UNIVERSITY OF TWENTE.

EXAMPLE: ADP FOR PATIENT ADMISSION PLANNING

Martijn Mes

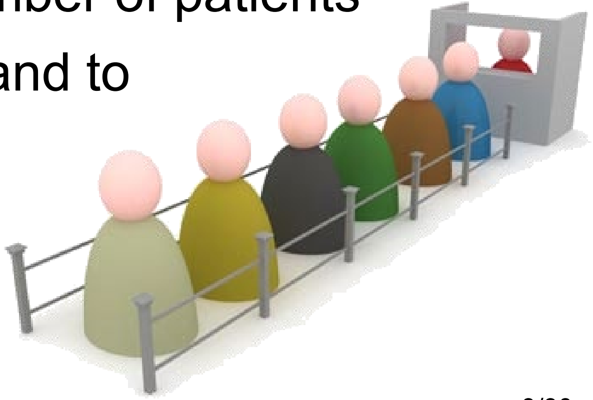
Universiteit Twente

CASE DESCRIPTION

- Motivation
  - Long access times due to lacking match of supply and demand
  - Limited control of serving the strategically agreed number of patients
  - Opportunities to improve resource utilization
- Our focus: integrated decision making on the tactical planning level:
  - Patient care processes connect multiple departments and resources, which require an integrated approach.
  - Operational decisions often depend on a tactical plan, e.g., tactical allocation of blocks of resource time to specialties and/or patient categories (master schedule / block plan).
- Care process: a chain of care stages for a patient, e.g., consultation, surgery, or a visit to the outpatient clinic

OBJECTIVES

- To control access times and care pathway durations
  - To ensure quality of care for the patient and to prevent patients from seeking treatment elsewhere
  - Decreasing care pathway duration decreases the delay between costs invested and revenues incurred
- To serve the strategically agreed number of patients
- To achieve high resource utilization and to balance workload
- Decreased costs and increased staff satisfaction



TACTICAL PLANNING IN OUR STUDY

- Typical setting: *8* care processes, *8* weeks as a planning horizon, and *4* resource types.
- Current way of creating/adjusting tactical plans: biweekly meeting with decision makers using spreadsheet solutions.
- Our objective: to provide an optimization step that supports rational decision making in tactical planning.

		Patient admission plan						
Care pathway	Stage	*Week 1*	*Week 2*	*Week 3*	*Week 4*	*Week 5*	*Week 6*	*Week 7*
Knee	1. Consultation	5	10	12	11	9	5	4
Knee	2. Surgery	6	5	10	12	11	9	5
Hip	1. Consultation	2	7	4	6	8	3	2
Hip	2. Surgery	10	0	7	0	1	4	4
Shoulder	1. Consultation	4	9	7	8	8	9	3
Shoulder	2. Surgery	2	8	5	5	7	10	8

PROBLEM FORMULATION [1/3]

- Discretized finite planning horizon $t \in \{1,2,\ldots,T\}$
- Patients:
  - Set of patient care processes $g \in \{1,2,\ldots,G\}$
  - Each care process consists of a set of stages $\{1,2,\ldots,e_g\}$
  - A patient following care process g follows the stages $K_g = \{(g,1),(g,2),\ldots,(g,e_g)\}$
- Resources:
  - Set of resource types $r \in \{1,2,\ldots,R\}$
  - Resource capacities $\eta_{r,t}$ per resource type and time period
  - To service a patient in stage $j = (g,a)$ of care process g requires $s_{j,r}$ of resource r

PROBLEM FORMULATION [2/3]

- From now on, we denote each stage in a care process by a queue j.
- State: $S_t = (S_{t,j,u}) \forall j, u$ gives the number of patients waiting u time units in queue j, $\forall j, u$.
- After service in queue i, we have a probability $q_{i,j}$ that the patient is transferred to queue j.
- Probability to leave the system: $q_{i,0} = 1 - \sum_{\forall j} q_{i,j}$
- Newly arriving patients joining queue i: $\lambda_{i,t}$
- Waiting list: $S_{t,j} = \left(S_{t,j,0}, S_{t,j,1}, \dots\right)$.
- Decision: for each time period, we determine a patient admission plan: $x_{t,j} = \left(x_{t,j,0}, x_{t,j,1}, \dots\right)$, where $x_{t,j,u}$ indicates the number of patients to serve in time period t that have been waiting precisely u time periods at queue j.

