Lecture 6
Introduction to Dynamic Programming

MATH3220 Operations Research and Logistics
Jan. 27, 2015

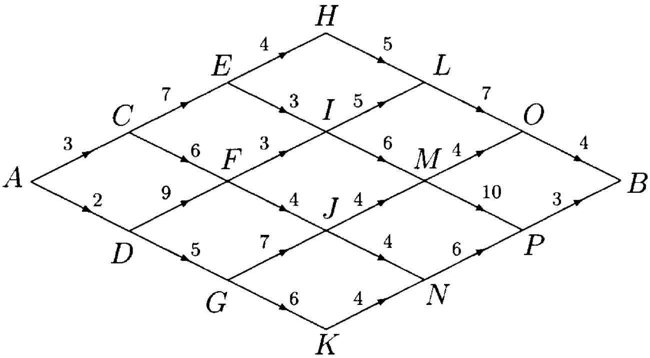
Pan Li
The Chinese University of Hong Kong

Agenda

What is DP?

- *Dynamic Programming* (DP) is an approach that is designed to economize the computational requirements for solving large problems.
- The basic idea in using DP to solve a problem is to split up the problem into a number of stages.
  Each stage is associated with one *subproblem*, and the subproblems are linked together by some form of *recurrence relations*.
  The solution of the whole problem is obtained by solving these subproblems using *recursive computations*.
- Three steps:
  1. Defining subproblems
  2. Finding recurrences
  3. Solving the base cases

A simple path problem - a shortest path from A to B

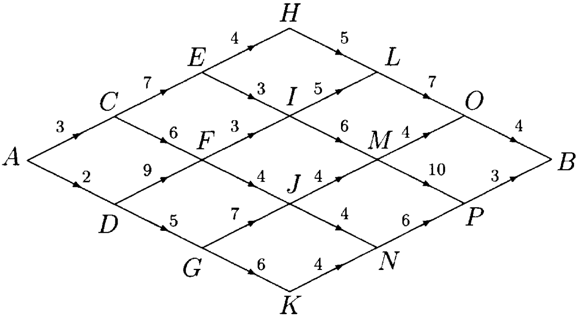


Define

$$S(A) = \text{the length of the shortest path from } A \text{ to } B$$

$$S(i) = \text{the length of the shortest path from } i \text{ to } B$$

A simple path problem - a shortest path from A to B



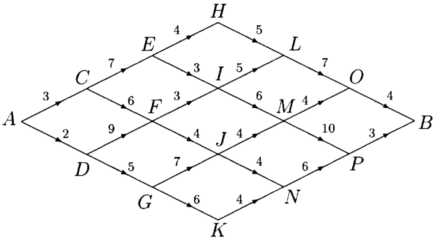
Define

$$S(A) = \text{the length of the shortest path from } A \text{ to } B$$

$$S(i) = \text{the length of the shortest path from } i \text{ to } B$$

$$\Rightarrow S(A) = \min\{a_{AC} + S(C), a_{AD} + S(D)\},$$

A simple path problem - a shortest path from A to B

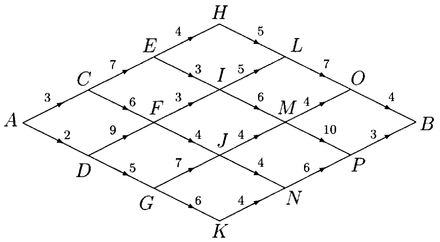


Repeating this argument, we can set up a recurrence relation as follows:

$$
\begin{aligned}
S(A) &= \min\{3 + S(C), 2 + S(D)\} \\
S(C) &= \min\{7 + S(E), 6 + S(F)\} \\
S(D) &= \min\{9 + S(F), 5 + S(G)\} \\
&\vdots \\
S(O) &= 4 + S(B) \\
S(P) &= 3 + S(B).
\end{aligned}
$$

Clearly we can see that $S(B) = 0$.

