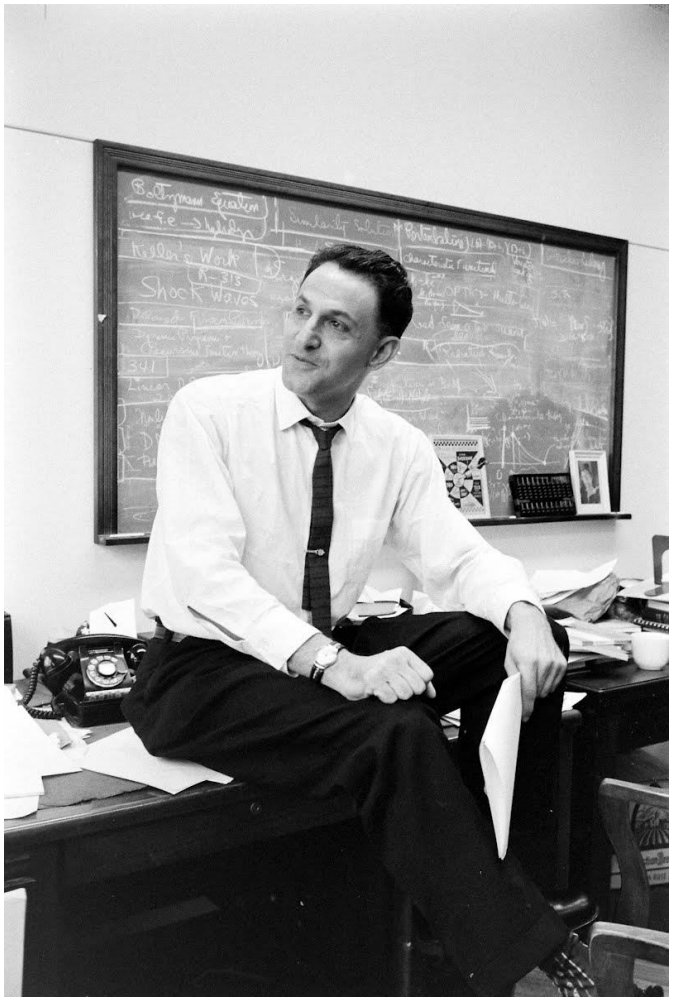
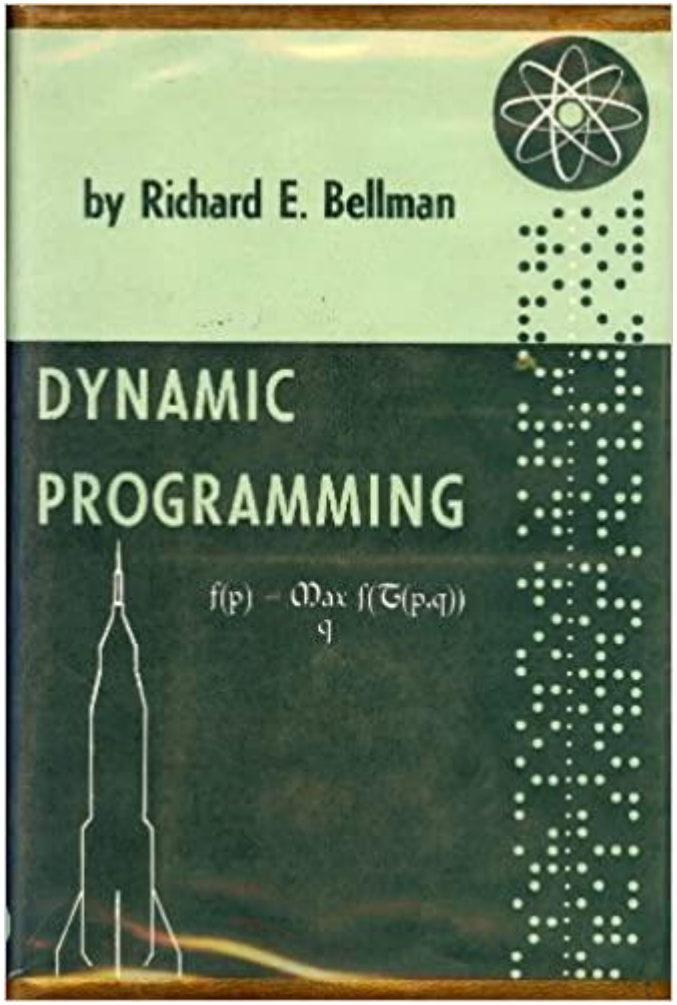
Chapter 6
Dynamic Programming

The Founder of Dynamic Programming





Professor at University of Southern California

*1920 (New York)
† 1984 (Los Angeles)

Dynamic Programming (DP)

Dynamic Programming is an approach to optimize a problem sequentially in multiple stages, each of which solves for an optimal single decision (or variable). The sequence of these single decisions constitutes the overall optimal solution.

Difficulties:

(1) To appropriately model a problem to be solved by dynamic programming is usually difficult and requires experience.

(2) There is no general algorithm for solving a dynamic program such as the Simplex algorithm for solving an LP. Rather, the problem must be specifically formulated based on the principals of DP.

6.1 Introductory Example: The Commuter's Problem

The Commuter's Problem (CP)

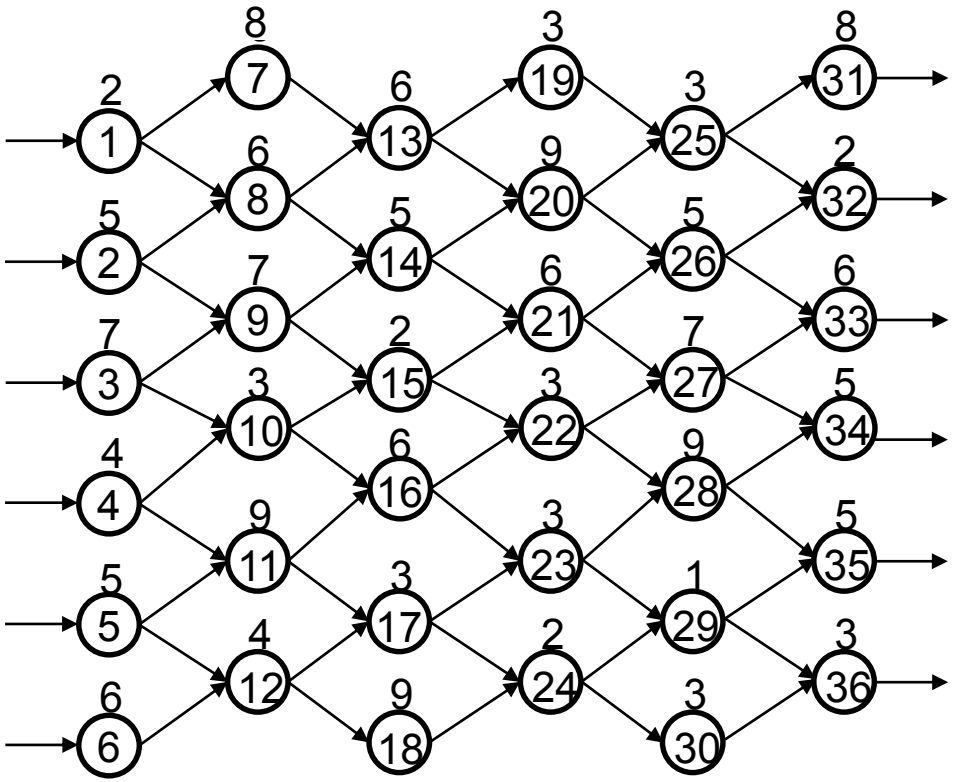
Residential areas

Intersections

Car lots

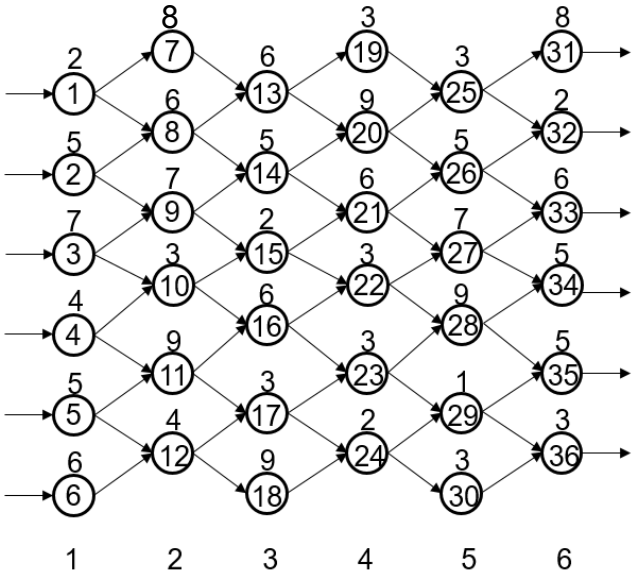
6 ← waiting time at intersection

(8) ← intersection number



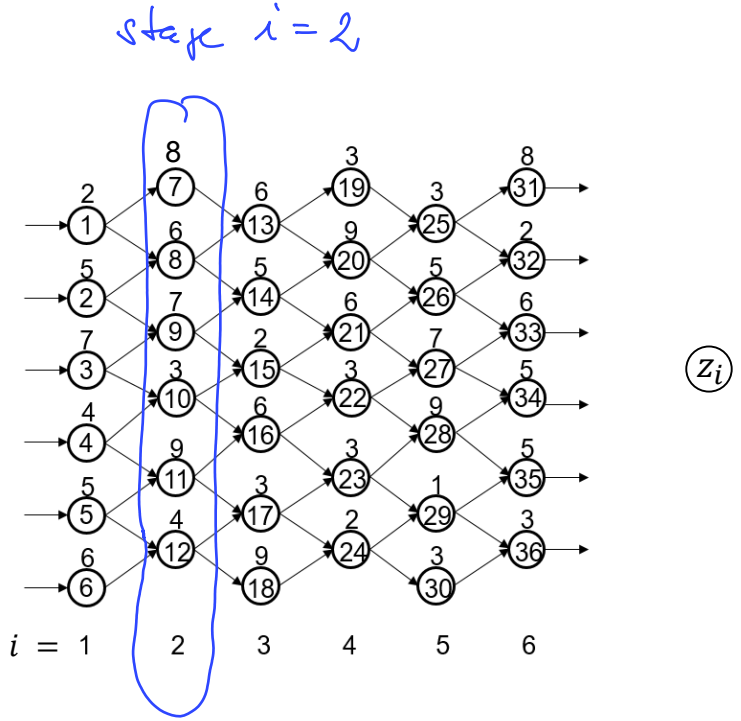
#intersections on the route 1 2 3 4 5 6

Solving the Commuter's Problem with Brute Force



Stages

n	Number of stages of the problem
i	Index for stages ($i = 1, \ldots, n$)
z_i	Specific state at stage i
Z_i	Set of states at stage i



$n = 6$ stages

$z_2 = \{7, 8, \ldots, 12\}$

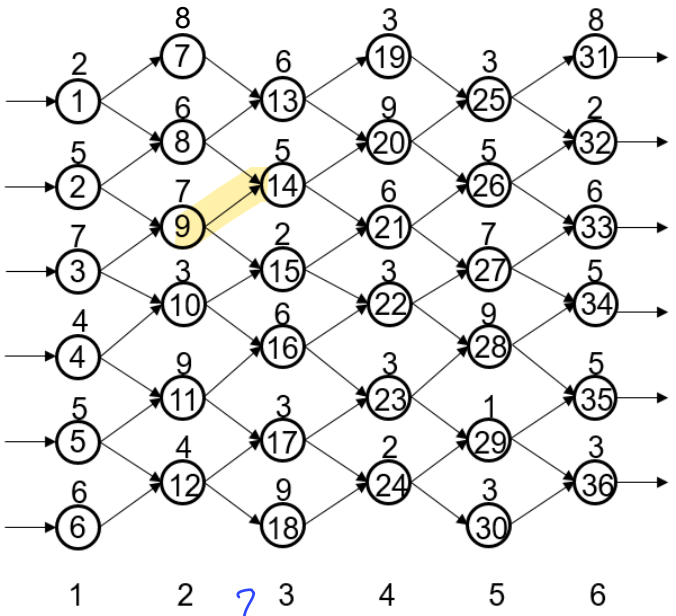
$z_2 = 11$ (driver is at intersection 11)

Decisions

(z_i, z_{i+1}) Street leading from the i-th intersection z_i to the $(i+1)$-th intersection z_{i+1}