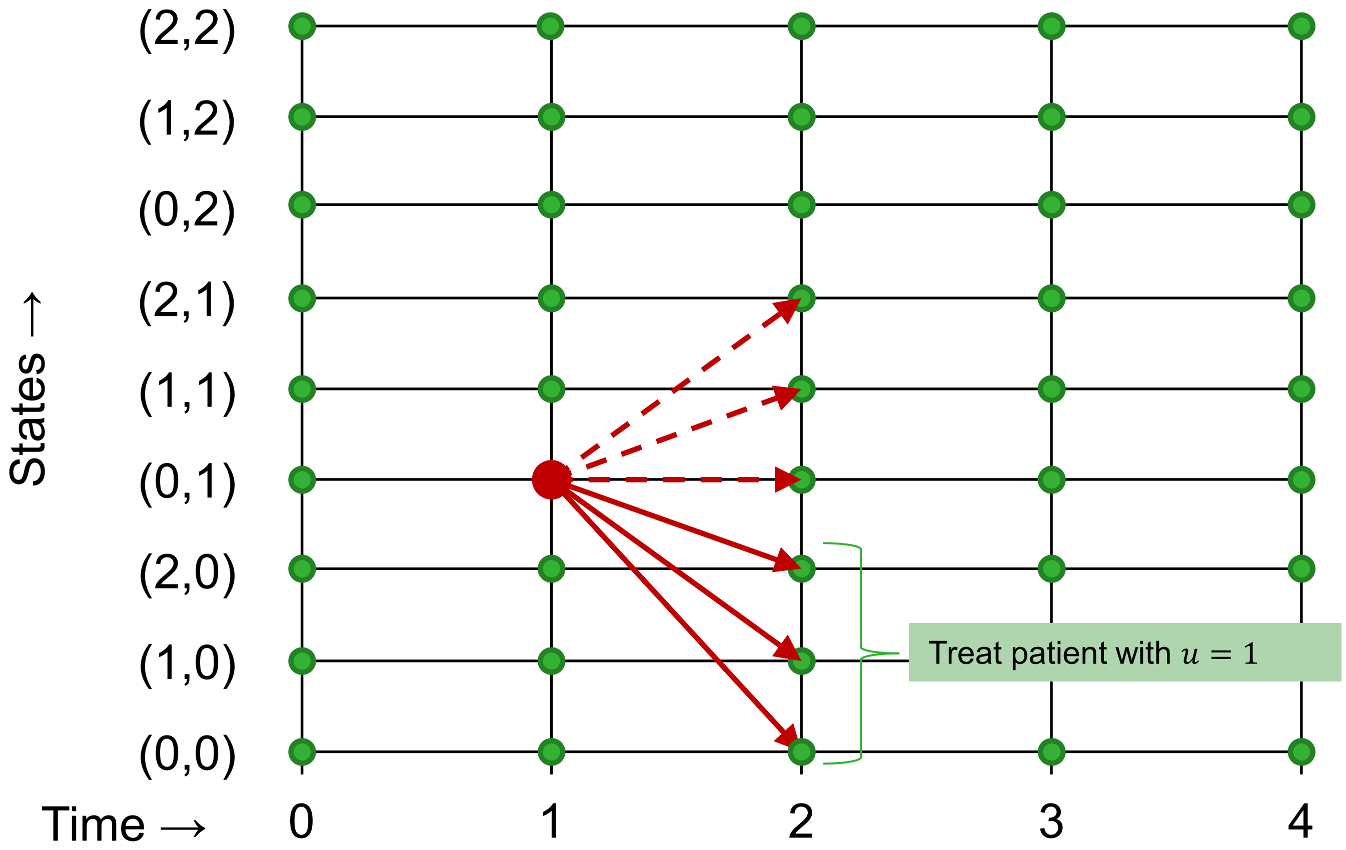
PROBLEM FORMULATION [3/3]

- Time lag $d_{i,j}$ between service in i and entrance to j (might be medically required to recover from a procedure).
- Patients entering queue j:

$$S_{t,j,0} = \lambda_{j,t} + \sum_{\forall i} \sum_{\forall u} q_{i,j} x_{t-d_{i,j},i,u}$$

- Assume: we have an upper bound U for u, longer waiting times are just counted as u.
- Temporarily assume: patient arrivals, patient transfers, resource requirements, and resource capacities are deterministic and known.

ILLUSTRATION:
1 queue (1 care process with 1 stage), 0/1 waiting time



MIXED INTEGER LINEAR PROGRAM

Number of patients in queue *j* at time *t* with waiting time *u*

Number of patients to treat in queue *j* at time *t* with a waiting time *u*

$$\min \sum_{t\in\mathcal{T}} C_t\left(S_t, x_t\right) = \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{J}} \sum_{u\in\mathcal{U}} c_{j,u}\left(S_{t,j,u} - x_{t,j,u}\right), \quad [1]$$

subject to

Assume time lags $d_{i,j} = 1$

Assume upper bound *U* on *u*

Updating waiting list & bound on *u*

$$S_{t,j,0} = \lambda_{j,t} + \sum_{i\in\mathcal{J}} \sum_{u\in\mathcal{U}} q_{i,j} x_{t-1,i,u}, \qquad \forall j \in \mathcal{J}, t \in \mathcal{T},$$

$$S_{t,j,U} = \sum_{u=U-1}^{U} \left(S_{t-1,j,u} - x_{t-1,j,u}\right), \qquad \forall j \in \mathcal{J}, t \in \mathcal{T},$$

$$S_{t,j,u} = S_{t-1,j,u-1} - x_{t-1,j,u-1}, \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}, u \in \mathcal{U}\setminus\{0, U\}$$

Limit on the decision space

$$x_{t,j,u} \leq S_{t,j,u}, \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}, u \in \mathcal{U},$$