A simple path problem - a shortest path from A to B

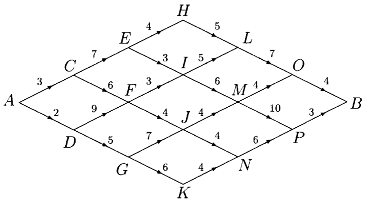


We can compute the values of $S(i)$ recursively by considering nodes further and further away from B:

$$\begin{aligned} S(O) &= 4 + S(B) = 4;\ S(P) = 3 + S(B) = 3 \\ S(L) &= 7 + S(O) = 11;\ S(M) = \min\{4 + S(O), 10 + S(P)\} = 8; \\ S(N) &= 6 + S(P) = 9;\ \cdots \\ S(A) &= \min\{3 + S(C),\ 2 + S(D)\} = 25. \end{aligned}$$

The solution of the simple shortest path problem is now readily seen. The length of the shortest path from A to B is given by $S(A) = 25$.

A simple path problem - a shortest path from A to B



We can compute the values of $S(i)$ recursively by considering nodes further and further away from B:

$$\begin{aligned}
S(O) &= 4 + S(B) = 4, \quad P(O) = B; \\
S(P) &= 3 + S(B) = 3, \quad P(P) = B \\
S(L) &= 7 + S(O) = 11, \quad P(L) = O; \\
S(M) &= \min\{4 + S(O), 10 + S(P)\} = 8, \quad P(M) = O; \\
S(N) &= 6 + S(P) = 9, \quad P(N) = P; \; \cdots \\
S(A) &= \min\{3 + S(C),\; 2 + S(D)\} = 25, \qquad P(A) = C
\end{aligned}$$

The shortest path is obtained by following the direction given by $P(i)$. $P(A) = C$; $P(C) = F$; $P(F) = J$; $P(J) = M$; $P(M) = O$; $P(O) = B$. So the shortest path is $A - C - F - J - M - O - B$

Terminology and Comments

Stage

- The problem can be divided into stages, with a policy decision required at each stage.

- The stages represent different time periods in the problem's planning horizon.
  For example, Inventory problem
- Sometimes the stages do not have time implications.
  For example, shortest path problem

Terminology and Comments

States

- Each stage has a number of states associated with the beginning of that stage.
- The states reflect the information required to fully assess the consequences that the current decision has upon future actions.
  For example, Inventory problem: the inventory level on hand of the commodity
  Shortest path problem: the intersection a commuter is in at a particular stage
- No rules for specifying the states of a system.

Terminology and Comments

- Essential properties that should motivate the selection of states:
  - The state should convey enough information to make future decisions without regard to how the process reached the current state; and
  - The number of state variables should be small.
- The effect of the policy decision at each stage is to transform the current state to a state associated with the beginning of the next stage (possibly according to a probability distribution.)
- The solution procedure is designed to find an optimal policy for the overall problem.

Terminology and Comments

Recursive Optimization

- Recursive optimization procedure builds to a solution of the overall N-stage problem by first solving a one-stage problem and sequentially including one stage at a time and solving one-stage problems until the over optimum has been found.
- Backward induction process,
  forward induction process.
- Basis of the recursive optimization: *principle of optimality*
  Any subpolicy of an optimum policy from any given state must itself be an optimum policy from that state to the terminal states.
  $\Rightarrow$ Given the current state, an optimal policy for the remaining stages is independent of the policy decisions adopted in previous stages.

Terminology and Comments

Optimal value function

- It measures the optimal value for each state at every stage.
- $S(i)$ in Example 1 is called an optimal value function, and i is called the argument of the function.
- There is no fixed rule to define these optimal value functions.

Recurrence relation

- Define a recurrence relation between the values of the optimal value functions.
- Makes DP well suited to compute solutions.
- For example, the recurrence relation at the node A is

$$S(A) = \min\{a_{AC} + S(C), a_{AD} + S(D)\}.$$

a_{AC} denotes the *immediate return* of the *decision* “up". The optimal value (e.g. $S(A)$) is given by choosing the decision that optimizes the sum of the immediate return and the optimal value of the remaining process.

