



**HAL**  
open science

## Set-Membership Method for Discrete Optimal Control

Rémy Guyonneau, Sébastien Lagrange, Laurent Hardouin, Mehdi Lhommeau

► **To cite this version:**

Rémy Guyonneau, Sébastien Lagrange, Laurent Hardouin, Mehdi Lhommeau. Set-Membership Method for Discrete Optimal Control. 10th International Conference on Informatics in Control, Automation and Robotics, ICINCO, 2013, Reykjavik, Iceland. 2013, 10.5220/0004458001930200 . hal-01113457

**HAL Id: hal-01113457**

**<https://hal.science/hal-01113457>**

Submitted on 5 Feb 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Set-Membership Method for Discrete Optimal Control



## 1. Problem

### Considered System

⇒ We consider a control system, defined by the differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (1)$$

$\mathbf{x}(t) \in \mathbb{R}^n$  the state vector

$\mathbf{u}(t) \in \mathbf{U}$  the control vector

⇒ The system is studied over  $[t_0, t_f]$

$$t_k = t_0 + k \times \delta_t, t_k \leq t_f, k \in \{1, \dots, m\} \quad (2)$$

It is assumed that  $\mathbf{u}(t_k)$  is bounded over  $[t_k, t_{k+1}]$  so it is possible to determinate a box  $[\mathbf{u}_k]$  such that  $\mathbf{u}(t_k) \in [\mathbf{u}_k]$  over  $[t_k, t_{k+1}]$

⇒ The flow map of the system is defined as

$$\varphi(t_0, t_k; \mathbf{x}_0, \mathbf{u}(t)) = \mathbf{x}(t) \quad (3)$$

⇒ The reachable set of the system at time  $t_k$  is

$$\begin{aligned} \varphi(t_0, t_k; \mathbf{X}_0, \mathbf{U}) = \{ & \varphi(t_0, t_k; \mathbf{x}_0, \mathbf{u}(t)) \mid \varphi(t_0, t_0; \mathbf{x}_0, \mathbf{u}(t)) = \mathbf{x}_0 \\ & \text{and } \varphi : [t_0, t_k] \times \mathbf{X}_0 \times \mathbf{U} \rightarrow \mathbb{R}^n \text{ is a} \\ & \text{solution of (1) for some } \mathbf{u}(t) \in \mathcal{U} \} \end{aligned} \quad (4)$$

where  $\mathcal{U} = \{ \mathbf{u} : [t_0, t_{k-1}] \rightarrow \mathbf{U} \mid \mathbf{u} \text{ is continuous over } [t_k, t_{k+1}] \}$  denotes the set of admissible controls and  $\mathbf{X}_0$  a set of possible initial values  $\mathbf{x}_0$

### Objective

⇒ Evaluate  $\mathbf{C}_{t_0, t_f}$  the subset of initial states of  $\mathbf{K}$  (state constraint) from which there exists at least one solution of (1) reaching the target  $\mathbf{T}$  in finite time  $t_f$  starting at a time  $t_0$ :

$$\mathbf{C}_{t_0, t_f} = \{ \mathbf{x}_0 \in \mathbf{K} \mid \exists \mathbf{u}(t) \in \mathcal{U}, \varphi(t_0, t_f; \mathbf{x}_0, \mathbf{u}(t)) \in \mathbf{T} \} \quad (5)$$