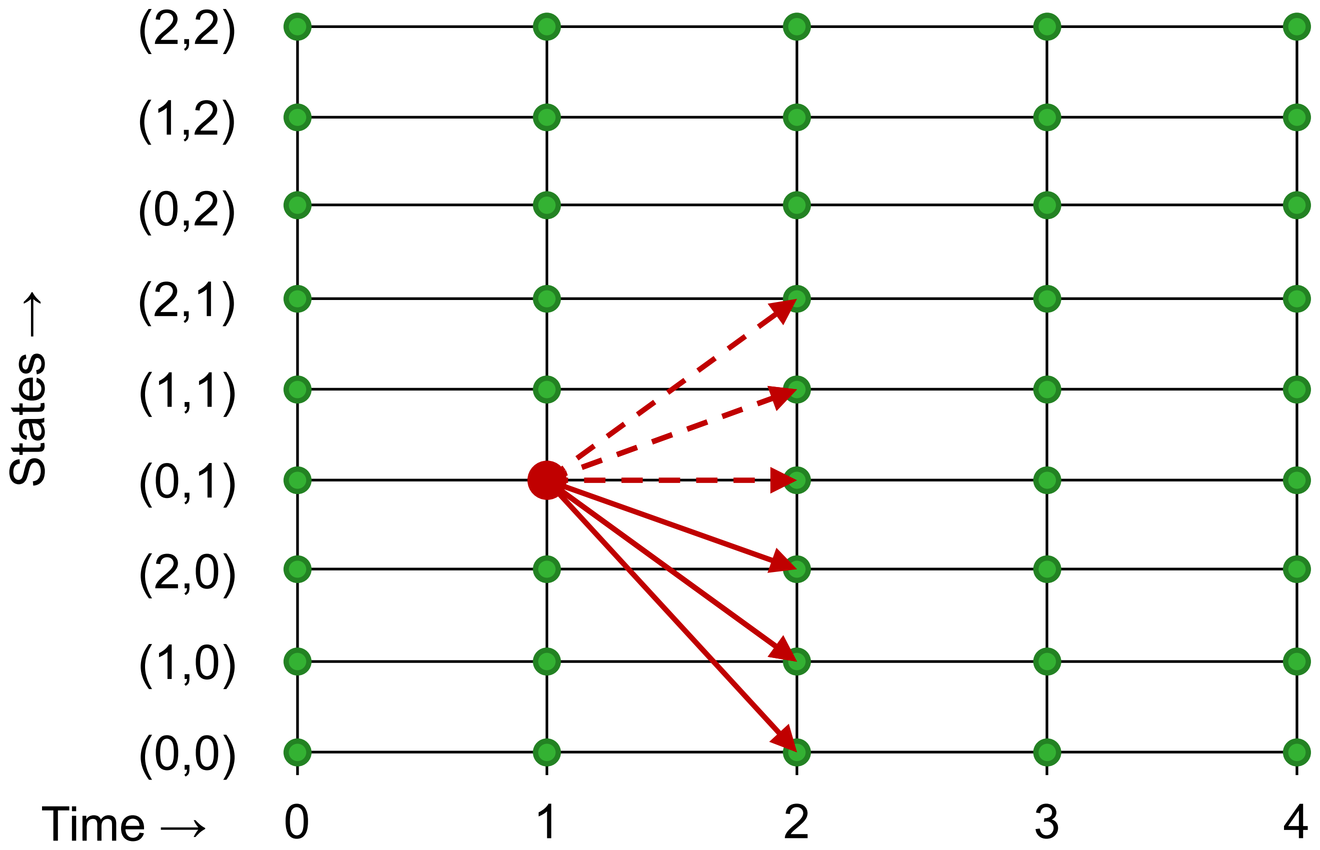
$$\sum_{j\in\mathcal{J}^r} s_{j,r} \sum_{u\in\mathcal{U}} x_{t,j,u} \leq \eta_{r,t}, \qquad \forall r \in \mathcal{R}, t \in \mathcal{T},$$

$$x_{t,j,u} \in \mathbb{Z}_+, \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}, u \in \mathcal{U}.$$

[1] Hulshof PJ, Boucherie RJ, Hans EW, Hurink JL. (2013) Tactical resource allocation and elective patient admission planning in care processes. Health Care Manag Sci. 16(2):152-66.

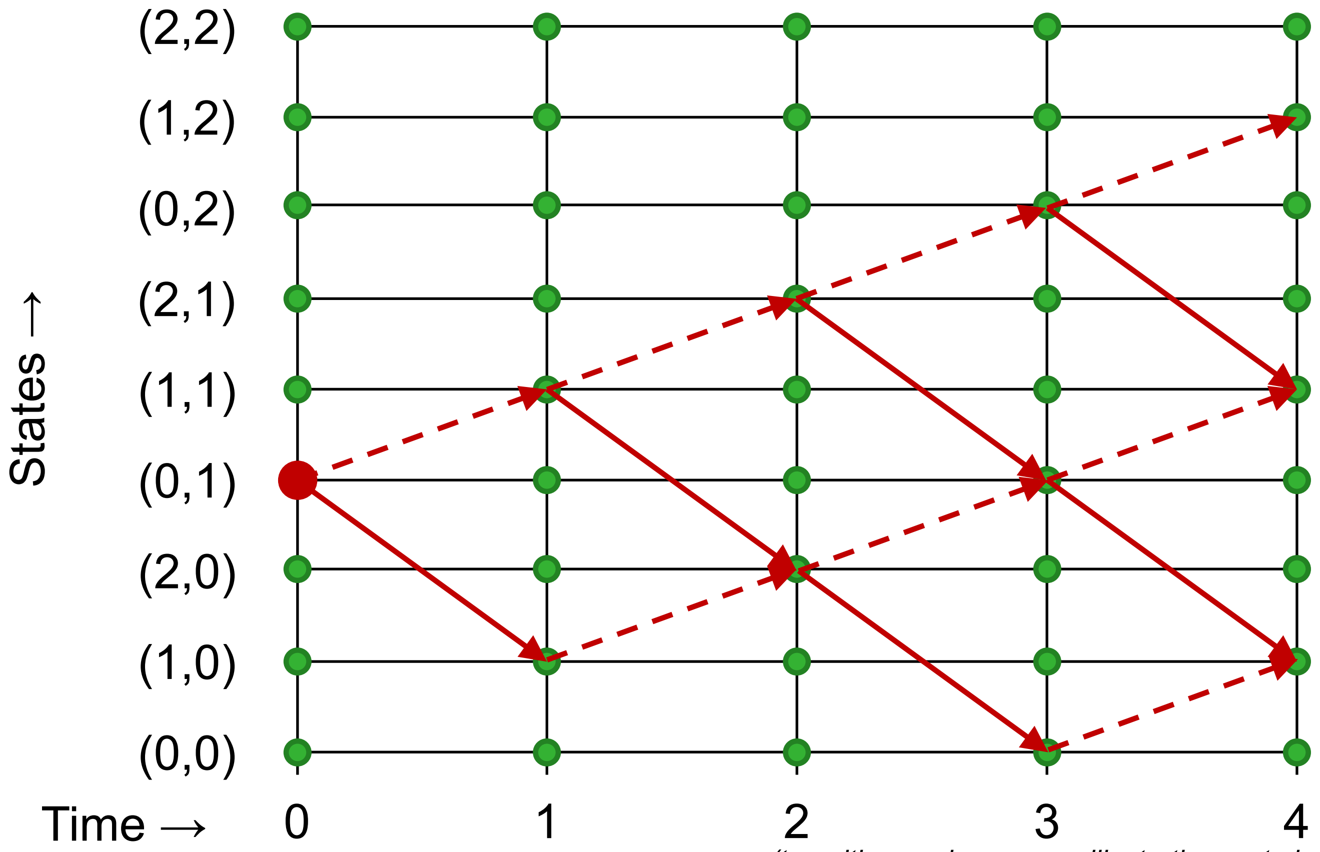
ILLUSTRATION MIXED INTEGER LINEAR PROGRAM