$x_i(z_i)$ Specific decision at state z_i of stage i

$X_i(z_i)$ Set of decisions at state z_i of stage i



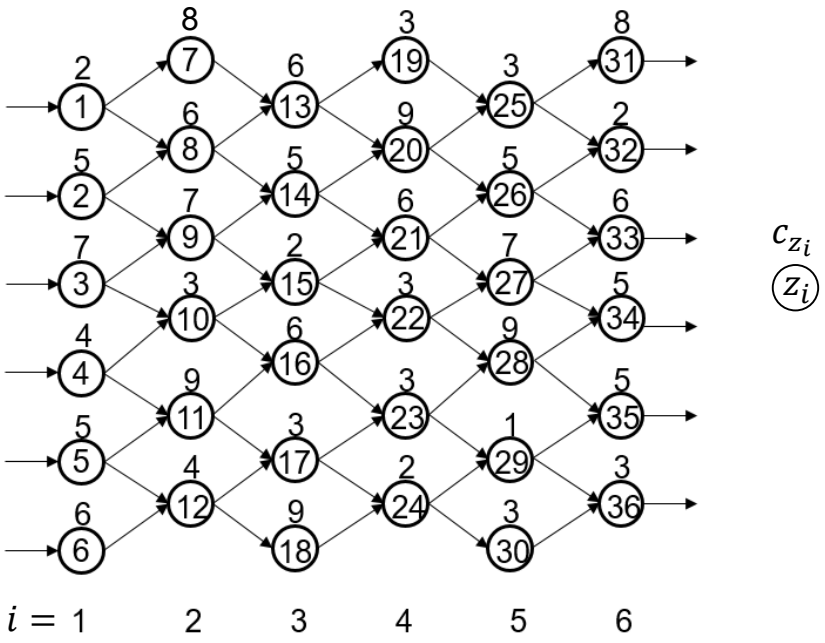
$(z_2 = 9,\ z_3 = 14)$

$x_2(z_2 = 9) = (z_2 = 9,\ z_3 = 14)$

$X_2(z_2 = 9) = \{(z_2 = 9,\ z_3 = 14),\ (z_2 = 9,\ z_3 = 15)\}$

State-Dependent Cost Function

$f_i(z_i) = c_{z_i}$

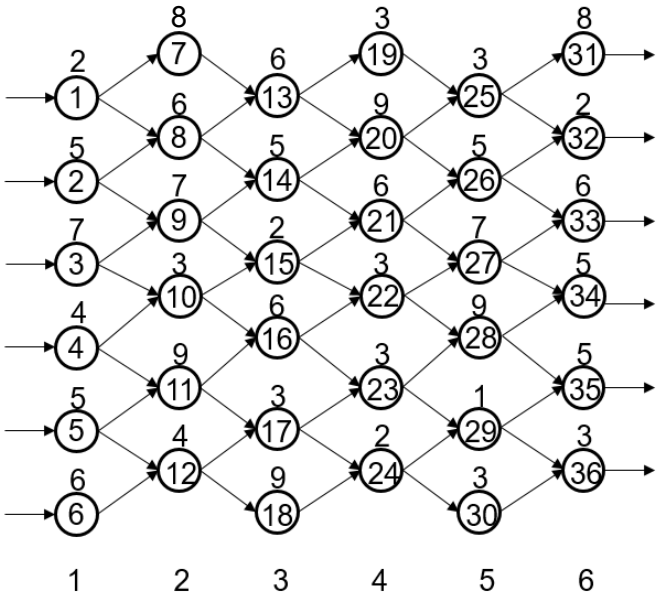


Backward Recursive Function

Total cost from state z_i in stage i to the final stage if decision $x_i(z_i)$ is taken

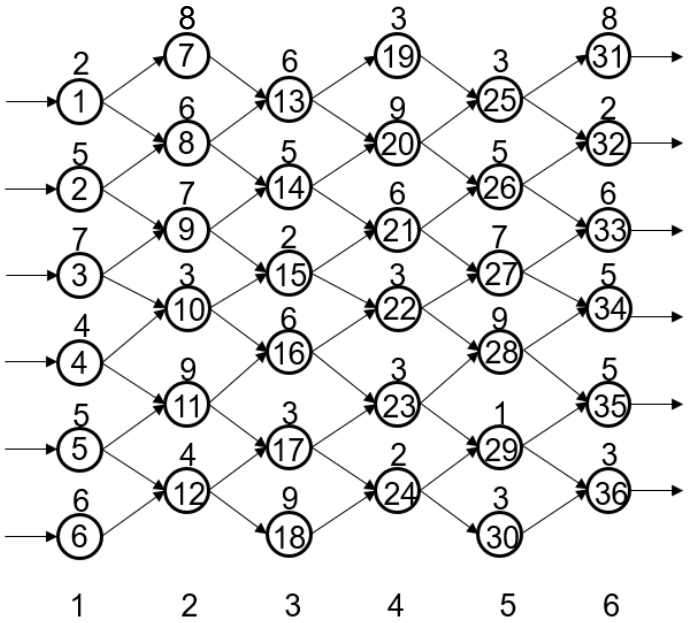
$$F_i(z_i, x_i) = f_i(z_i) + F^*_{i+1}(z_{i+1})$$

with $F^*_{i+1}(z_{i+1})$ being the minimum cost from state z_{i+1} to the final stage



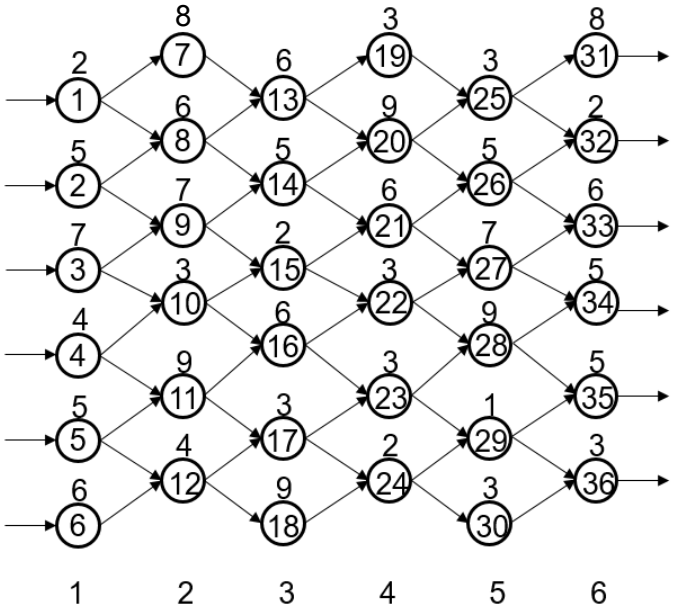
Minimum Total Cost Function

$$F_i^*(z_i) = \min_{x_i \in X_i(z_i)} \{f_i(z_i) + F_{i+1}^*(z_{i+1})\}$$



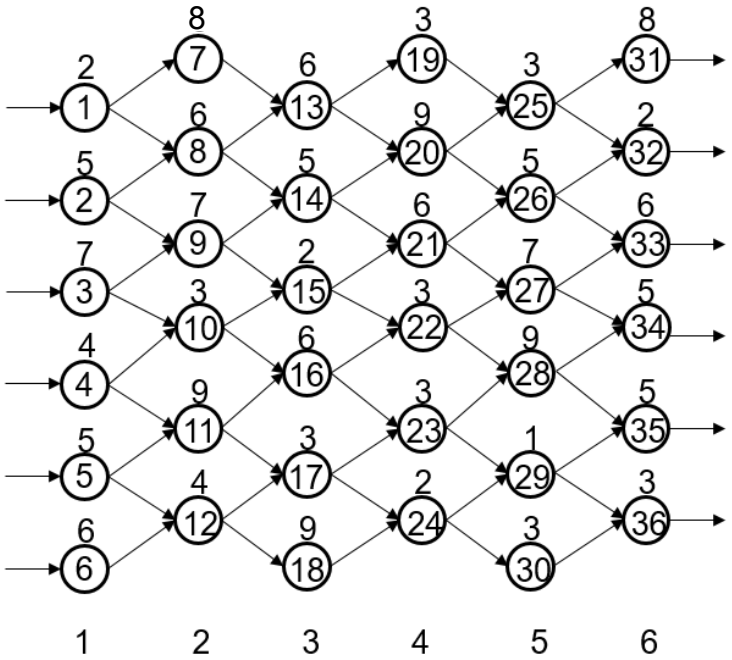
Optimality Principle of Dynamic Programming

For optimally solving a dynamic program it suffices to sequentially determine for each state z_i on each stage i the optimal decision $x_i^* \in X_i(z_i)$ in order to solve the overall problem to optimality.



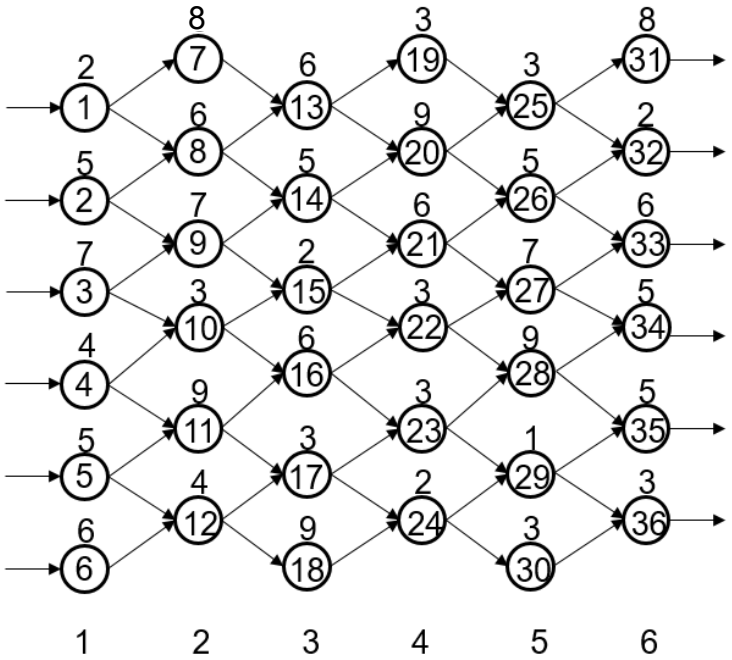
Solving the Commuter's Problem with DP: Initializing the Costs-to-Go in stage $i = 6$

z_6	31	32	33	34	35	36
$F_6^*(z_6) = c_{z_6}$						



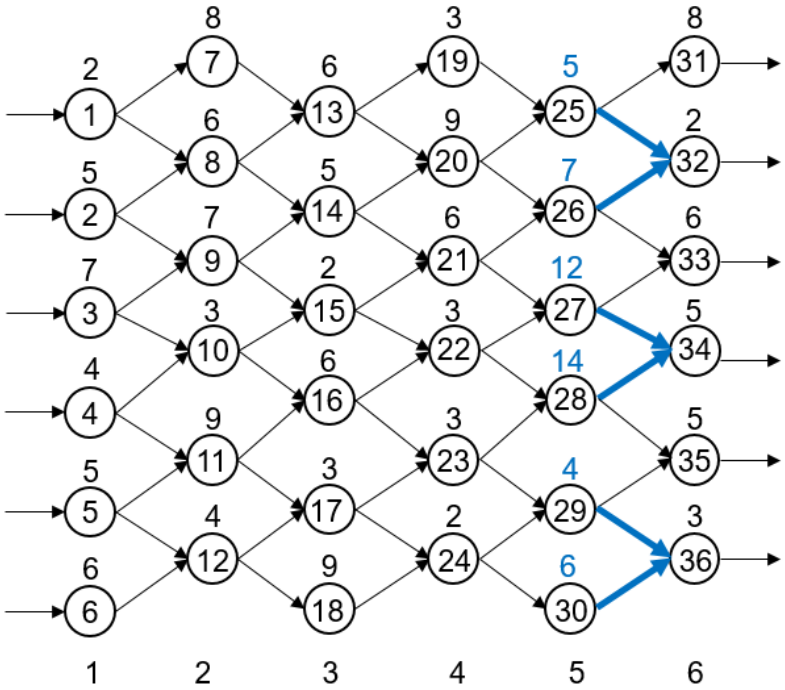
Solving the Commuter's Problem with DP: Stage $i = 5$

z_5	x_5	z_6	$f_5(z_5)$	$F_6^*(z_6)$	$F_5(z_5, x_5)$	$F_5^*(z_5)$



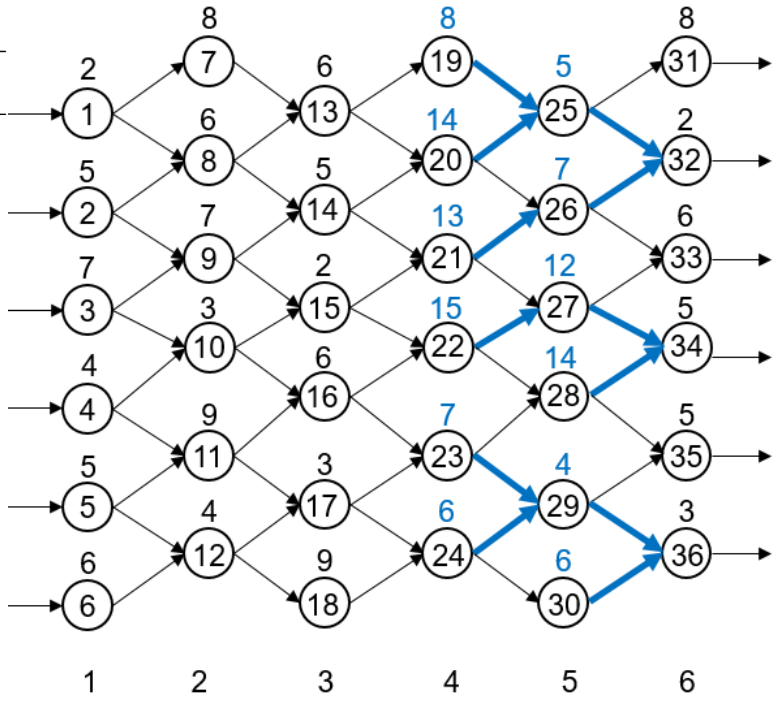
Solving the Commuter's Problem with DP: Stage $i = 4$