Terminology and Comments

Boundary conditions

- The solution procedure must start with arguments at which the values of the optimal value function are obvious.
- For example, $S(B) = 0$.

Optimal policy function

- The rule that associates the best decision with each subproblem.
- For example, $P(i)$.

Terminology and Comments

- To solve a problem by DP can be described simply as follows:
  1. Define an optimal value function.
  2. Using the principle of optimality, determine a recurrence relation.
  3. Identify the boundary conditions. Starting with the boundary conditions, and using the recurrence relation, determine concurrently the optimal value and policy functions.
  4. Determine the solution of the problem by using the optimal value and policy functions.
- Crux: Choosing a suitable optimal value function for which a recurrence relation can be determined.

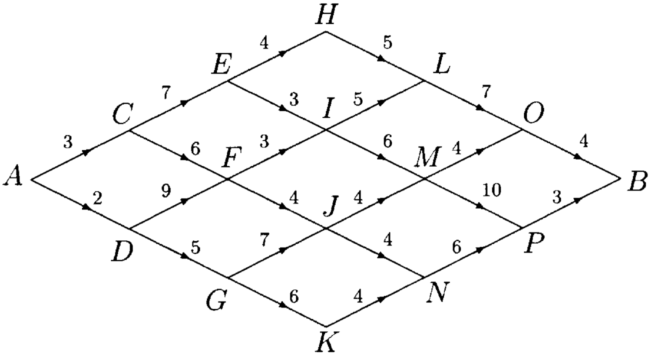
Terminology and Comments

LP vs. DP

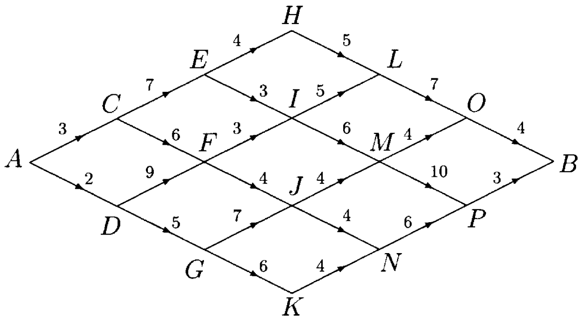
- LP refers to a specific mathematical model that can be solved by a variety of techniques.
- DP deals with a particular analytical approach, which can be applied to a variety of mathematical models.

Example 2

Solve the simple shortest path problem in Example 1 with the optimal value function $T(i)$ defined to be the length of the shortest path from node A to i.

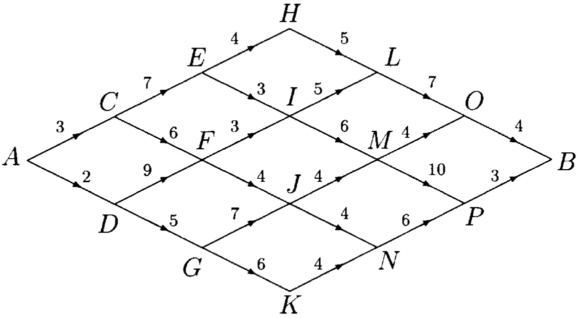


Example 2



$T(i)$ defined to be the length of the shortest path from node A to i.
Answer: $T(B)$ =the length of the shortest path from node A to B
Boundary condition: $T(A) = 0$

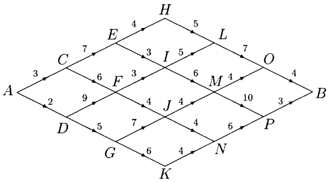
Example 2



Recurrence relation:

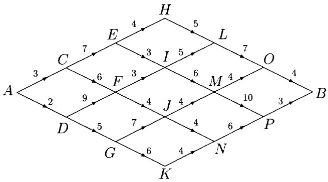
$$T(C) = 3 + T(A), \quad T(D) = 2 + T(A)$$
$$T(E) = 7 + T(C), \quad T(F) = \min\{6 + T(C), 9 + T(D)\}, \quad T(G) = 5 + T(D)$$
$$\vdots$$
$$T(O) = \min\{7 + T(L), 4 + T(M)\}, \quad T(P) = \min\{10 + T(M), 6 + T(N)\}$$
$$T(B) = \min\{4 + T(O), 3 + T(P)\}$$

Example 2



i	$T(i)$	$P(i)$
A	0	A
C	3	A
D	2	A
E	10	C
F	$\min\{6+T(C), 9+T(D)\} = 9$	C
G	$5+T(D) = 7$	D
H	$4+T(E) = 14$	E
I	$\min\{3+T(E), 3+T(F)\} = 12$	F
J	$\min\{4+T(F), 7+T(G)\} = 13$	F
K	$6+T(G) = 13$	G
L	$\min\{5+T(H), 5+T(I)\} = 17$	I
M	$\min\{6+T(I), 4+T(J)\}$=17	J
N	$\min\{4+T(J), 4+T(K)\} = 17$	J/K
O	$\min\{7+T(L), 4+T(M)\} = 21$	M
P	$\min\{10+T(M), 6+T(N)\} = 23$	N
B	$\min\{4+T(O), 3+T(P)\} = 25$	O

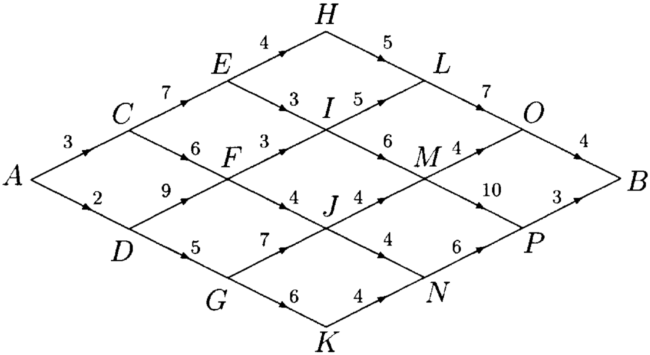
Example 2



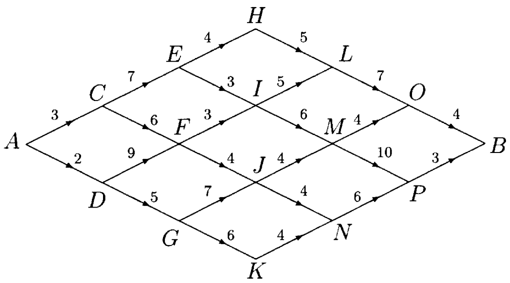
i	$T(i)$	$P(i)$
A	0	A
C	3	A
D	2	A
E	10	C
F	$\min\{6+T(C), 9+T(D)\} = 9$	C
G	$5+T(D) = 7$	D
H	$4+T(E) = 14$	E
I	$\min\{3+T(E), 3+T(F)\} = 12$	F
J	$\min\{4+T(F), 7+T(G)\} = 13$	F
K	$6+T(G) = 13$	G
L	$\min\{5+T(H), 5+T(I)\} = 17$	I
M	$\min\{6+T(I), 4+T(J)\}$=17	J
N	$\min\{4+T(J), 4+T(K)\} = 17$	J/K
O	$\min\{7+T(L), 4+T(M)\} = 21$	M
P	$\min\{10+T(M), 6+T(N)\} = 23$	N
B	$\min\{4+T(O), 3+T(P)\} = 25$	O

Example 3

Solve the simple shortest path problem in Example 1 with the cost of the path to be the largest cost between two nodes on the path.