Deterministic problem - expected values: 1 arrival



MIXED INTEGER LINEAR PROGRAM (MILP)

Evaluate costs for all possible sequences of decisions



(transitions only serve as illustration, not always feasible)

PROS & CONS OF THE MILP

- Pros:
  - Suitable to support integrated decision making for multiple resources, multiple time periods, and multiple patient groups.
  - Flexible formulation (other objective functions can easily be incorporated).
- Cons:
  - Quite limited in the state space.
  - Rounding problems with fraction of patients moving from one queue to another after service.
  - Model does not include any form of randomness.

MODELLING STOCHASTICITY [1/2]

- We introduce W_t: vector of random variables representing all the new information that becomes available between time *t−1* and *t*.
- We distinguish between *exogenous* and *endogenous* information:

$$W_t = \left(\widehat{S}_t^e , \widehat{S}_t^o \left(x_{t-1} \right) \right) , \qquad \forall t \in \mathcal{T}$$

Patient arrivals from outside the system

Patient transitions as a function of the decision vector x_{t-1}, the number of patients we decided to admit in the previous time period.

MODELLING STOCHASTICITY [2/2]

- Transition function to capture the evolution of the system over time as a result of the decisions and the random information: $S_t = S^M(S_{t-1}, x_{t-1}, W_t)$
- Where

$$S_{t,j,0} = \widehat{S}^e_{t,j} + \sum_{i \in \mathcal{J}} \widehat{S}^o_{t,i,j}\left(x_{t-1,i}\right), \qquad \forall j \in \mathcal{J}, t \in \mathcal{T},$$

$$S_{t,j,U} = \sum_{u=U-1}^{U} \left(S_{t-1,j,u} - x_{t-1,j,u}\right), \qquad \forall j \in \mathcal{J}, t \in \mathcal{T},$$

$$S_{t,j,u} = S_{t-1,j,u-1} - x_{t-1,j,u-1}, \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}, u \in \mathcal{U} \setminus \{0, U\}$$

- Stochastic counterpart of the first three constraints in the ILP

OBJECTIVE [1/2]

- Find a policy (a decision function) to make decisions about the number of patients to serve at each queue.
- Decision function $X_t^{\pi}(S_t)$ that returns a decision $x_t \in \mathcal{X}_t(S_t)$ under the policy $\pi \in \Pi$. The set Π refers to the set of potential policies.
- The set $\mathcal{X}_t(S_t)$ refers to the set of feasible decisions at time t, which is given by:

$$
\begin{aligned}
\mathcal{X}_t\left(S_t\right) = \{ \quad & x_t | & \\
& x_{t,i,u} \leq S_{t,i,u}, & \forall i \in \mathcal{J}, t \in \mathcal{T}, u \in \mathcal{U} \\
& \textstyle\sum_{j \in \mathcal{J}^r} s_{j,r} \sum_{u \in \mathcal{U}} x_{t,j,u} \leq \eta_{r,t}, & \forall r \in \mathcal{R}, t \in \mathcal{T} \\
& x_{t,j,u} \in \mathbb{Z}_+ & \forall i \in \mathcal{J}, t \in \mathcal{T}, u \in \mathcal{U}\}
\end{aligned}
$$

- Equal to the last three constraints in the ILP

OBJECTIVE [2/2]

- Our goal is to find a policy π that minimizes the expected costs over all time periods given initial state S_0:

$$\min_{\pi \in \Pi} \mathbb{E} \left\{ \sum_{t \in \mathcal{T}} C_t \left(S_t, X_t^{\pi} \left(S_t \right) \right) | S_0 \right\}$$

where $S_{t+1} = S^M(S_t, x_t, W_{t+1})$ and $x_t \in \mathcal{X}_t(S_t)$.

- By Bellman's principal of optimality, we can find the optimal policy:

$$V_t \left(S_t \right) = \min_{x_t \in \mathcal{X}_t(S_t)} C_t \left(S_t, x_t \right) + \mathbb{E} \left\{ V_{t+1} \left(S_{t+1} \right) | S_t, x_t, W_{t+1} \right\}$$

- Compute expectation evaluating all possible outcomes $w_{i,j}$ for the number of patients transferred from *i* to *j*, with $w_{0,j}$ representing external arrivals and $w_{j,0}$ patients leaving the system.