z_4	x_4	z_5	$f_4(z_4)$	$F_5^*(z_5)$	$F_4(z_4, x_4)$	$F_4^*(z_4)$



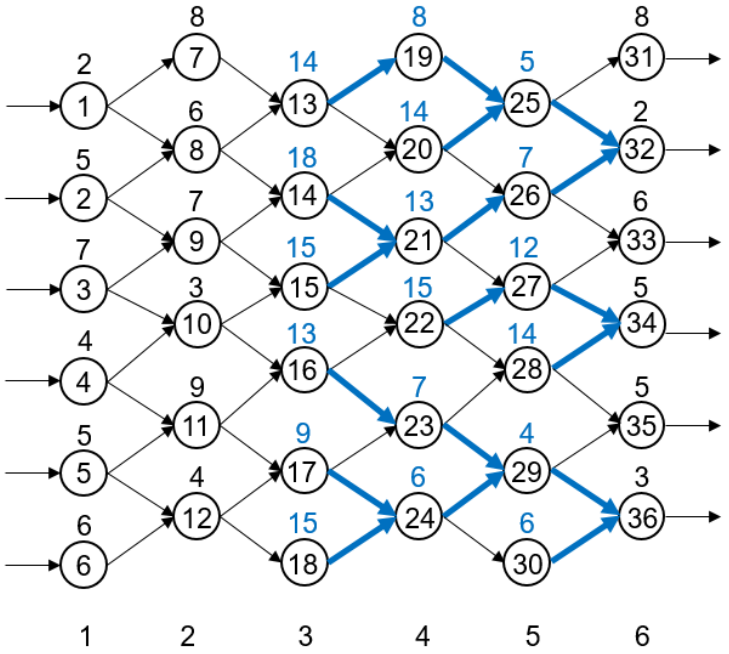
Solving the Commuter's Problem with DP: Stage $i = 3$

z_3	x_3	z_4	$f_3(z_3)$	$F_4^*(z_4)$	$F_3(z_3, x_3)$	$F_3^*(z_3)$



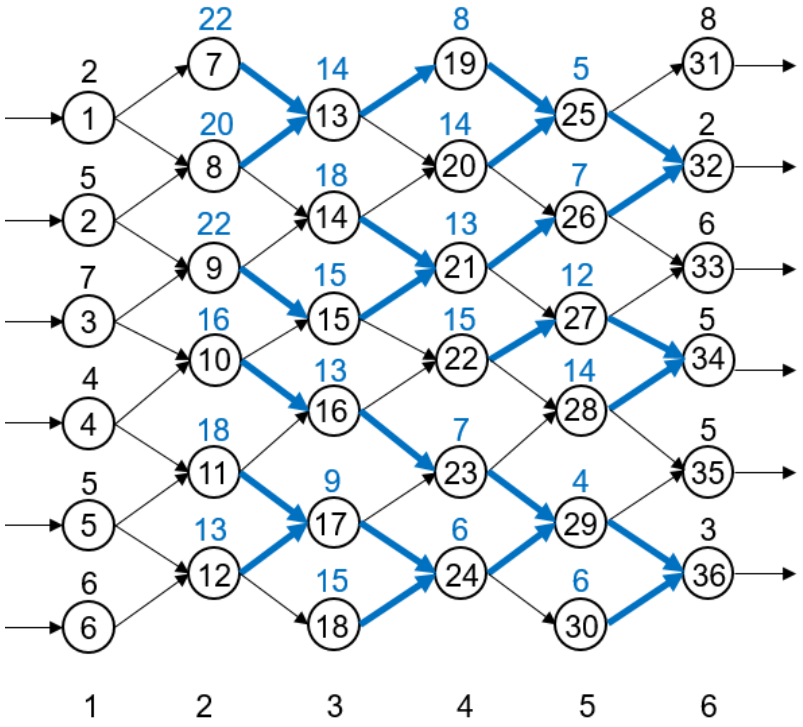
Solving the Commuter's Problem with DP: Stage $i = 2$

z_2	x_2	z_3	$f_2(z_2)$	$F_3^*(z_3)$	$F_2(z_2, x_2)$	$F_2^*(z_2)$

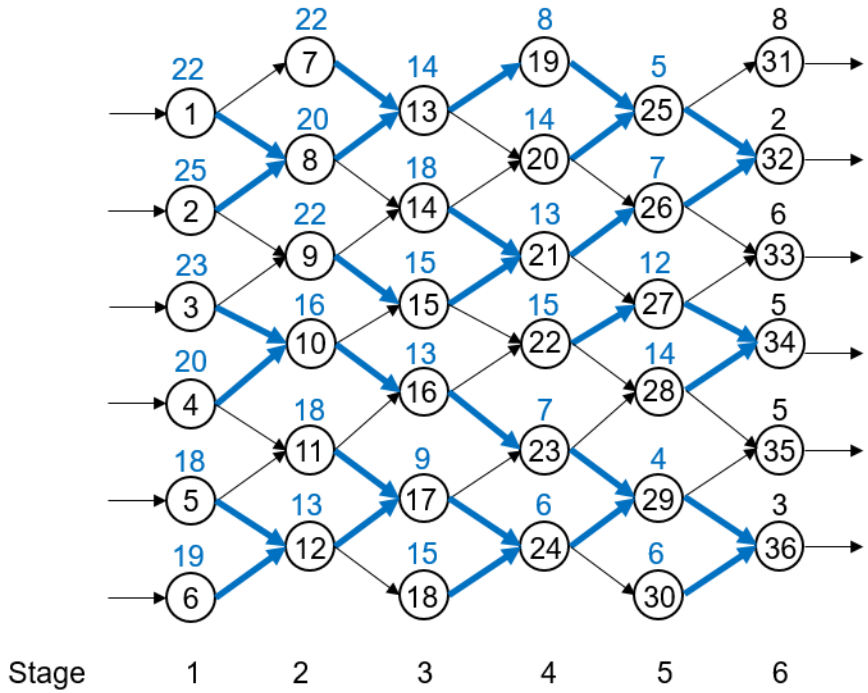


Solving the Commuter's Problem with DP: Stage $i = 1$

z_1	x_1	z_2	$f_1(z_1)$	$F_2^*(z_2)$	$F_1(z_1, x_1)$	$F_1^*(z_1)$



Optimal Solution and Computational Effort



6.2 Dynamic Ordering Problem

Problem Description Dynamic Ordering Problem

In the following 5 periods, the demands are given as follows:

Period i	1	2	3	4	5
Demand d_i	220	280	360	140	270

- Fixed cost of 250 for each delivery
- Variable cost of 1 for each unit in inventory at the end of a period
- No limit on the number of delivered units and the number of units in inventory
- No inventory at the begin of period 1 and the end of period 5
- All demands must be satisfied.

Decision**:** How many units shall be ordered (and delivered) at the beginning of each period such that demand is met and cost are minimized.

“Single Order in Period 1” Policy

i	1	2	3	4	5
d_i	220	280	360	140	270

Order quantity x_i

Inventory z_i

Inventory cost $1 \cdot z_i$

Delivery cost $\longrightarrow$ 250, if $x_i > 0$
0, otherwise

Total cost:

Inventory cost
+ Delivery cost
= Total cost

“Just-in-Time” Policy

i	1	2	3	4	5
d_i	220	280	360	140	270

Order quantity x_i

Inventory z_i

Inventory cost $1 \cdot z_i$

Delivery cost $\longrightarrow$ 250, if $x_i > 0$
0, otherwise

Total cost:

Inventory cost
+ Delivery cost
= Total cost

Properties of Optimal Policies

For an optimal policy the following holds for each period i=1,…,n:

1) An order for period i only takes place if the inventory at the end of period i-1 is 0.

2) An order $x_i > 0$ in period i equals the demand of periods i to τ with τ=i,…,n.

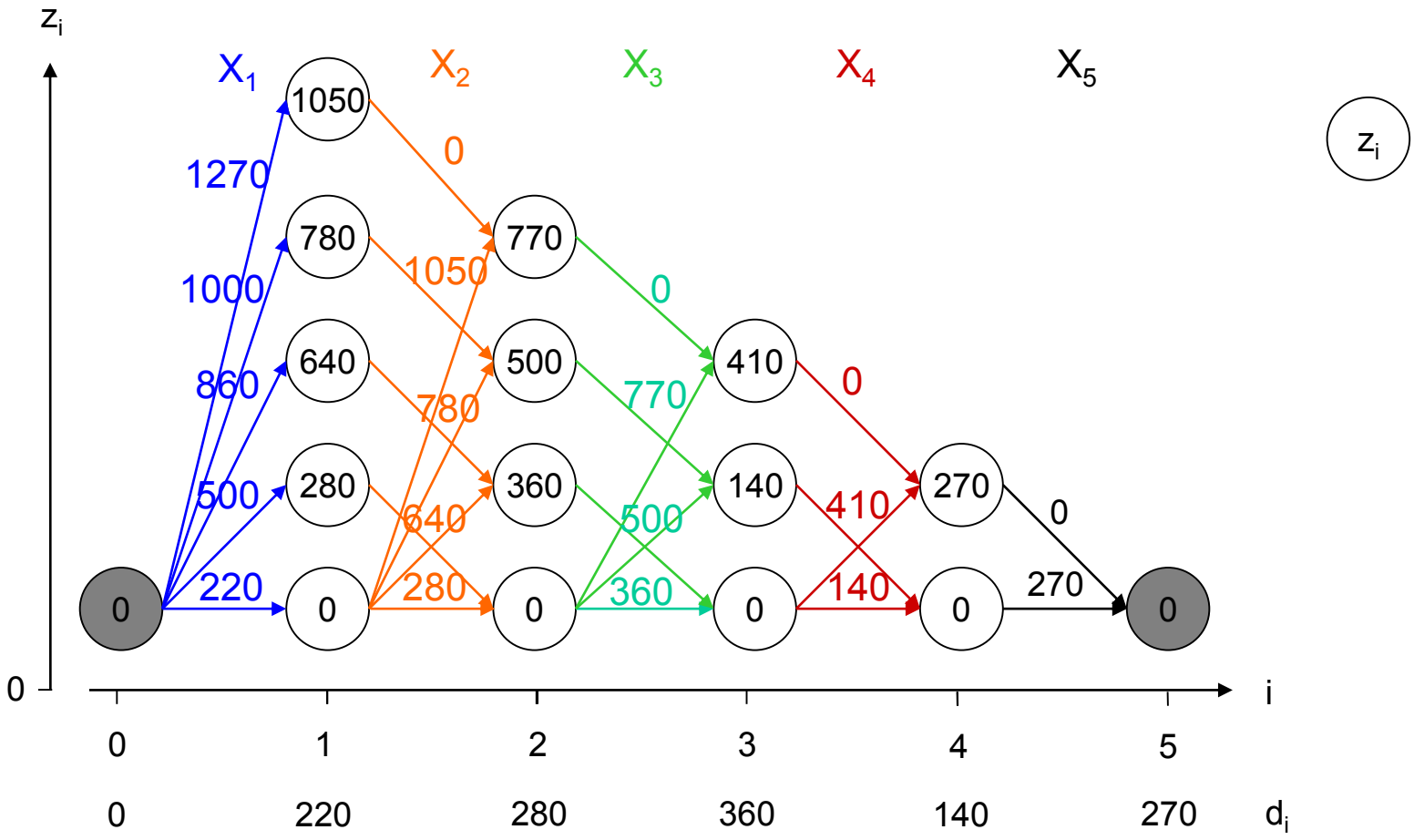
Stages and Decisions

n	Number of stages of the problem
x_i	Quantity ordered (and delivered) at the beginning of period i (i=1,…,5)
X_i	Set of possible order quantities for period i

$$x_1 \in X_1 = \left\{ \sum_{\tau=1}^{u} d_\tau \;\middle|\; u = 1,\ldots,n \right\}$$

$$x_i \in X_i = \left\{ 0, \sum_{\tau=i}^{u} d_\tau \;\middle|\; u = i,\ldots,n \right\} \qquad (i = 2,\ldots,n)$$

Stages and Decisions



Note: All downward arrows are associated with the zero order quantity.