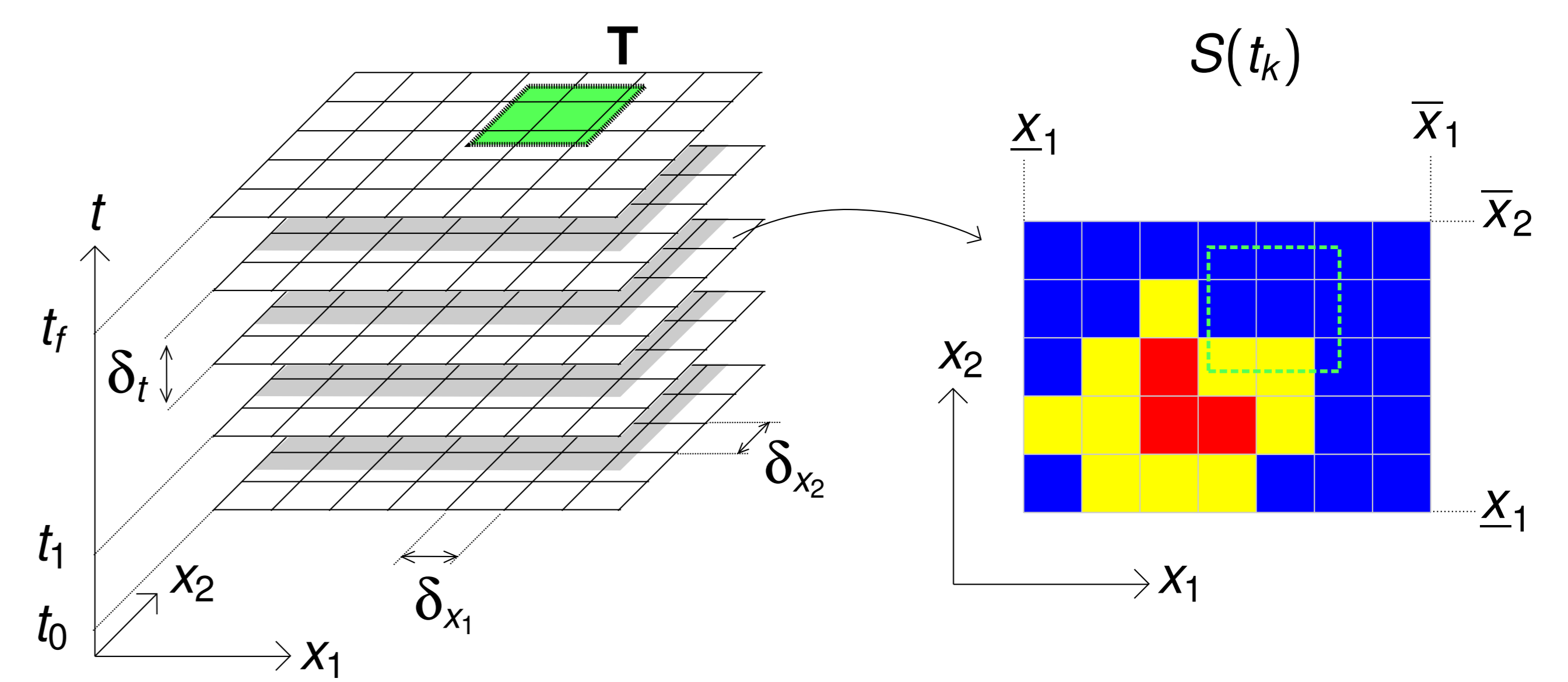
⇒ Using interval analysis to compute an inner and an outer characterisations of  $\mathbf{C}_{t_0, t_f}$

$$\mathbf{C}_{t_0, t_f}^- \subseteq \mathbf{C}_{t_0, t_f} \subseteq \mathbf{C}_{t_0, t_f}^+ \quad (6)$$