DYNAMIC PROGRAMMING FORMULATION

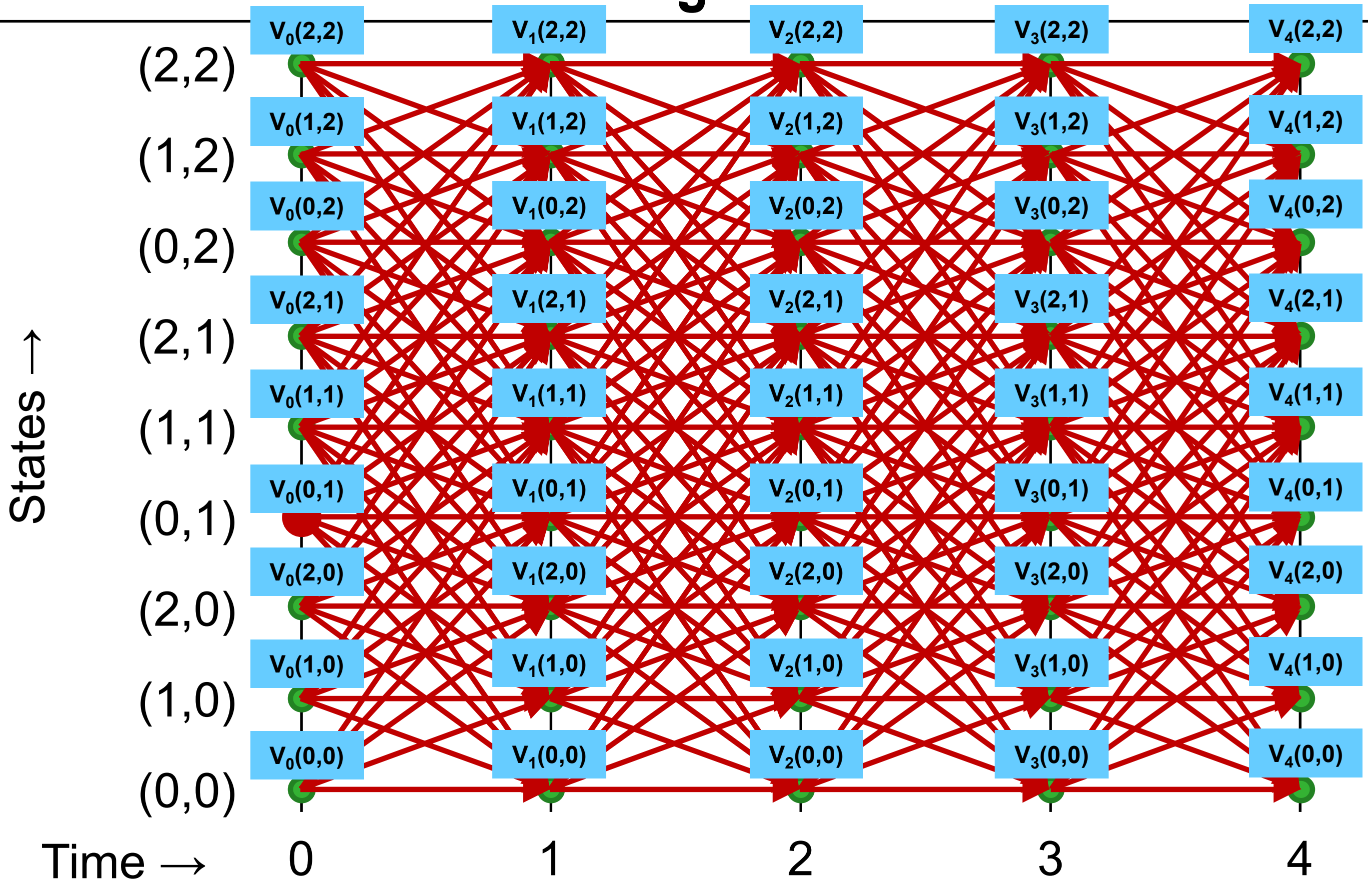
- Solve: $V_t\left(S_t\right) = \min_{x_t \in \mathcal{X}(S_t)} C_t\left(S_t, x_t\right) + \sum_{w'} P(w'|x_t) V_{t+1}\left(S_{t+1}|S_t, x_t, w'\right)$
- By backward induction.
- Where $P(w'|x_t) = \sum_{w_0=0}^{\infty} P(w_0) \times$

$$\sum_{\left\{\begin{array}{c} w_{ij}, i=0,\ldots,|\mathcal{J}|, j=0,\ldots,|\mathcal{J}| : \\ w_{ij} \geq 0, w_{ij}=0 \text{ if } p_{i,j}=0, w_{00}=0, \\ w_j = \sum_{u \in \mathcal{U}} x_{t,j,u}, j=1,\ldots,|\mathcal{J}|, \\ \sum_{j=0}^{|\mathcal{J}|} w_{ij} = w_j, i=0,\ldots,|\mathcal{J}|, \\ \sum_{i=0}^{|\mathcal{J}|} w_{ij} = w'_j, j=0,\ldots,|\mathcal{J}| \end{array}\right\}} \prod_{i=0}^{|\mathcal{J}|} \binom{w_i}{w_{i0},\ldots,w_{i|\mathcal{J}|}} \prod_{j=0}^{|\mathcal{J}|} p_{i,j}^{w_{ij}}$$

$$P(w_0) = \frac{\lambda_{0,t}^{w_0}}{w_0!} e^{-\lambda_{0,t}}, \quad \text{with } \lambda_{0,t} = \sum_{j=1}^{|\mathcal{J}|} \lambda_{j,t} \quad p_{i,j} = \begin{cases} q_{i,j}, & \text{when } i=1,\ldots,|\mathcal{J}|, j=0,\ldots,|\mathcal{J}| \\ \frac{\lambda_{j,t}}{\lambda_{0,t}}, & \text{when } i=0, j=1,\ldots,|\mathcal{J}| \\ 0, & \text{when } i=0, j=0 \end{cases}$$