States

z_i State variable: Inventory at the end of period i $(i = 0, \dots, 5)$

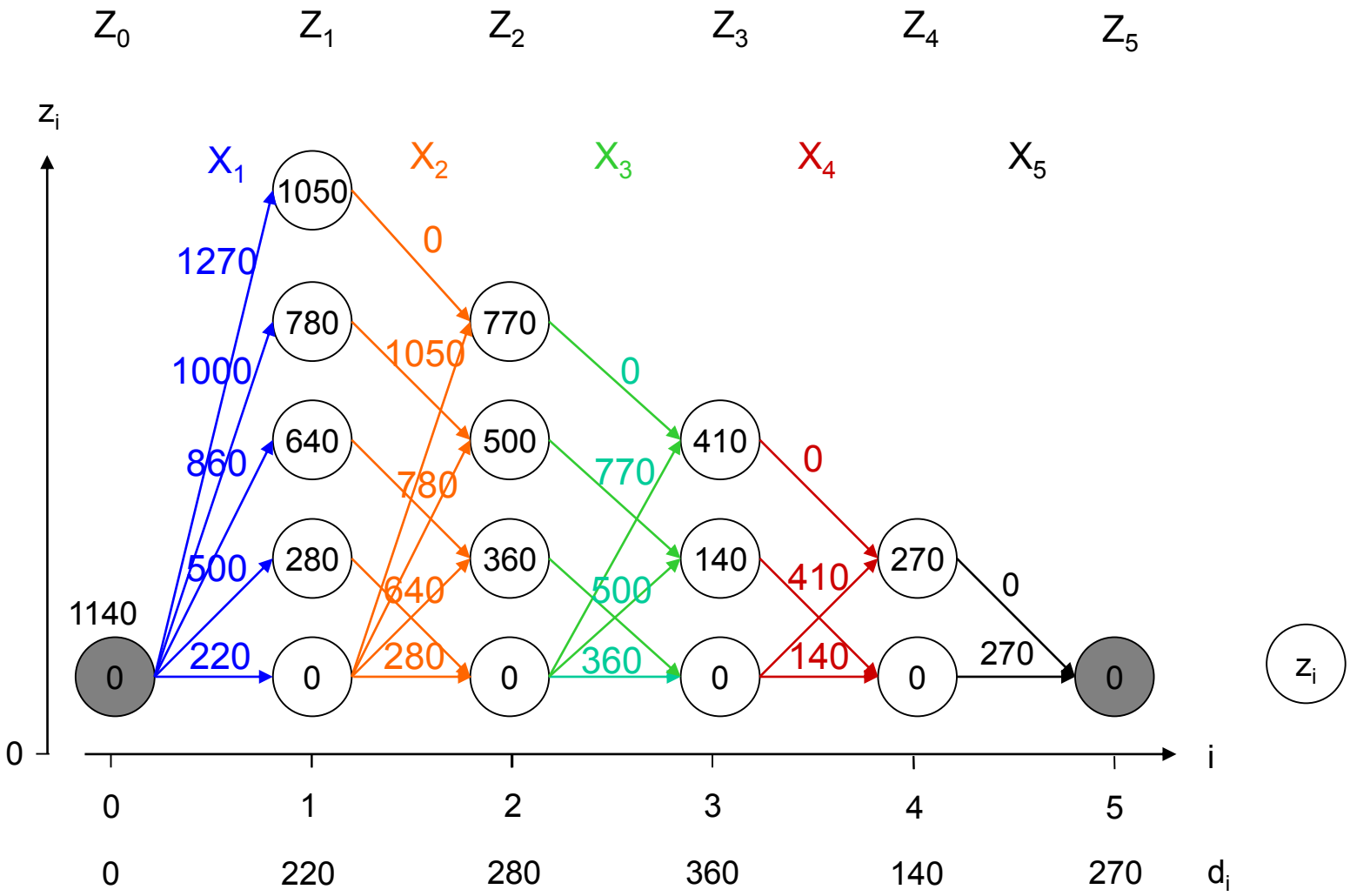
Single state in period 0: $Z_0 = \{0\}, z_0 = 0$

Single state in period 5: $Z_5 = \{0\}$, $z_5 = 0$

Set of states in period $i = 1, \dots, 4$

$$Z_i = \left\{ 0, \sum_{\tau=i+1}^{u} d_\tau \;\middle|\; u = i+1, \dots, n \right\} \qquad (i = 1, \dots, 4)$$

States


Z0
Z1
Z2
Z3
Z4
Z5
zi
X1
X2
X3
X4
X5
1050
780
640
280
0
770
500
360
0
410
140
0
270
0
1140
1270
1000
860
500
220
0
1050
780
640
280
770
500
360
0
410
140
0
270
0
270
zi
0
i
0
1
2
3
4
5
0
220
280
360
140
270
di

State Transition Function

$z_i = z_{i-1} + x_i - d_i$ (i=1,…,5) „Dynamic inventory balance constraint“

Example: $z_2=0$, $x_3=500$, $d_3=360$

Period i	1	2	3	4	5
Demand d_i	220	280	360	140	270

State-Dependent Cost Function

$$f_i(x_i, z_i) = \left(250 \cdot \begin{cases} 1, \text{ if } x_i > 0 \\ 0, \text{ otherwise} \end{cases} \right) + 1 \cdot z_i \qquad (i=1,\ldots,5)$$

Fixed cost per order

Variable inventory cost per unit and period

Backward Recursive Function

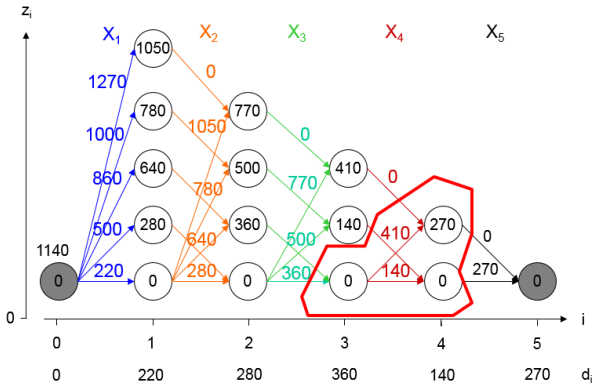
Cost in stage i depending on state z_i and decision x_{i+1}

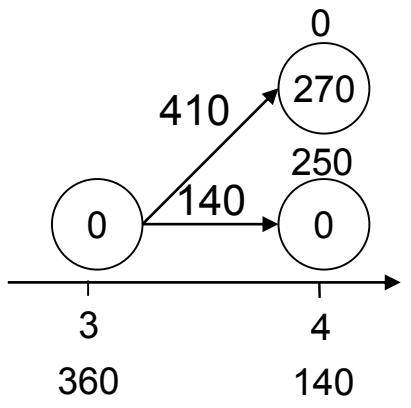
$$F_i(z_i, x_{i+1}) = f_{i+1}(z_{i+1}, x_{i+1}) + F^*_{i+1}(z_{i+1})$$

$$= 250 \cdot \begin{Bmatrix} 1, \text{ if } x_{i+1} > 0 \\ 0, \text{if } x_{i+1} = 0 \end{Bmatrix} + 1 \cdot z_{i+1} + F^*_{i+1}(z_{i+1})$$

($i=4,\ldots,0$; $z_i \in Z_i$; $x_{i+1} \in X_{i+1}$)

Example: i=3, z_3=0, $x_4 \in \{140, 410\}$



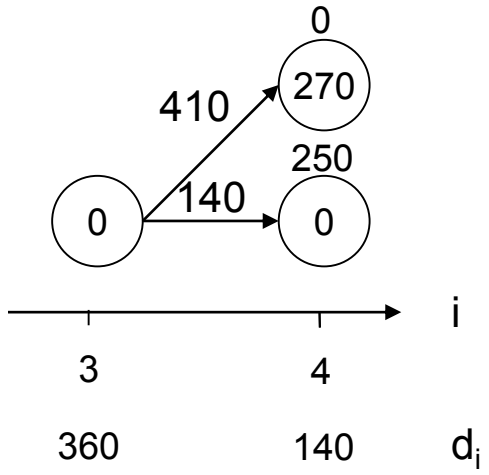


Backward Recursive Function: Optimality Principle of Bellman

Minimal cost in stage i depending on state z_i

$$F_i^*(z_i) = \min\left\{F_i(z_i, x_{i+1}) \mid x_{i+1} \in X_{i+1}(z_i)\right\}$$ (i=4,…,0; $z_i \in Z_i$)

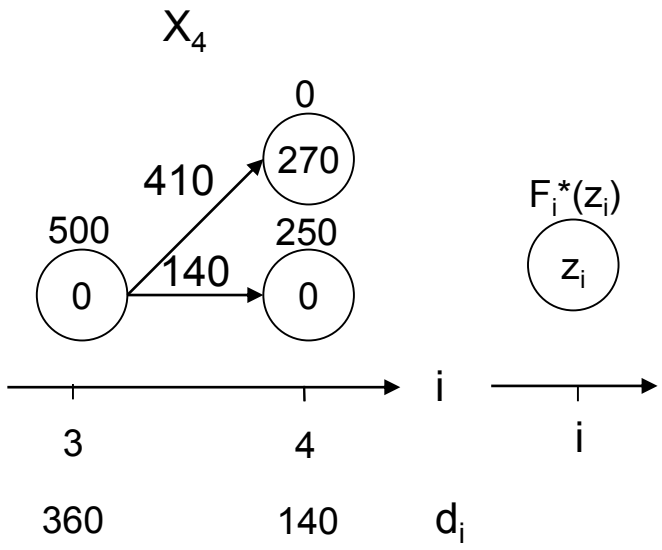
Example: i=3, z_3=0, $x_4 \in \{140, 410\}$



