA table






IDeA Labs
Information & Decision Algorithms Laboratories

# Error Bounds



- Already known that
  - $\Gamma^{t} \geq \Gamma^{t+1}\ \forall\ t$
  - $e_{STA} \leq \Gamma^{\hat{t}_{STA}}[\|q\|_1 - (\hat{t}_{STA} + 1)] = e_{STA}^{max}$
- Theorem:
  - $e_{ATA} \leq \Gamma^{\hat{t}_{ATA}}(\|q\|_1 - \hat{t}_{ATA}) = e_{ATA}^{max}$
  - $e_{ATA}^{max} \leq e_{STA}^{max}$

Key:

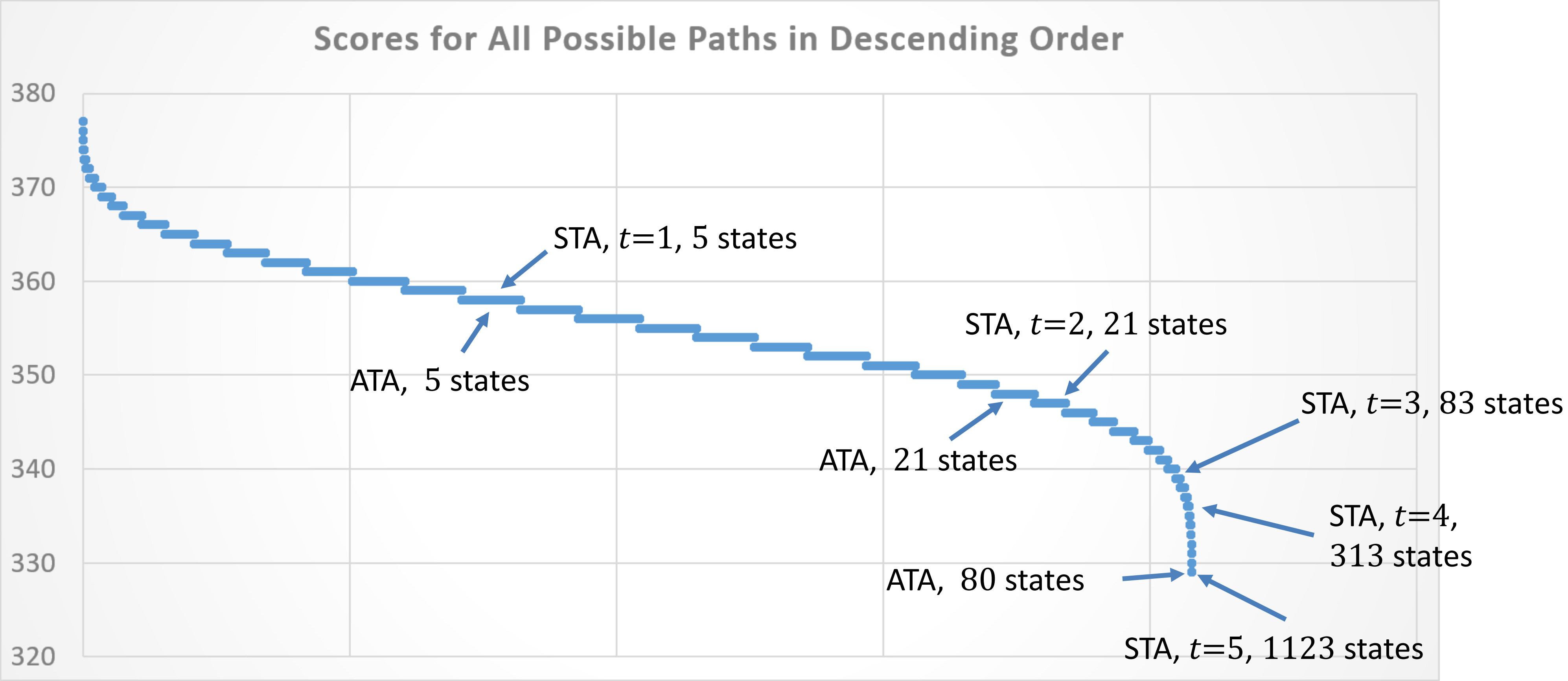
$\hat{t}_{STA} \leq \hat{t}_{ATA}$



IDeA Labs
Information & Decision Algorithms Laboratories

# Example System








Information & Decision Algorithms Laboratories

# Contributions of this work

- Determine which blocks are *rigid*
- Characterize sequences that never forget initial condition
- Formally present the ATA
- Prove that the ATA is always at least as good as the STA for the same complexity constraints
- Prove that the optimal path on the ATA has error bounds tighter than those for the optimal path on the STA

# Future Work



How can we generalize this to more systems?

- What does *rigid* mean in other contexts?
- Is there a way to classify systems where the ATA is applicable?
- What other applications are relevant today?

Is there a method that works best for solving the shortest path problem in the ATA?


Is there a better method for solving outside of using an approximation?



IDeA Labs
Information & Decision Algorithms Laboratories

# Thank You

IDeA Labs

Information & Decision Algorithms Laboratories