Example 3

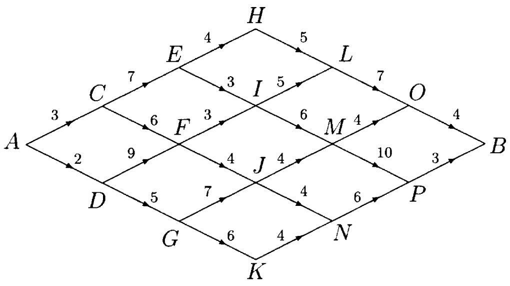


Define
$S(i) =$ the length of the shortest path from i to B

Answer: $S(A)$

Boundary condition: $S(B) = 0$

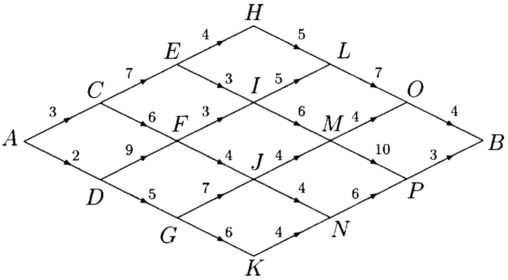
Example 3



Recurrence Relation:

$$S(A) = \min\{\max\{a_{AC}, S(C)\}, \max\{a_{AD}, S(D)\}\}$$

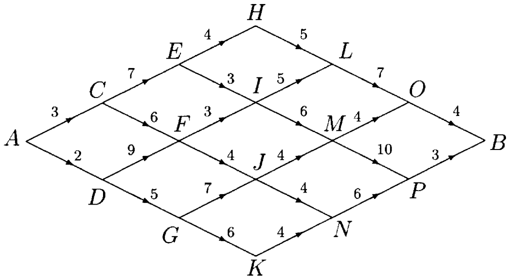
Example 3



Recurrence Relation:

$$S(A) = \min\{\max\{a_{AC}, S(C)\}, \max\{a_{AD}, S(D)\}\}$$
$$S(C) = \min\{\max\{a_{CE}, S(E)\}, \max\{a_{CF}, S(F)\}\}$$
$$S(D) = \min\{\max\{a_{DF}, S(F)\}, \max\{a_{DG}, S(G)\}\}$$
$$\vdots$$
$$S(O) = \max\{a_{OB}, S(B)\}$$
$$S(P) = \max\{a_{PB}, S(B)\}$$

Example 3



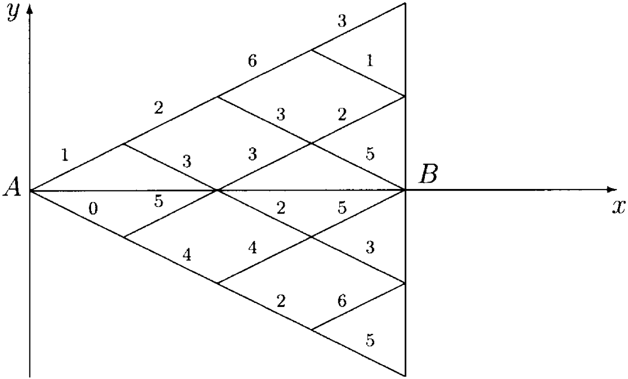
Recurrence Relation:

$$S(A) = \min\{\max\{a_{AC}, S(C)\}, \max\{a_{AD}, S(D)\}\}$$
$$S(C) = \min\{\max\{a_{CE}, S(E)\}, \max\{a_{CF}, S(F)\}\}$$
$$\vdots$$
$$S(O) = \max\{4, S(B)\} = 4, S(P) = \max\{3, S(B)\} = 3$$

$\Rightarrow$ The optimal solution $S(A) = 6$ and the optimal path is $A \to C \to F \to I \to M \to O \to B$ and $A \to D \to G \to K \to N \to P \to B$.

Example 4

We seek the path connecting A with any point on line B which minimizes the total cost, where the number associated with each arc is the cost of traversing that arc.
Assume that admissible paths are always continuous and always move toward the right.