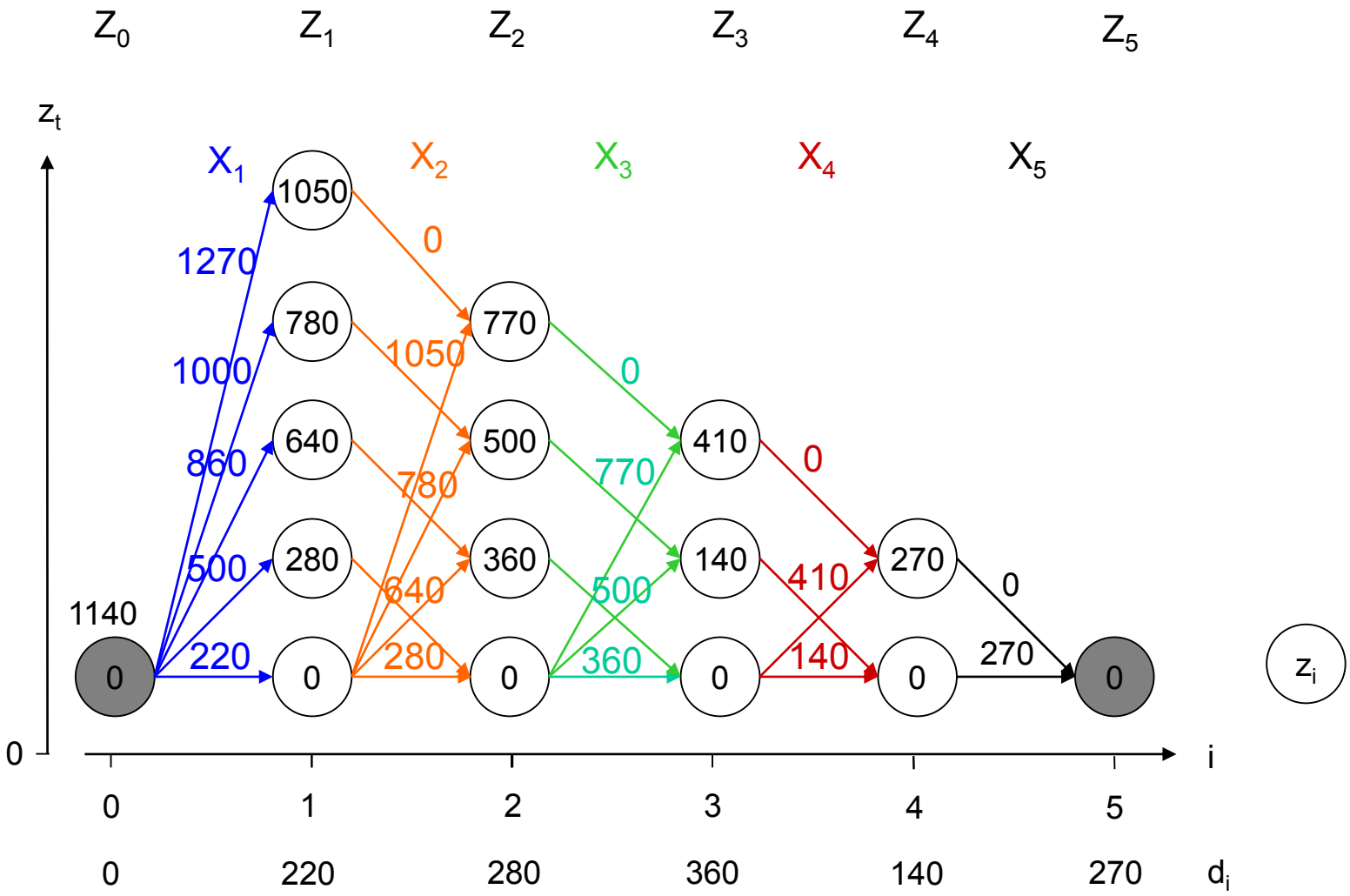
Backward Recursion in Table Format

i=3, z_3=0, $x_4 \in \{140, 410\}$

z_3	x_4	z_4	$f_4(z_4,x_4)$	$F^*_4(z_4)$	$F_3(z_3)$

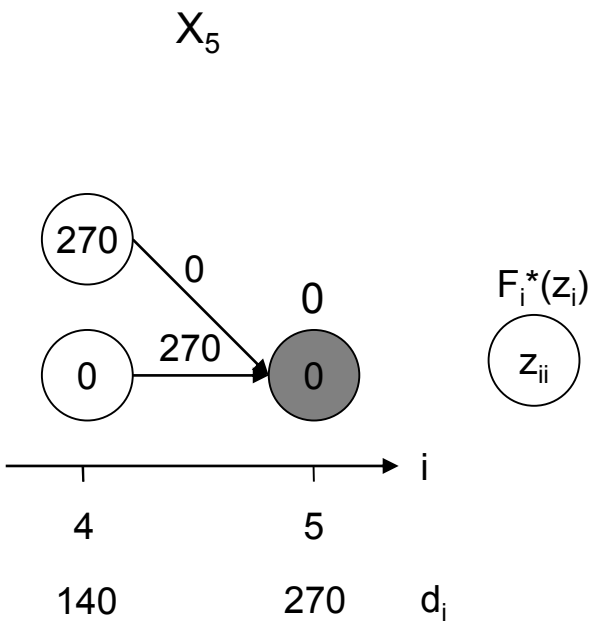


Recursion


Z0
Z1
Z2
Z3
Z4
Z5
zt
X1
X2
X3
X4
X5
1050
780
640
280
0
770
500
360
0
410
140
0
270
0
1140
1270
1000
860
500
220
0
1050
780
640
280
0
770
500
360
0
410
140
0
270
0
zi
i
0
1
2
3
4
5
0
220
280
360
140
270
di

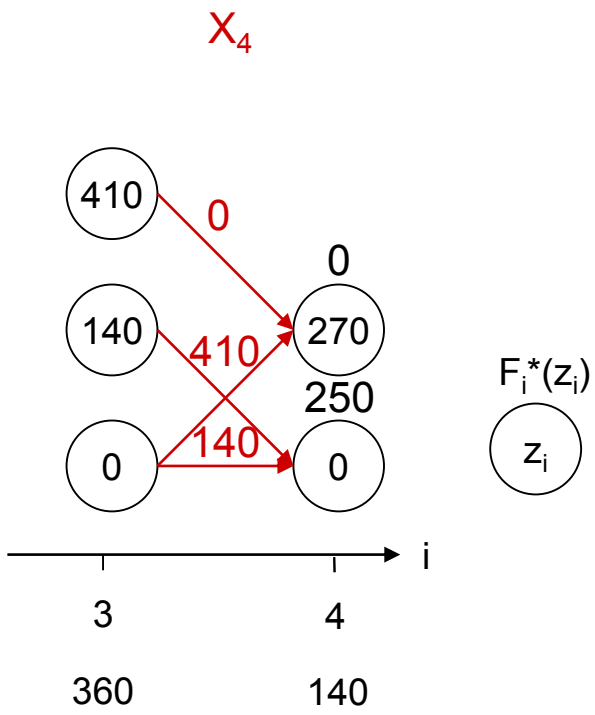
Recursion for i=4

z_4	x_5	z_5	$f_5(z_5,x_5)$	$F^*_5(z_5)$	$F_4(z_4)$

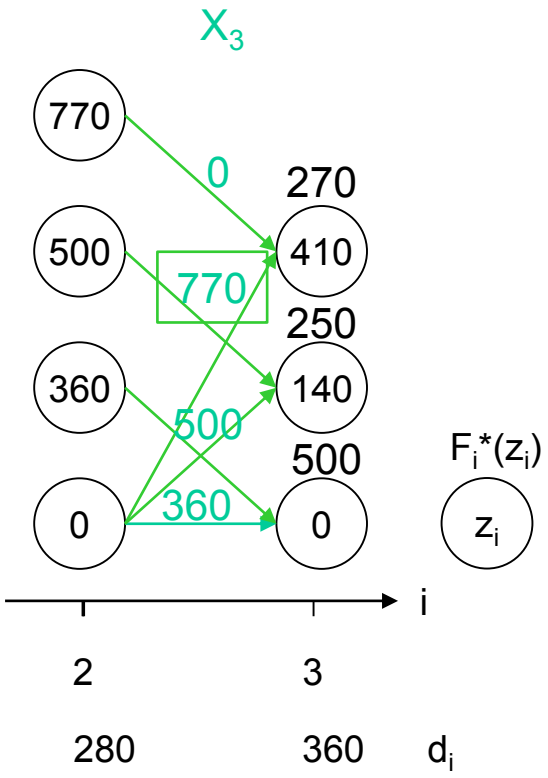


Recursion for i=3

z_3	x_4	z_4	$f_4(z_4,x_4)$	$F^*_4(z_4)$	$F_3(z_3)$

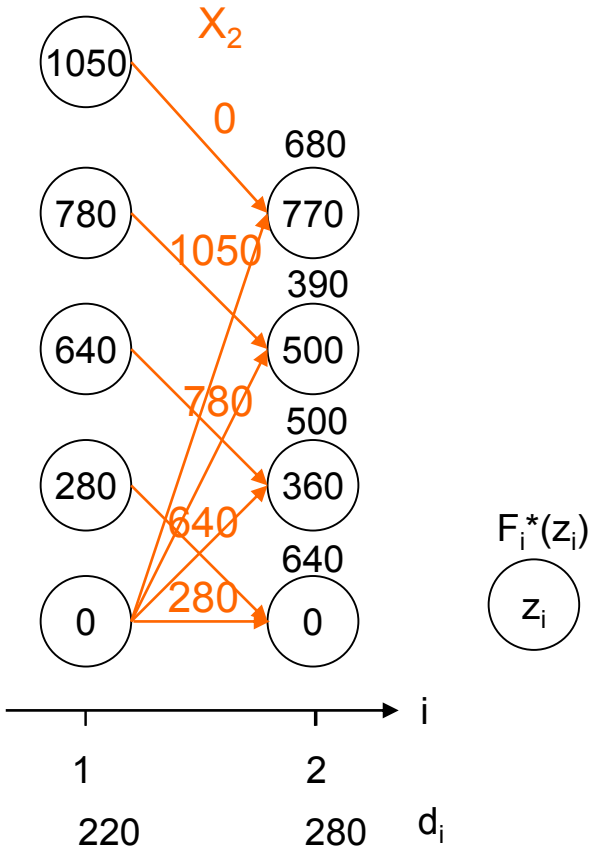


Recursion for i=2



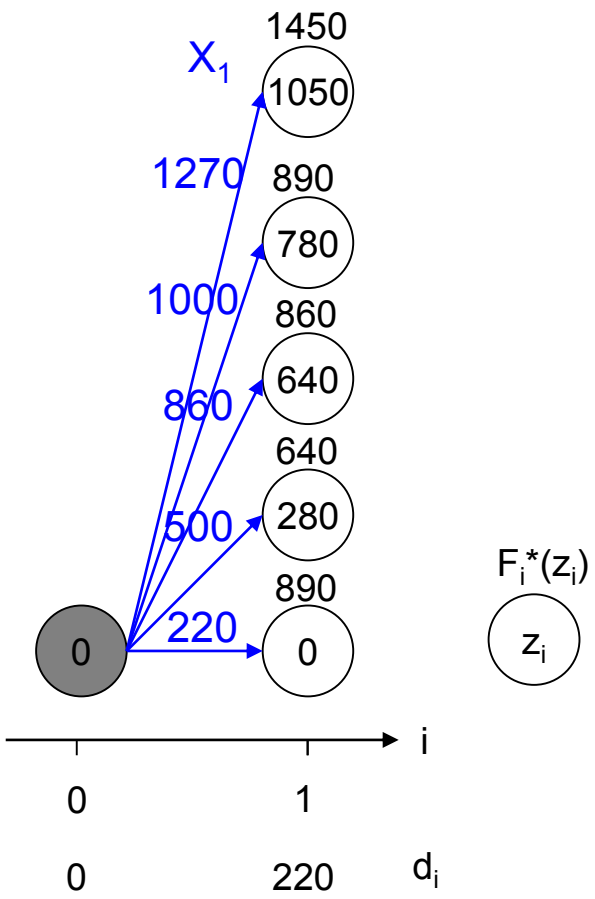
z_2	x_3	z_3	$f_3(z_3,x_3)$	$F^*_3(z_3)$	$F_2(z_2)$

Recursion for i=1



z_1	x_2	z_2	$f_2(z_2,x_2)$	$F^*_2(z_2)$	$F_1(z_1)$

Recursion for i=0



z_0	x_1	z_1	$f_1(z_1,x_1)$	$F^*_1(z_1)$	$F_0(z_0)$

Optimal Policy

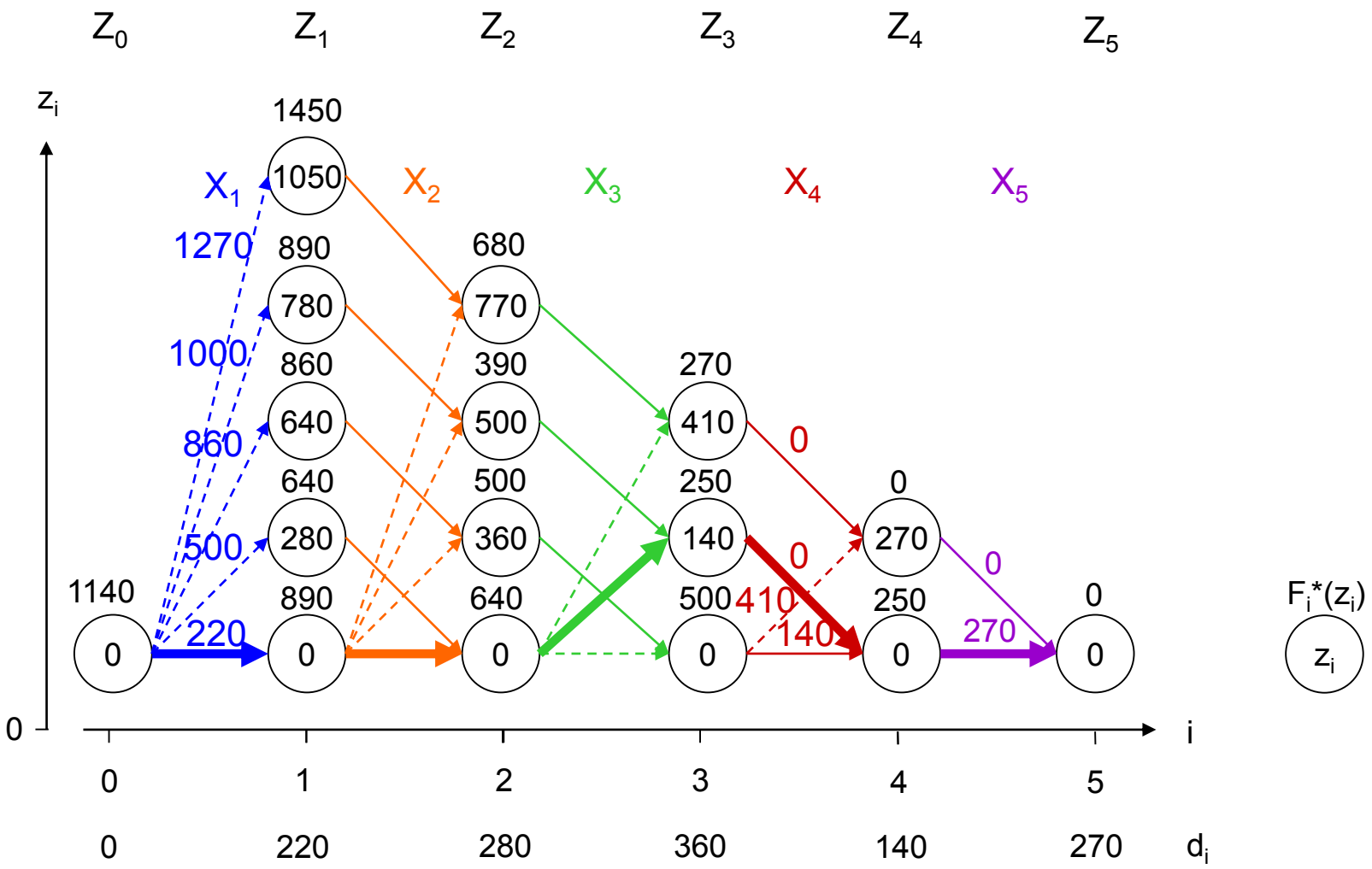
z_4	x_5	z_5	$f_5(z_5, x_5)$	$F_5^*(z_5)$	$F_4(z_4)$	
0	270	0	250	0	$250 + 0 = 250$	$*$
270	0	0	0	0	$0 + 0 = 0$	$*$
z_3	x_4	z_4	$f_4(z_4, x_4)$	$F_4^*(z_4)$	$F_3(z_3)$	
0	140	0	250	250	$250 + 250 = 500$	$*$
0	270+140=410	270	$250 + 270 = 520$	0	$520 + 0 = 520$	
140	0	0	0	250	$0 + 250 = 250$	$*$
410	0	270	270	0	$270 + 0 = 270$	$*$
z_2	x_3	z_3	$f_3(z_3, x_3)$	$F_3^*(z_3)$	$F_2(z_2)$	
0	360	0	250	500	$250 + 500 = 750$	
0	140+360=500	140	250+140=390	250	390+250=640	$*$
0	410+360=770	410	410+250=660	270	660+270=930	
360	0	0	0	500	$0 + 500 = 500$	$*$
500	0	140	140	250	140+250=390	$*$
770	0	410	410	270	410+270=680	$*$
z_1	x_2	z_2	$f_2(z_2, x_2)$	$F_2^*(z_2)$	$F_1(z_1)$	
0	280	0	250	640	$250 + 640 = 890$	$*$
0	360+280=640	360	250+360=610	500	610+500=1.110	
0	500+280=780	500	250+500=750	390	750+390=1.140	
0	770+280=1.050	770	250+770=1.020	680	1.020+680=1.700	
280	0	0	0	640	$0 + 640 = 640$	$*$
640	0	360	0+360=360	500	360+500=860	$*$
780	0	500	500	390	500+390=890	$*$
1.050	0	770	770	680	770+680=1.450	$*$
z_0	x_1	z_1	$f_1(z_1, x_1)$	$F_1^*(z_1)$	$F_0(z_0)$	
0	220	0	250	890	$250 + 890 = 1.140$	$*$
0	280+220=500	280	250+280=530	640	530+640=1.170	
0	640+220=860	640	250+640=890	860	890+860=1.750	
0	780+220=1.000	780	250+780=1.030	890	1.030+890=1.920	
0	1.050+220=1.270	1.050	250+1.050=1.300	1.450	1.300+1.450=2.750	