DYNAMIC PROGRAMMING (DP)

Calculate the exact cost-to-go backwards



(transitions only serve as illustration, not always feasible)

THREE CURSUS OF DIMENSIONALITY

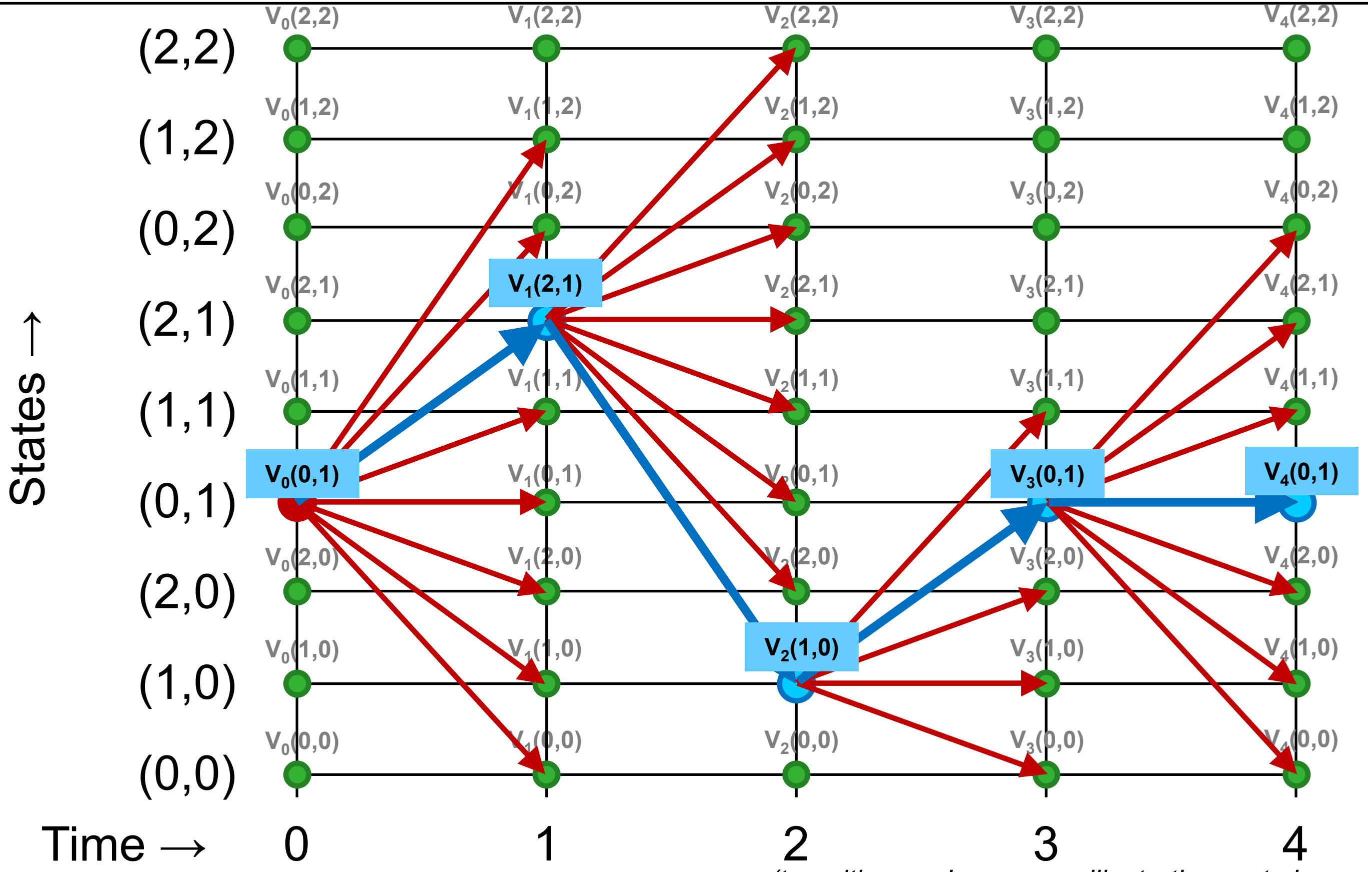
1. State space S_t too large to evaluate $V_t(S_t)$ for all states:
   - Suppose we have a maximum $\bar{S}$ for the number of patients per queue and per number of time periods waiting. Then, the number of states per time period is $\bar{S}^{|J|\times|U|}$.
   - Suppose we have *40* queues (e.g., *8* care processes with an average of *5* stages), and a maximum of *4* time periods waiting. Then we have $\bar{S}^{160}$ states, which is intractable for any $\bar{S} > 1$.
2. Decision space $\mathcal{X}_t(S_t)$ (combination of patients to treat) is too large to evaluate the impact of every decision.
3. Outcome space (possible states for the next time period) is too large to compute the expectation of cost-to-go). Outcome space is large because state space and decision space is large.

APPROXIMATE DYNAMIC PROGRAMMING (ADP)

- How ADP is able to handle realistic-sized problems:
  - Large state space: generate sample paths, stepping forward through time.
  - Large outcome space: use post-decision state.
  - Large decision space: problem remains (although evaluation of each decision becomes easier).
- ADP formulation uses all of the constraints from the MILP and uses a similar objective function (although formulated in a recursive manner).