”Give a dynamic-programming formulation" means:

(i) Define an appropriate *optimal value function*, including both a specific definition of its arguments and the meaning of the value of the function.

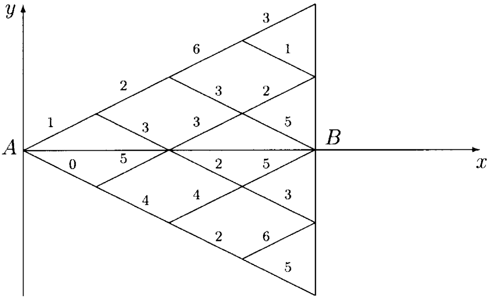
(ii) Write an appropriate *recurrence relation*.

(iii) Define an appropriate *optimal policy function*.

(iv) Note the appropriate *boundary conditions*.

(v) A representation of the answer.

Example 4



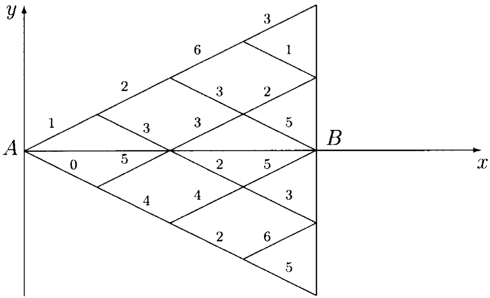
(i) OPTIMAL VALUE FUNCTION: $f(x, y)$ is defined to be the value of the minimum cost path from node (x, y) to any point on line B, for $x = 0, 1, \cdots, 4$ and $y = -x, -x + 2, \cdots, x$.

(ii) RECURRENCE RELATION:

$$f(x, y) = \text{Min}\{a_u(x, y) + f(x+1, y+1), a_d(x, y) + f(x+1, y-1)\}$$

where $a_u(x, y)$ is the cost of the arc that goes upward from (x, y), and $a_d(x, y)$ is the cost of the arc that goes downward.

Example 4

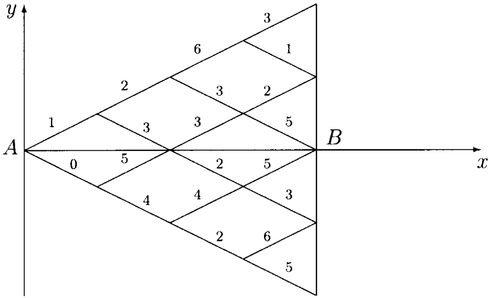


(iii) OPTIMAL POLICY FUNCTION: $P(x, y) = U$ (up) if $a_u(x, y) + f(x + 1, y_1) \leq a_d(x, y) + f(x + 1, y - 1)$, otherwise $P(x, y) = D$ (down). Here U means that the optimum path should go diagonally upward at (x, y) and D means downward.

(iv) BOUNDARY CONDITIONS: $f(4, y) = 0$ for $y = -4, -2, 0, 2, 4$.

(v) ANSWER TO BE SOUGHT: $f(0, 0)$.

Example 4



With the formulation, we get the following solution

$$
\begin{aligned}
f(3,3) &= \min\{3,1\} = 1 && P(3,3) = D \text{ (down)} \\
f(3,1) &= \min\{2,5\} = 2 && P(3,1) = U \\
&\vdots \\
f(0,0) &= \min\{1 + f(1,1), 0 + f(1,-1)\} \\
&= \min\{1 + 7, 0 + 10\} \\
&= 8 && P(0,0) = U
\end{aligned}
$$

Now, since $P(0,0) = U$, $P(1,1) = U$, $P(2,2) = D$, $P(3,1) = U$, the optimal path is $(0,0) \to (1,1) \to (2,2) \to (3,1) \to (4,2)$.