Optimal Policy

Period i	1	2	3	4	5
Order	220	280	500	0	270

Minimum Cost = 1,140

State Graph and Optimal Policy



6.3 Capital Budgeting Problem

Example

A company considers investing a budget of 14 million Euros into the following four investments, each of which, if selected, must be made once in the full amount.

Investment (i)	Value (c_i) (m. Euros)	Amount (a_i) (m. Euros)
1	16	5
2	22	7
3	12	4
4	8	3

Which investments should the company make in order to maximize the total value?

DP: Stages, States, and Decisions

$n = 4$ There are 4 projects to be considered. (4 Stages).

x_i =1, if project i (in stage i) is selected, or 0 otherwise.

z_i = Total investments made after the decision in stage i, that is, from 1,…,i

X_i Set of possible decisions that depend on state z_{i-1}

State Transition Function and State-dependent Cost Function

State Transition Function

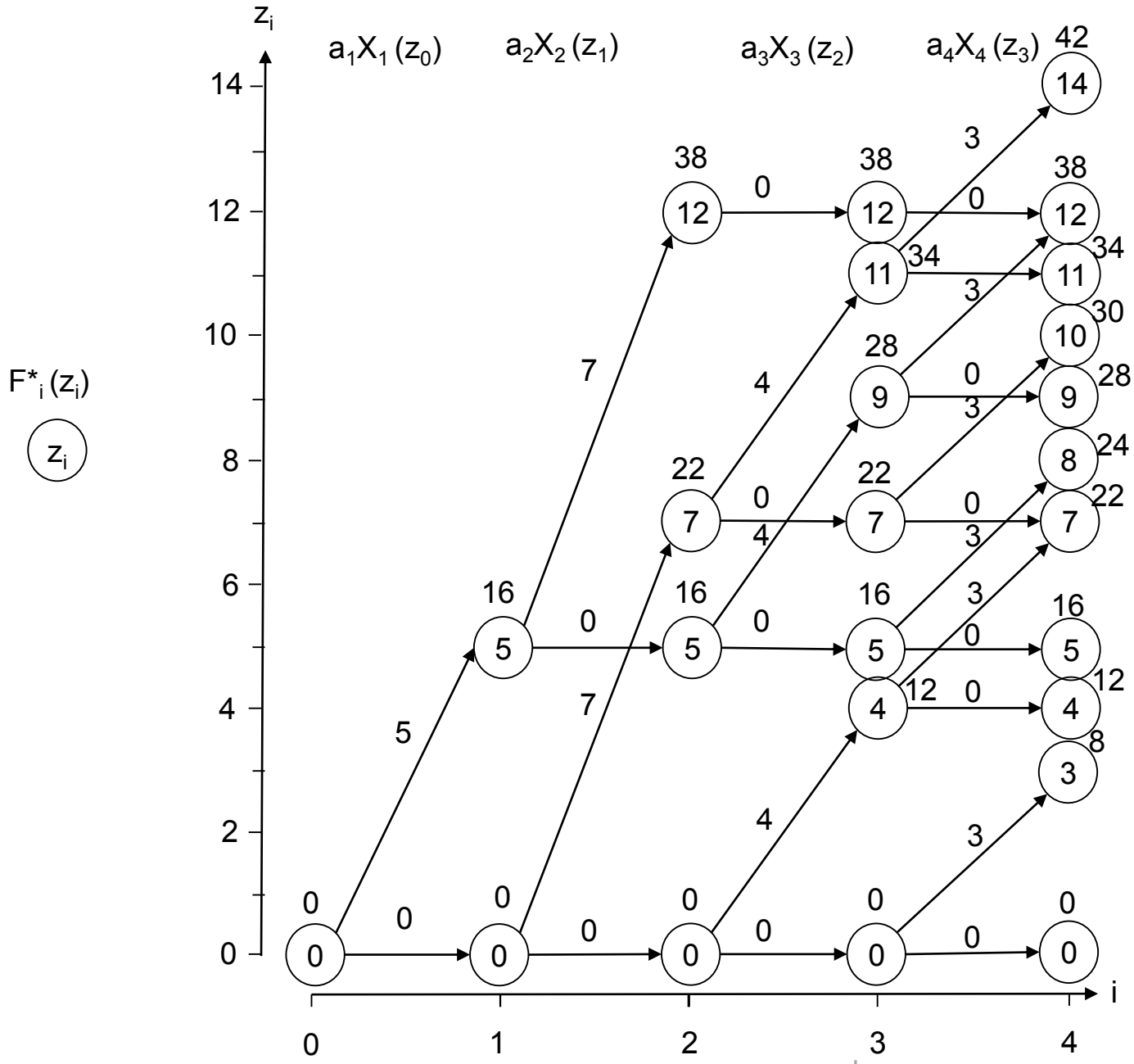
State-dependent Cost Function

Forward Recursion Function and Total Cost Function

Cost incurred in stage i depending on state z_i and decision x_i

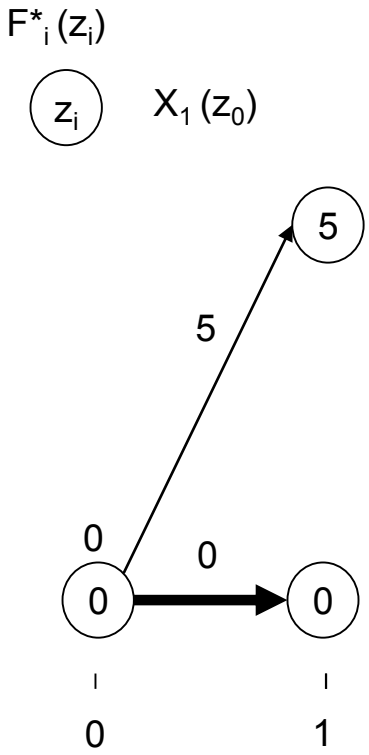
Maximum cost in stage i depending on state z_i

Transition Diagram

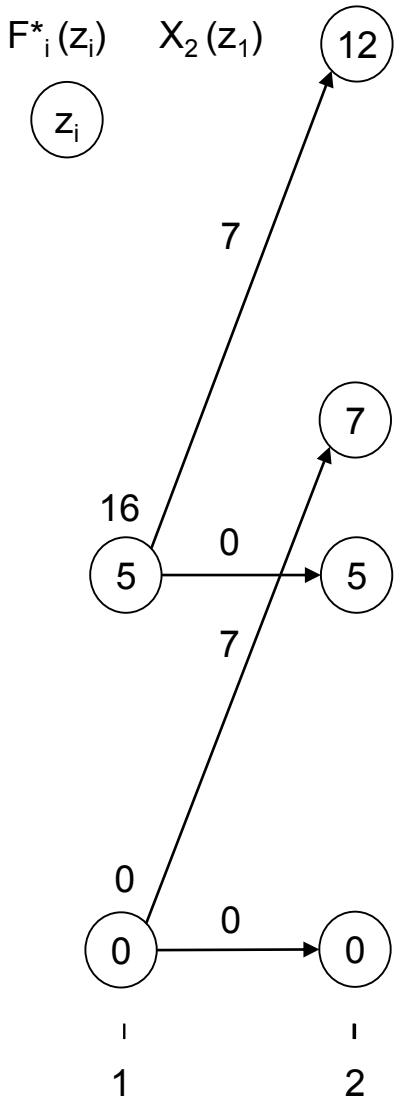


Recursion for i=1

z_1	x_1	z_0	$f_1(x_1)$	$F^*_0(z_0)$	$F_1(z_1)$

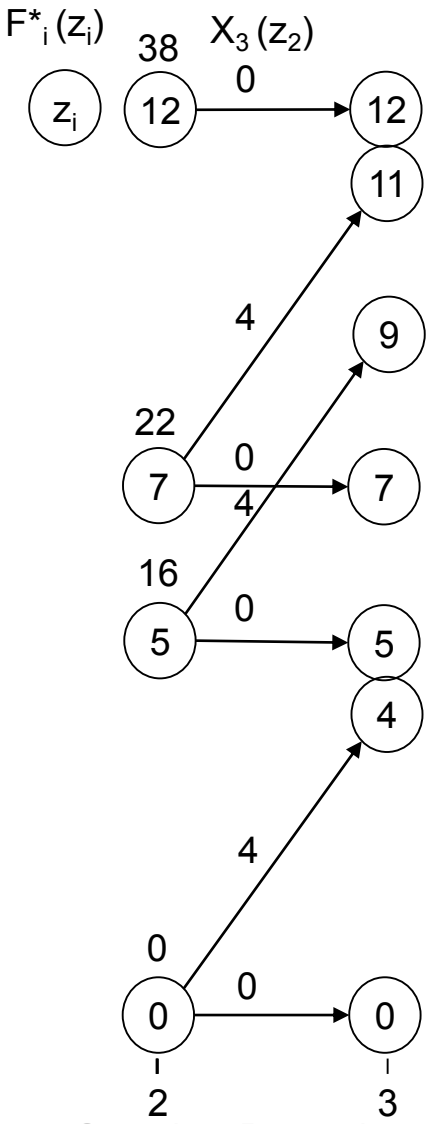


Recursion for i=2



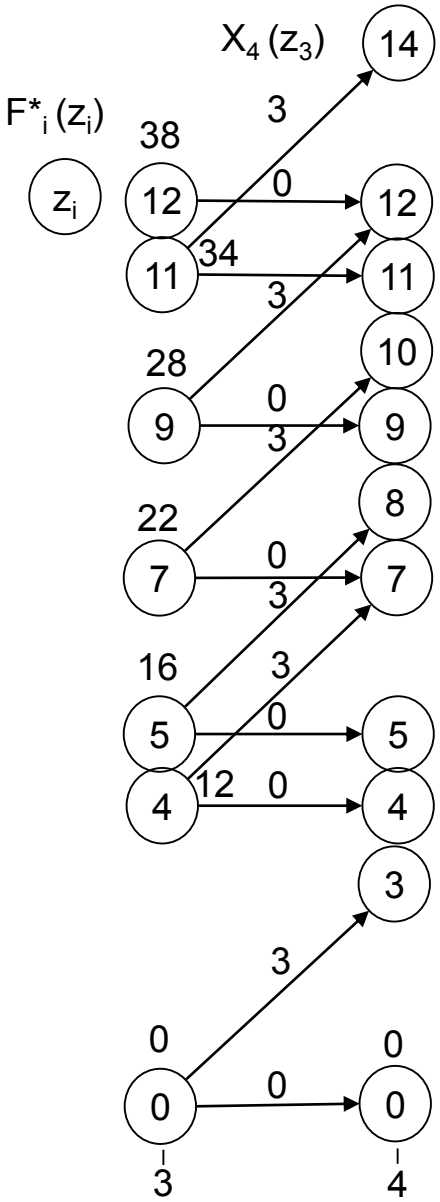
z_2	x_2	z_1	$f_2(x_2)$	$F^*_1(z_1)$	$F_2(z_2)$