## 2. Characterisation computation

### Proposed approach

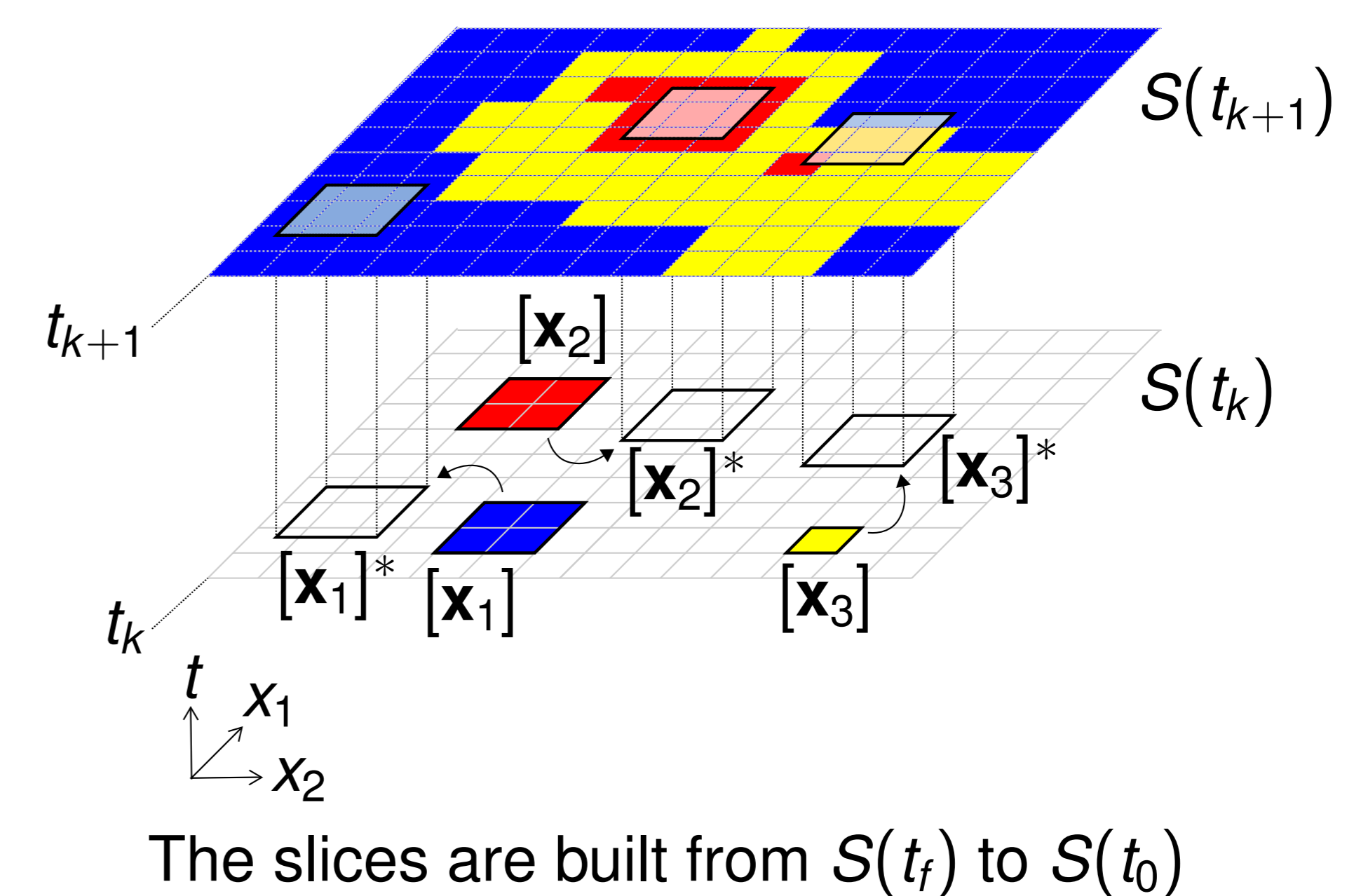
For each time  $t_k$  the algorithm computes a gridding of  $\mathbf{K}$  (a slice), noted  $S(t_k)$ . The resolution of the gridding is  $\delta_{\mathbf{K}} = (\delta_{x_1}, \dots, \delta_{x_i}, \dots, \delta_{x_n})$  where  $\delta_{x_i}$  corresponds to the  $i^{\text{th}}$  dimension of  $\mathbf{K}$



### Slice computation

We propose an iterative algorithm that classifies the cells of each slice in three categories:

- *unreachable* (blue), no state inside the cell allows the system to reach the target at time  $t_f$
- *reachable* (red), all the states inside the cell allow the system to reach the target at time  $t_f$
- *indeterminate* (yellow), neither *reachable* nor *unreachable*