Example 4

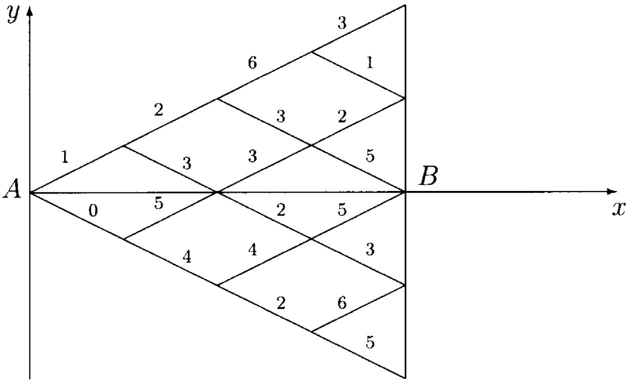
Normally, DP computations are done by computers. In case that the computations are carried out by hand, it is better to write down the results in a tabular form. For the above example, the computation results can be written as in the following table.

x \ y	-4	-3	-2	-1	0	1	2	3	4
4	0	$*$	0	$*$	0	$*$	0	$*$	0
3	$*$	$5D$	$*$	$3D$	$*$	$2U$	$*$	$1D$	$*$
2	$*$	$*$	$7U$	$*$	$5U$	$*$	$5D$	$*$	$*$
1	$*$	$*$	$*$	$10U$	$*$	$7U$	$*$	$*$	$*$
0	$*$	$*$	$*$	$*$	$8U$	$*$	$*$	$*$	$*$

Exercise 1

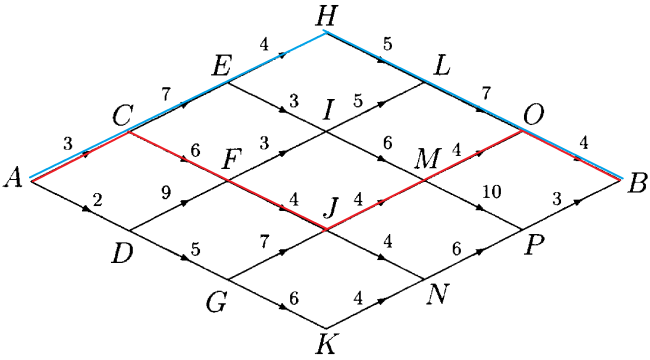
Solve the problem in Figure by using the optimal value function

$S(x, y) =$ the value of the minimum cost path from A to node (x, y).

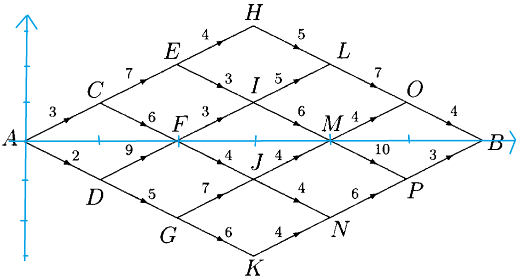


A more complicated example

The numbers associated with the arcs are the costs of traversing these arcs. At any vertex on our way from *A* to *B* if we turn rather than continue in a straight line, an additional cost of 3 is assessed. No penalty is assessed if we continue straight on. Find the shortest path from *A* to *B*.

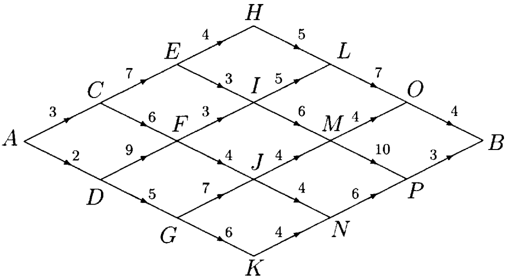


A more complicated example - con't



(i) OPTIMAL VALUE FUNCTION:
$S(x, y, z)$ = the minimum attainable sum of arc numbers plus turn penalties if we start at the vertex (x, y), go to B, and move initially in the direction indicated by z, where z equals 0 denotes diagonally upward and z equals 1 denotes diagonally downward.