Recursion for i=3



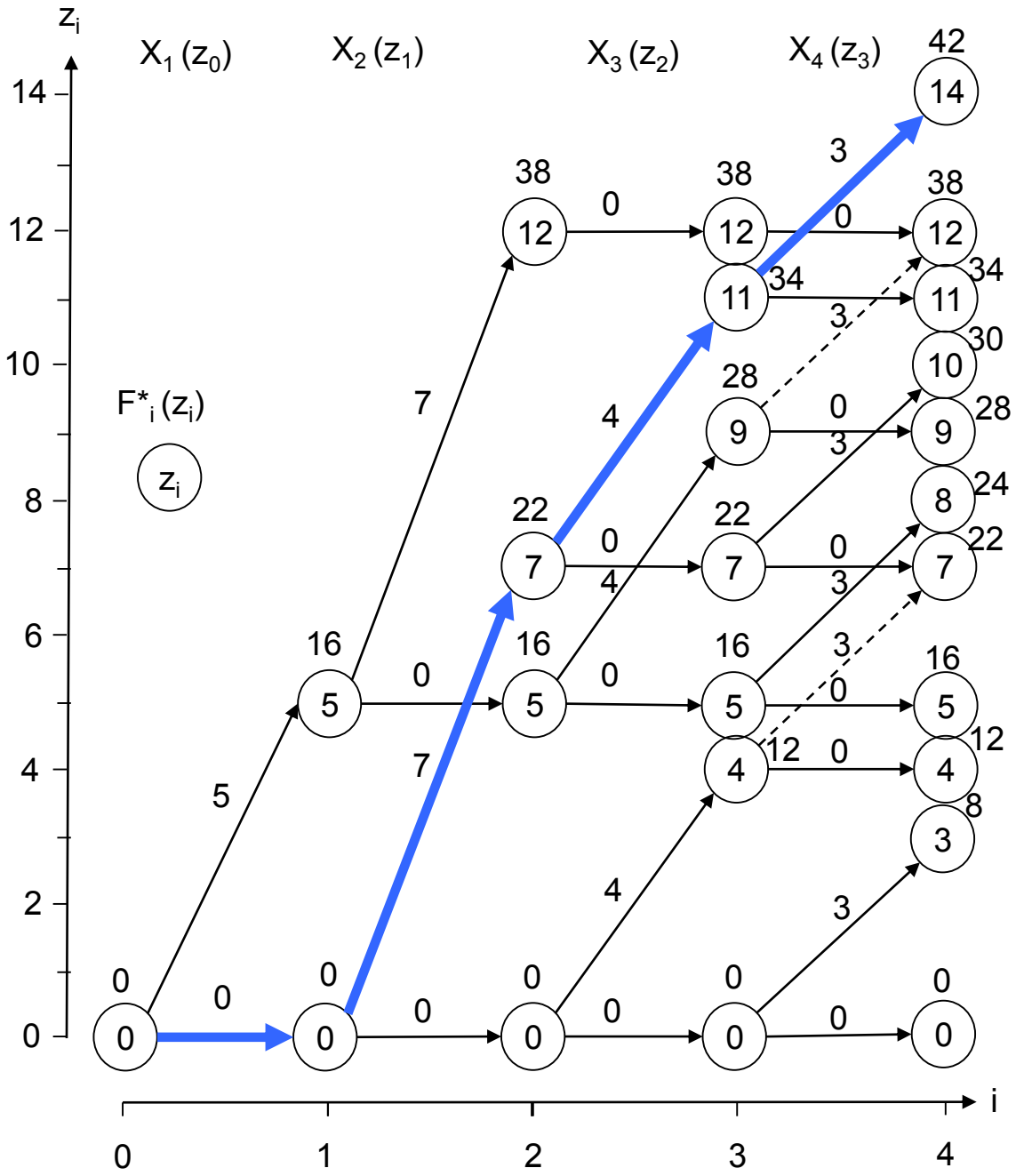
z_3	x_3	z_2	$f_3(x_3)$	$F^*_2(z_2)$	$F_3(z_3)$

Recursion for i=4



z_4	x_4	z_3	$f_4(x_4)$	$F^*_3(z_3)$	$F_4(z_4)$

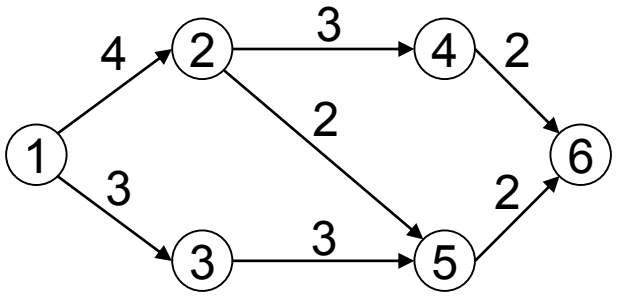
Optimal Policy



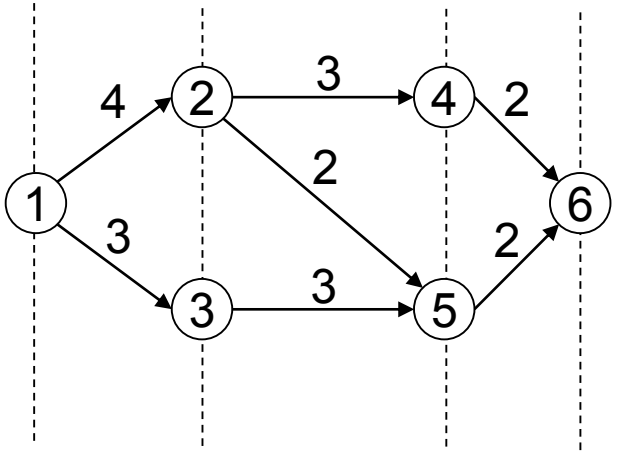
6.4 Solving the Shortest Path Problem with Dynamic Programming

Example

Shortest path problem addressed in Chapter 3.1.2:



Stages and States

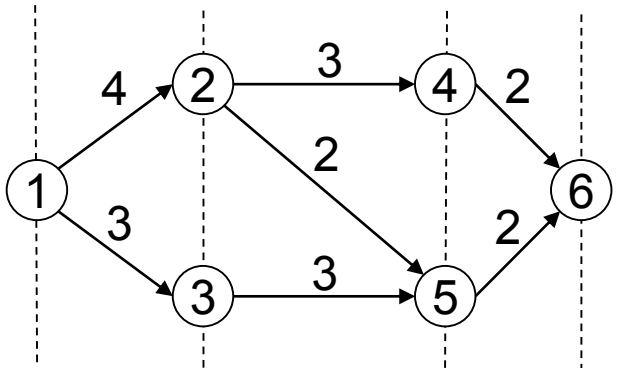


Stages (i=1,…,n) 1 2 3 4

States z_i:

Set of states Z_i:

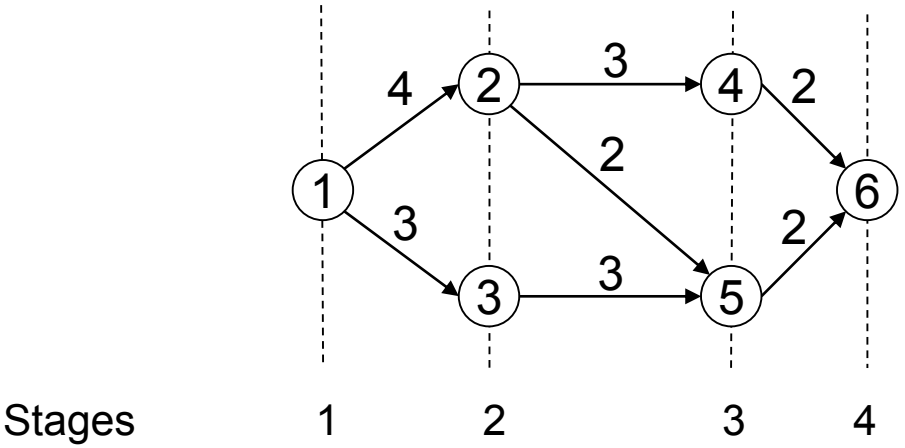
Decisions



Decisions x_i:

Decision set X_i:

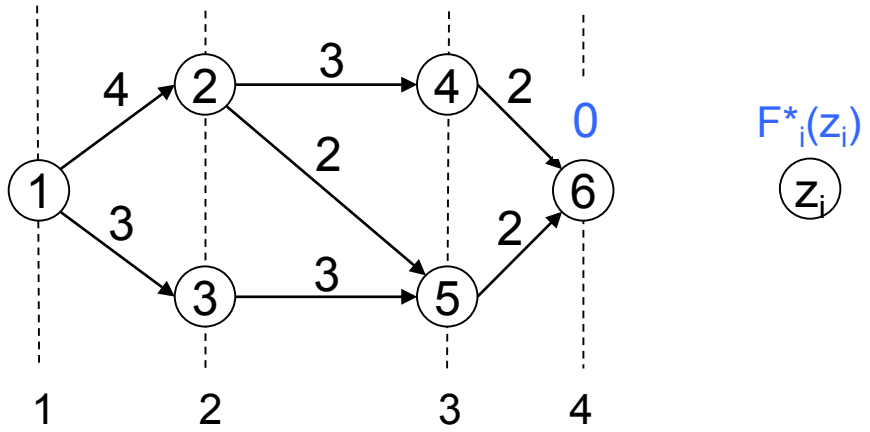
State-Dependent Cost Function and Backward Recursion Function



State-dependent cost function:

Backward recursion function:

Stage i=3

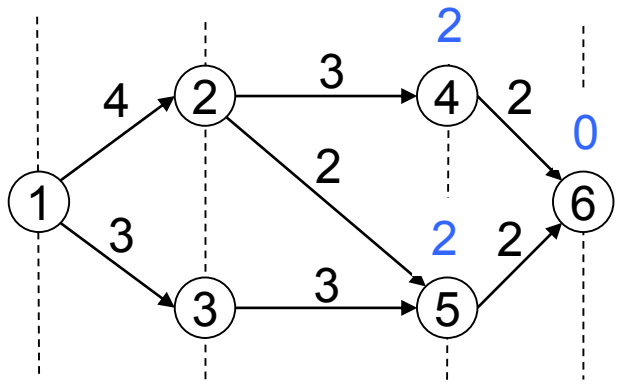


$F^*_i(z_i)$

z_i

z_3	x_3	z_4	$f_3(z_3,z_4)$	$F^*_4(z_4)$	$F_3(z_3)$

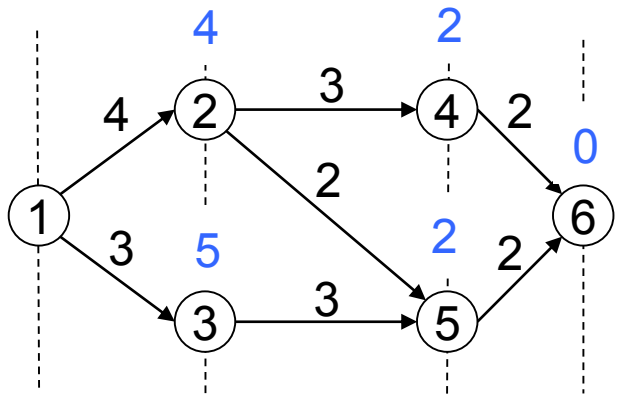
Stage i=2



Stages 1 2 3 4

z_2	x_2	z_3	$f_2(z_2,z_3)$	$F^*_3(z_3)$	$F_2(z_2)$

Stage i=1



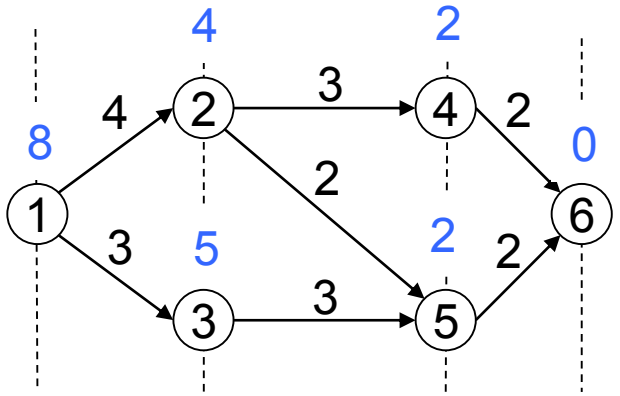
z_1	x_1	z_2	$f_1(z_1,z_2)$	$F^*_2(z_2)$	$F_1(z_1)$