APPROXIMATE DYNAMIC PROGRAMMING (ADP)

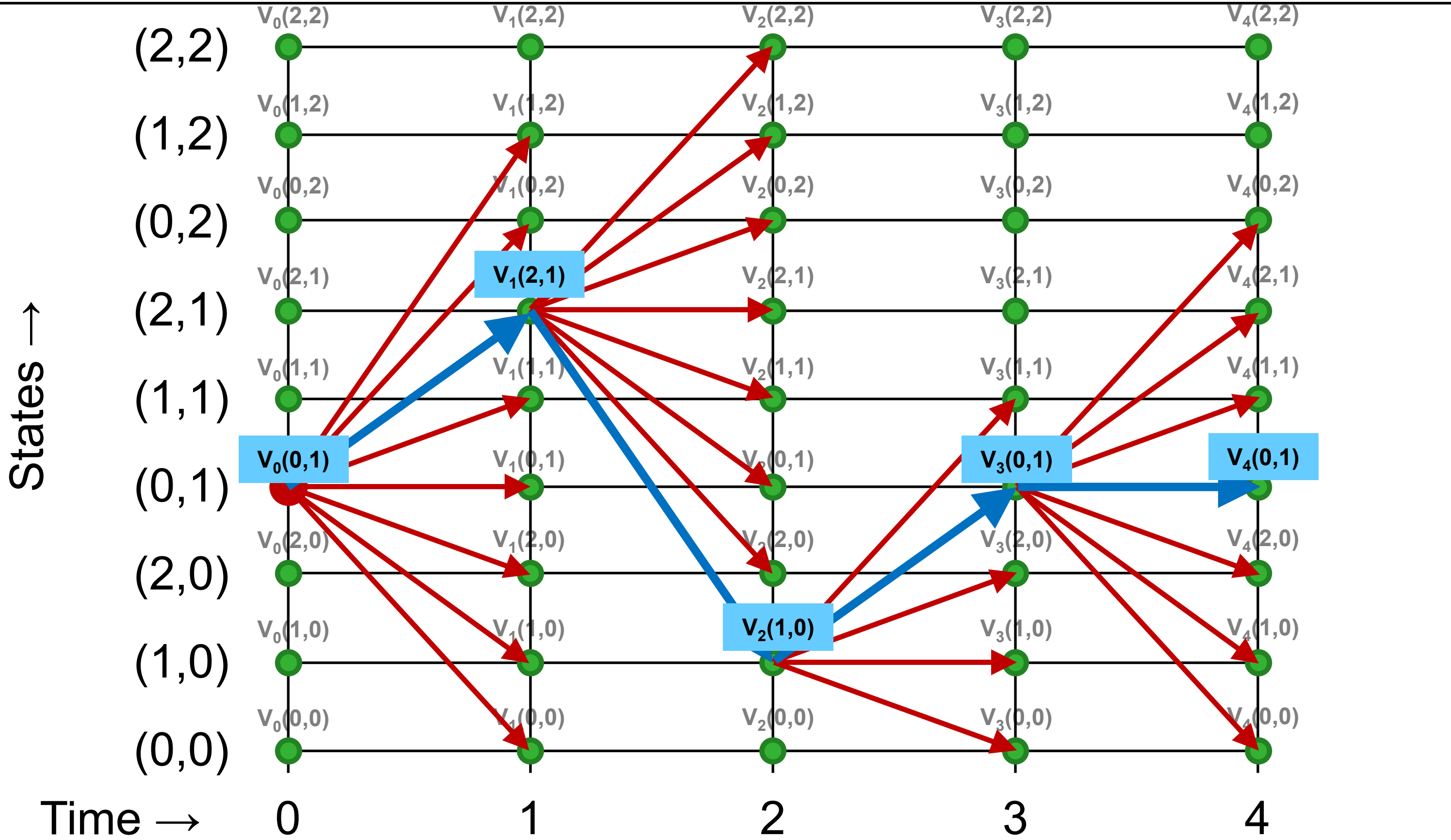
Learn cost-to-go forwards iteratively: single pass



(transitions only serve as illustration, not always feasible)

APPROXIMATE DYNAMIC PROGRAMMING (ADP)

Learn cost-to-go forwards iteratively: double pass



(transitions only serve as illustration, not always feasible)

TRANSITION TO POST-DECISION STA

Expected transitions of the treated patients

- Besides the earlier transition function, we now define a transition function from pre S_t to post S_t^x: $S_t^x = S^{M,x}(S_t, x_t)$.
- With

$$S^x_{t,j,0} = \sum_{i \in \mathcal{J}} \sum_{u \in \mathcal{U}} q_{i,j} x_{t,i,u} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}$$

$$S^x_{t,j,U} = \sum_{u=U-1}^{U} \left(S_{t,j,u} - x_{t,j,u}\right) \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}$$

$$S^x_{t,j,u} = S_{t,j,u-1} - x_{t,j,u-1} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T}, u \in \mathcal{U} \setminus \{0, U\}$$

- Deterministic function of the current state and decision.
- Expected results of our decision are included, not the new arrivals.