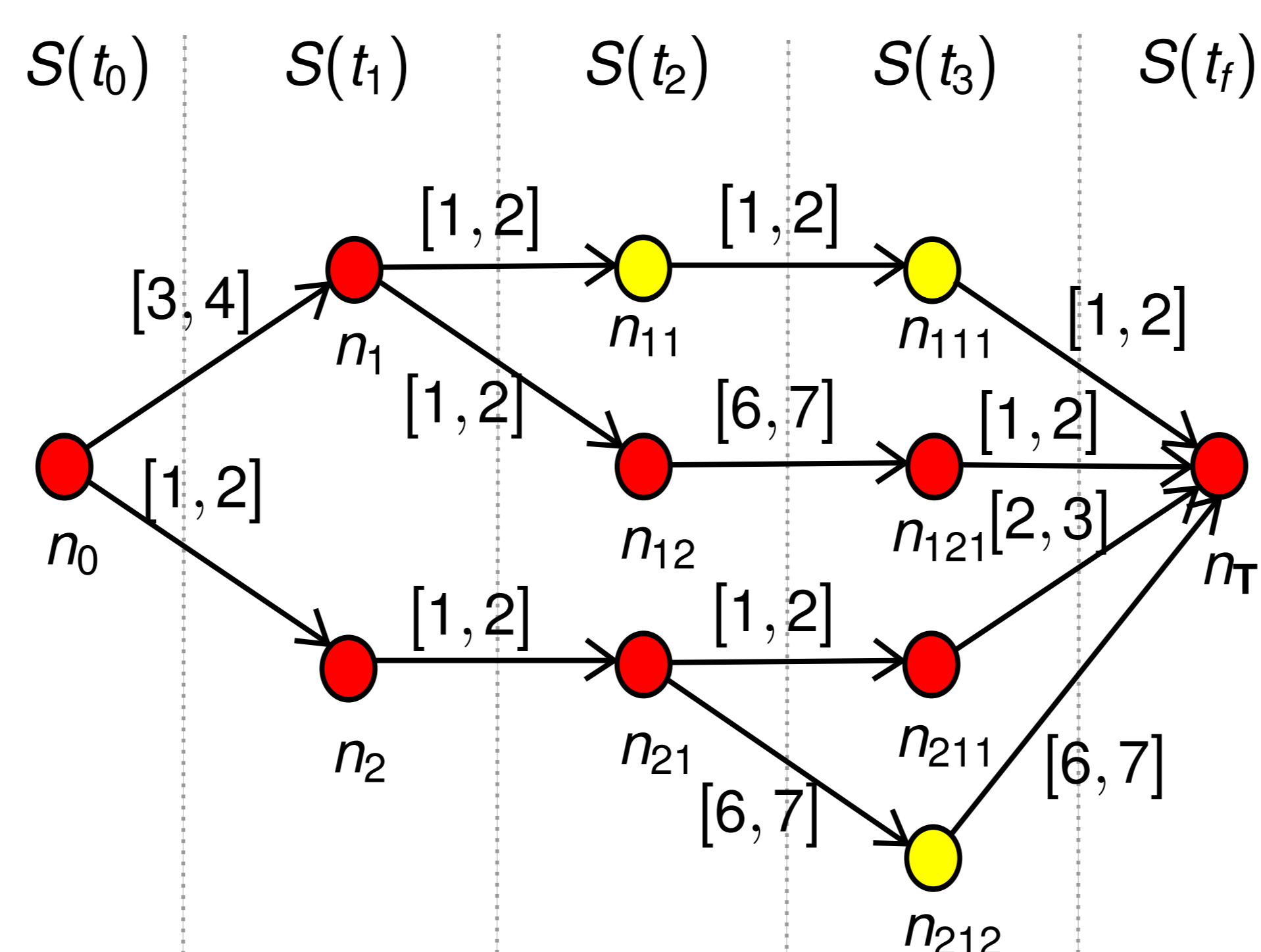
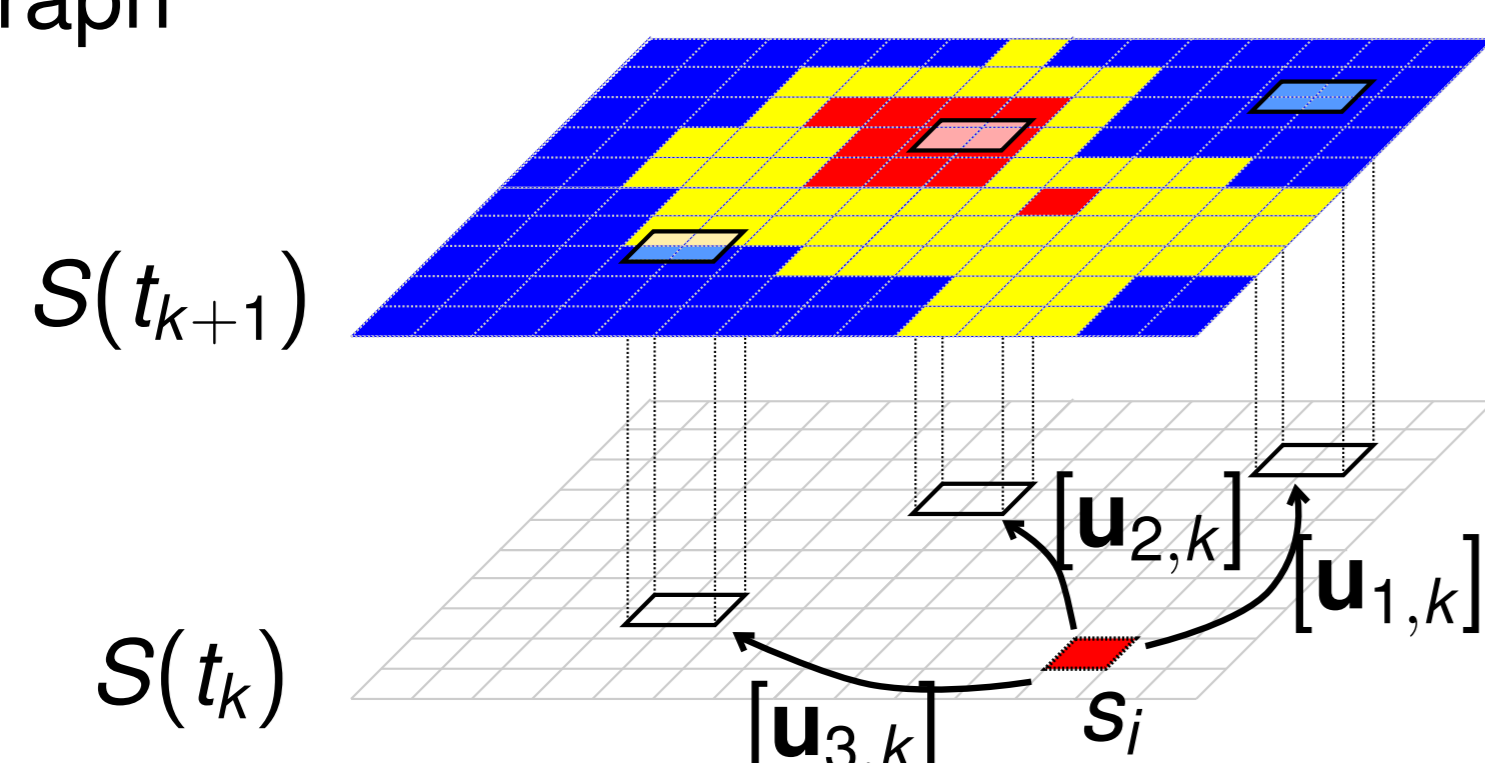