Optimal Policy

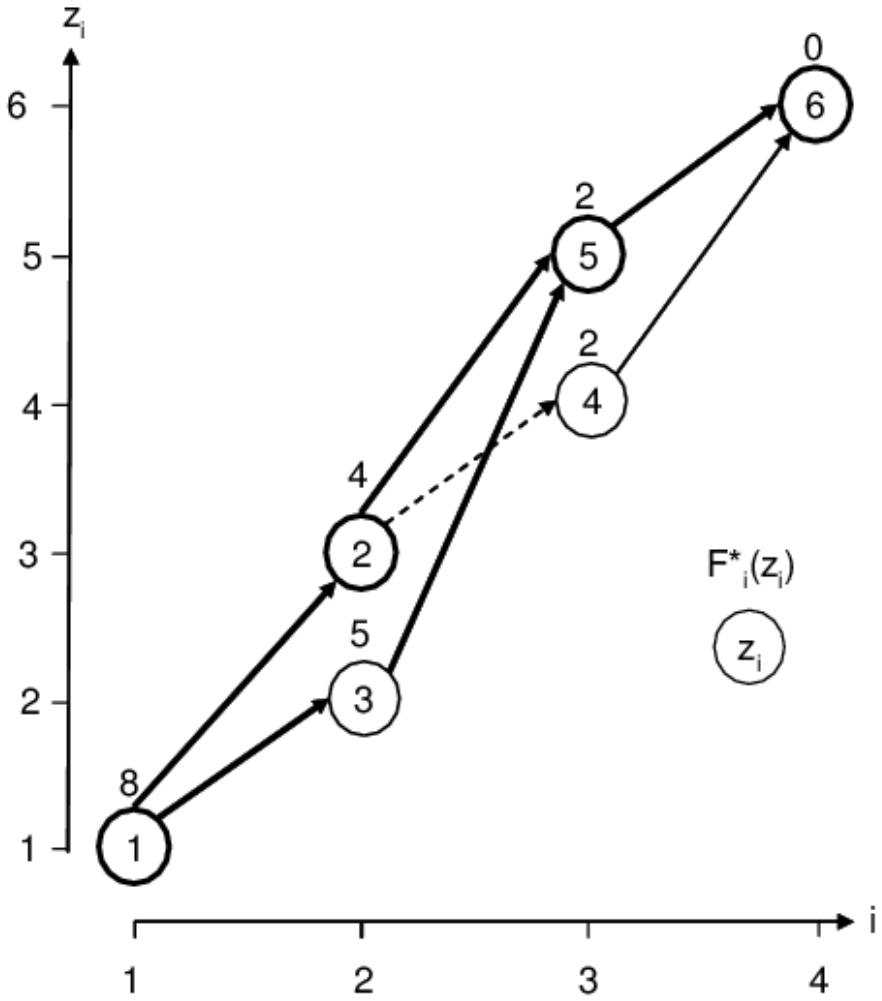
z_3	x_3	z_4	$f_3(z_3, z_4)$	$F_4^*(z_4)$	$F_3(z_3)$	
4	(4,6)	6	2	0	2	*
5	(5,6)	6	2	0	2	*

z_2	x_2	z_3	$f_2(z_2, z_3)$	$F_3^*(z_3)$	$F_2(z_2)$	
2	(2,4)	4	3	2	5	
2	(2,5)	5	2	2	4	*
3	(3,5)	5	3	2	5	*

z_1	x_1	z_2	$f_1(z_1, z_2)$	$F_2^*(z_2)$	$F_1(z_1)$	
1	(1,2)	2	4	4	8	*
1	(1,3)	3	3	5	8	*



State Space



6.5 Resource Allocation

Allocation of Sales Workers

A company wants to introduce a new product into the market, which is divided into three sub-markets. 5 sales staff workers are to be allocated to the three sub-markets. The success in each sub-market depends on the number of allocated sales workers.

	Probability of Success		
Number of staff allocated	**Sub-Market 1**	**Sub-Market 2**	**Sub-Market 3**
0	0.00	0.00	0.00
1	0.60	0.50	0.30
2	0.80	0.70	0.55
3	0.85	0.85	0.70

The total probability of success is the product of the probabilities of success of all three sub-markets. How should the company allocate the sales workers in order to maximize the total probability of success?

Preliminary Considerations

	Probability of Success $p_i(x_i)$		
Workers allocated to market i (x_i)	**Market $i = 1$**	**Market $i = 2$**	**Market $i = 3$**
0	0.00	0.00	0.00
1	0.60	0.50	0.30
2	0.80	0.70	0.55
3	0.85	0.85	0.70

Decision: The number of sales workers x_i allocated to market $i = 1, \dots, 3$.

Objective function:

Minimum number of sales persons for each sub-market:

Maximum number of sales persons per sub-market:

Stages, States and Decisions

$n = 3$ The decisions are to be made in 3 stages or for 3 markets.

x_i = The number of sales persons allocated to market i.

X_i $\in$ {1,2,3}; possible number of salesperson which can be assigned to market i

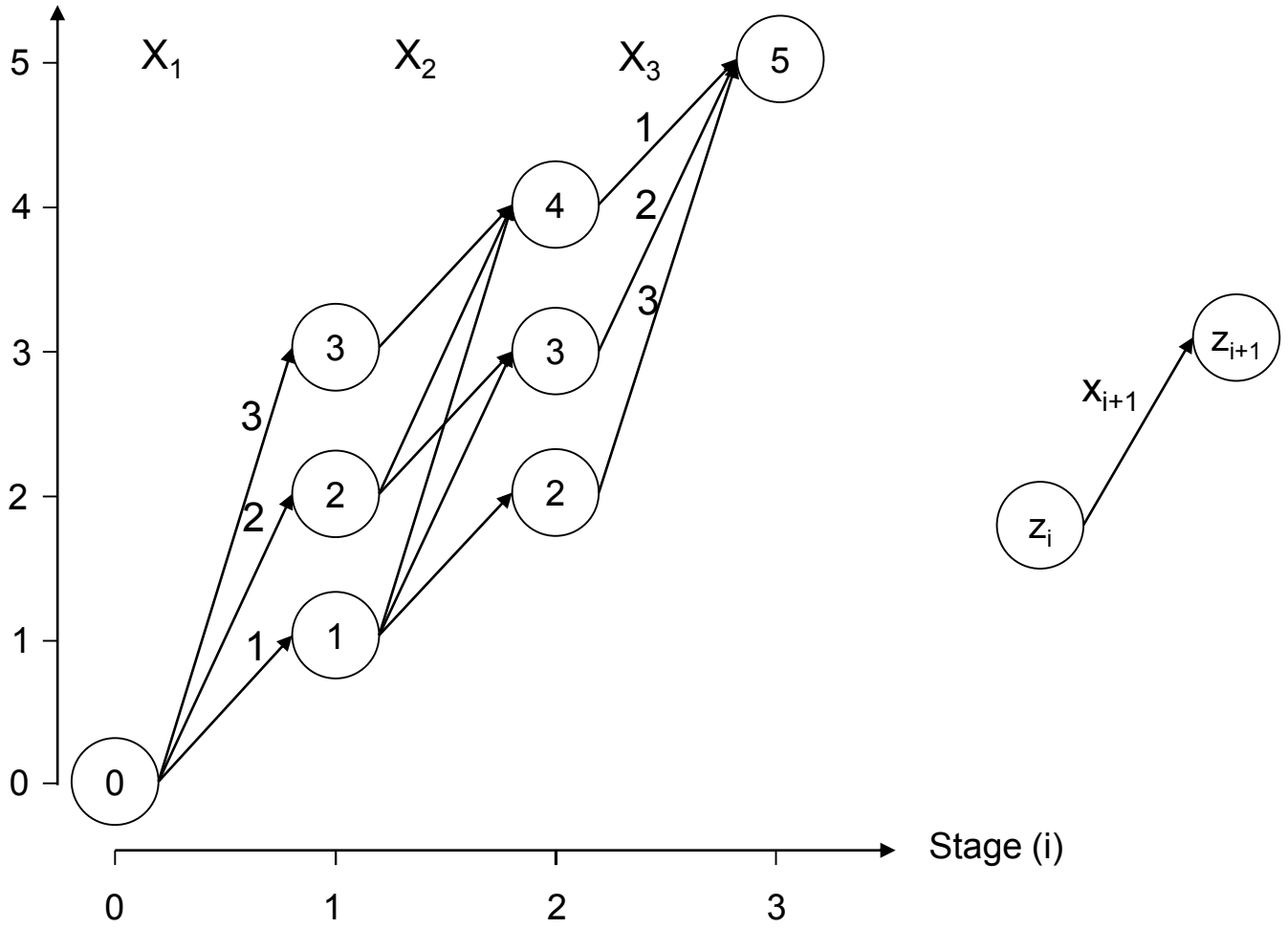
z_i = The number of salespersons available for markets 0 to i

Z_i Set of possible states in stage i

State Transition Diagram

z_i = number of workers available in stages 0,…,i

x_i= number of workers allocated in stage i



State Transition Function
State-Dependent Cost Function

State Transition Function

Forward calculation:

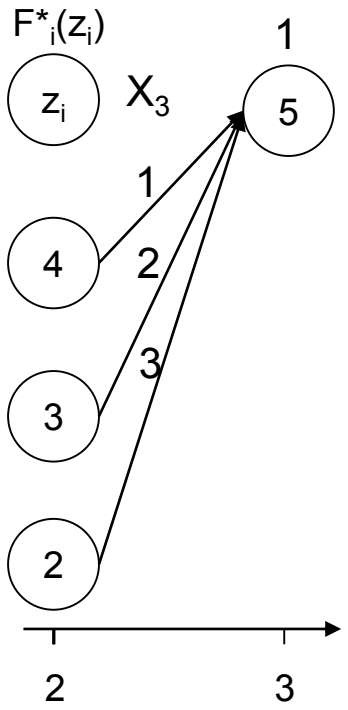
Backward calculation:

State-Dependent Cost Function

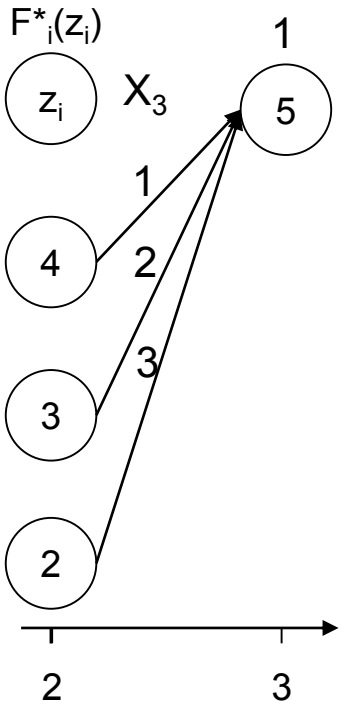
Backward Recursion Function

Success probability of state z_i in stage i with decision x_{i+1}

Maximal success probability of state z_i in stage i

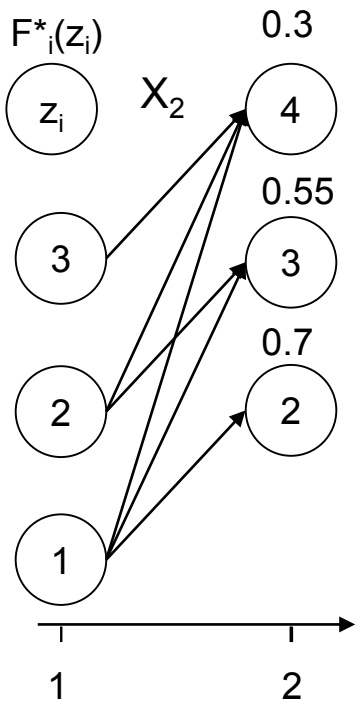


Backward Recursion for i=2



z_2	x_3	z_3	$f_3(x_3)$	$F^*_3(z_3)$	$F_2(z_2)$

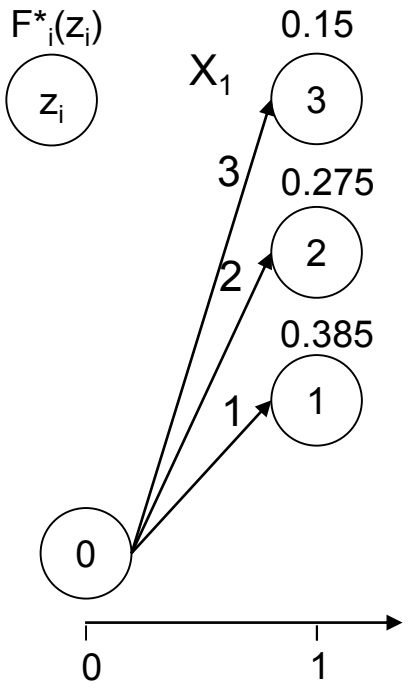
Backward Recursion for i=1



z_1	x_2	z_2	$f_2(x_2)$	$F^*_2(z_2)$	$F_1(z_1)$

Backward Recursion for i=0

z_0	x_1	z_1	$f_1(x_1)$	$F^*_1(z_1)$	$F_0(z_0)$



Optimal Policy

z_i = number of workers available in stages 0,…,i