ADP FORMULATION

- We rewrite the DP formulation as

$$V_t(S_t) = \min_{x_t \in \mathcal{X}_t(S_t)} \Big(C_t(S_t, x_t) + V_t^x(S_t^x) \Big)$$

where the value function $V_t^x(S_t^x)$ for the cost-to-go of the post-decision state S_t^x is given by

$$V_t^x(S_t^x) = \mathbb{E}[V_{t+1}(S_{t+1}) | S_t^x]$$

- We replace this function with an approximation $\bar{V}_t^{x,n-1}(S_t^x)$.
- We now have to solve

$$\tilde{x}_t^n = \arg \min_{x_t \in \mathcal{X}_t(S_t)} \Big(C_t(S_t, x_t) + \bar{V}_t^{x,n-1}(S_t^x) \Big)$$

- The value of decision $\tilde{x}_t^n$ is represented by $\hat{v}_t^n$

APPROXIMATE DYNAMIC PROGRAMMING (ADP)

Deterministic optimization

Statistics

Simulation

1. Initialization: approximation $\bar{V}_t^0(S_t), \forall t$, initial state S_1 and *n=1*.
2. Do for t=1,…,T
   - Solve:

     $$\tilde{x}_t^n = \arg \min_{x_t \in \mathcal{X}_t(S_t)} \left(C_t(S_t, x_t) + \bar{V}_t^{x,n-1}(S_t^x) \right)$$

   - If *t>1* update approximation $\bar{V}_{t-1}^{x,n}(S_{t-1}^x)$ for the previous post-decision state S_{t-1}^x using the value $\hat{v}_t^n$ resulting from decision $\tilde{x}_t^n$.
   - Find the post-decision state S_t^x.
   - Obtain a sample realization W_{t+1} and compute new pre-decision state S_{t+1}.
3. Increment *n*. If $n \leq N$ go to 2.
4. Return $\bar{V}_t^{x,N}(S_t^x), \forall t$.

REMAINING CHALLENGES

- What we have so far:
  - ADP formulation that uses all of the constraints from the ILP formulation and uses a similar objective function (although formulated in a recursive manner).
  - ADP differs from the other approaches by using sample paths. These sample paths visit one state per time period. For our problem, we are able to visit only a fraction of the states per time unit ($\ll 1\%$).
- Remaining challenge:
  - To design a proper approximation for the ‘future’ costs $\bar{V}_t^n(S_t^x)$ (1) that is computationally tractable, (2) provides a good approximation of the actual value, and (3) is able to generalize across the state space.

VALUE FUNCTION APPROXIMATION [1/2]

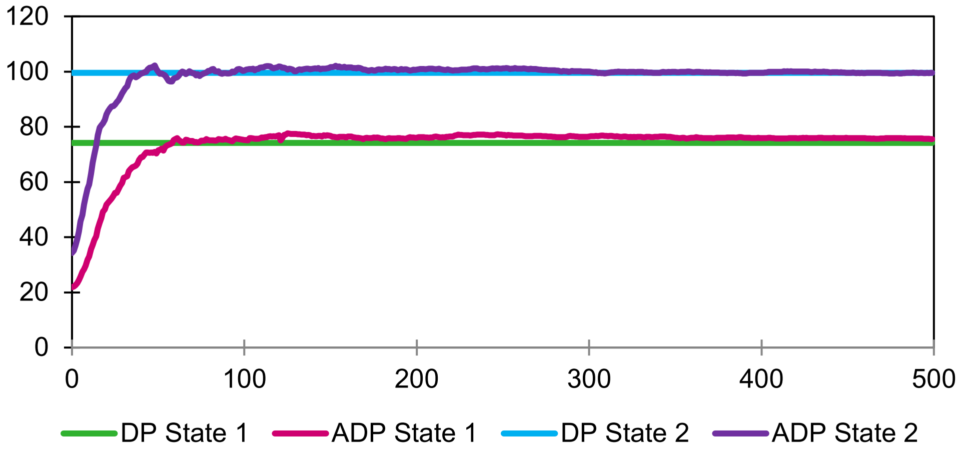
- Consider the patient admission planning case from the previous lecture: the state $S_t = (S_{t,j,u}) \forall j, u$ gives the number of patients waiting u time units in queue j, $\forall j, u$
- Size of the state space: $\bar{S}^{|J| \times |U|}$ (with $\bar{S}$ being an upper bound on the number of patients of the same type)
- Examples of features:
  - Total number of patients waiting in a queue
  - Average/longest waiting time of patients in a queue
  - Number of waiting patients requiring resource r
  - Combination of these

VALUE FUNCTION APPROXIMATION [2/2]

- We use the features "number of patients in queue *j* that are *u* time periods waiting at time *t*" in combination with a constant.
- This choice of basis functions explains a large part of the variance in the computed values with the exact DP approach ($R^2 = 0.954$). Instance considered:
  - We use *8* time units, *1* resource types, *1* care process, *3* stages in the care process (3 queues), *U=1* (zero or *1* time unit waiting), for DP max *8* patients per queue.
  - This results in $8 \times 8^{3\times2} = 2{,}097{,}152$ states in total.
- See https://link.springer.com/article/10.1007/s10696-015-9219-1

CONVERGENCE RESULTS ON SMALL INSTANCES

- Tested on *5000* random initial states.
- DP requires *120* hours, ADP *0.439* seconds for *N=500*.
- Convergence (bit overestimation due to truncated state space):



- Performance of ADP *2%* away from optimum.

PERFORMANCE ON SMALL AND LARGE INSTANCES

- Compare with greedy policy: fist serve the queue with the highest costs until another queue has the highest costs, or until resource capacity is insufficient.
- We train ADP using *100* replications after which we fix our value functions. We simulate the performance of using (i) the greedy policy and (ii) the policy determined by the value functions.
- ADP results in *29%* savings compared to greedy (higher fluctuations in resource availability or patient arrivals results in larger differences between ADP and greedy).

UNIVERSITY OF TWENTE.

QUESTIONS?

Martijn Mes
Full professor
University of Twente
Faculty of Behavioural, Management and Social sciences
Dept. Industrial Engineering and Business Information Systems

Contact
+31-534894062
m.r.k.mes@utwente.nl
https://www.utwente.nl/en/bms/iebis/staff/mes/