## 3. Optimal discrete path evaluation

### Slice modification and graph building

⇒ For each cell  $s_i \in S(t_k)$  is defined a set of input vectors  $\mathbf{U}(s_i)$  that leads  $s_i$  to reachable or indeterminate cells of  $S(t_{k+1})$

⇒ Gather the cells into nodes and build a graph



$$\begin{aligned} J(P(n_0, n_1, n_{11}, n_{111}, n_T)) &= [6, 10] \\ J(P(n_0, n_1, n_{12}, n_{121}, n_T)) &= [11, 15] \\ J(P(n_0, n_2, n_{21}, n_{211}, n_T)) &= [5, 9] \\ J(P(n_0, n_2, n_{21}, n_{212}, n_T)) &= [14, 18] \end{aligned}$$

### Obtained paths

⇒ Using the graph and a shortest path algorithm (e.g. Interval Dijkstra) it is possible to compute:

- an enclosure ( $\mathbf{P}^*$ ) of the optimal discrete control vector to reach the target from an initial state  $[\mathbf{x}_0] \in \mathbf{C}_{t_0, t_f}$
- an evaluation of the cost ( $J(\mathbf{P}^*)$ ) of this control vector

⇒ For instance

$$\begin{aligned} \mathbf{P}^* &= \{ P(n_0, n_1, n_{11}, n_{111}, n_T), \\ & P(n_0, n_2, n_{21}, n_{211}, n_T) \} \\ J(\mathbf{P}^*) &= [6, 10] \cup [5, 9] = [5, 9] \end{aligned}$$