A more complicated example - con't



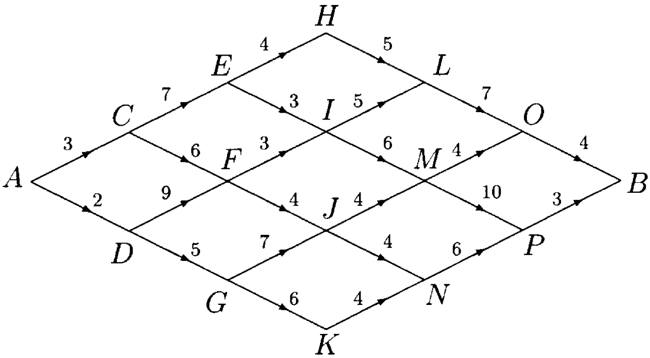
(ii) RECURRENCE RELATION:

$$S(x, y, 0) = a_u(x, y) + \text{Min}\left\{\begin{array}{l} S(x+1, y+1, 0) \\ 3 + S(x+1, y+1, 1) \end{array}\right\}$$

and

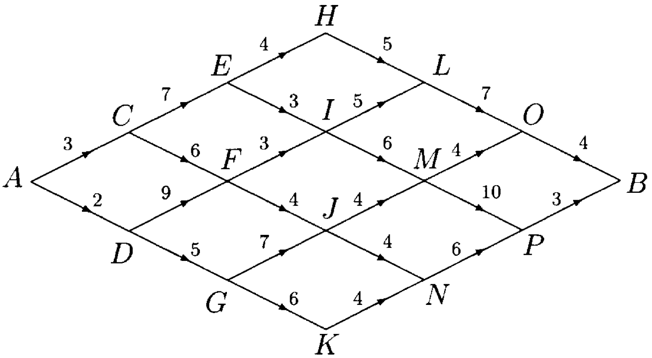
$$S(x, y, 1) = a_d(x, y) + \text{Min}\left\{\begin{array}{l} 3 + S(x+1, y-1, 0) \\ S(x+1, y-1, 1) \end{array}\right\}.$$

A more complicated example - con't



(iii) OPTIMAL POLICY FUNCTION:
$P(x, y, z) = U$ means that 'up' is the optimal <u>second</u> decision if we start at (x, y) and move first in the direction indicated by z.
A similar definition holds for $P(x, y, z) = D$.

A more complicated example - con't



(iv) BOUNDARY CONDITIONS: $S(6,0,0) = 0$ and $S(6,0,1) = 0$.

(v) ANSWER TO BE SOUGHT: $\min\{S(0,0,1), S(0,0,0)\}$.

Solution of the example

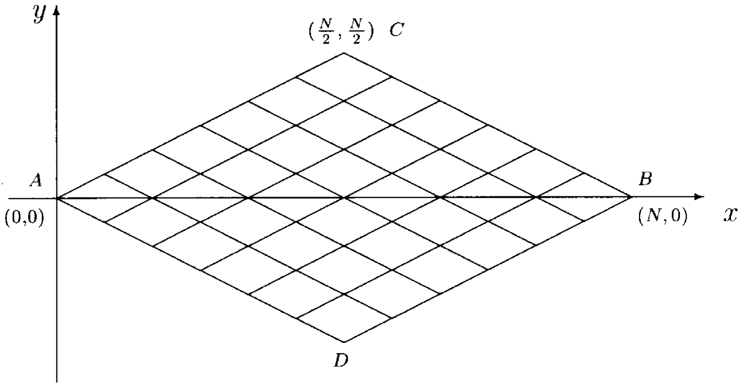
$s(x,y,0)$	y						
x	-3	-2	-1	0	1	2	3
6	*	*	*	0	*	*	*
5	*	*	3U	*	*	*	*
4	*	9U	*	11D	*	*	*
3	13U	*	15U	*	19D	*	*
2	*	22U	*	22U	*	23D	*
1	*	*	31U	*	30U	*	*
0	*	*	*	32D	*	*	*

$s(x,y,1)$	y						
x	-3	-2	-1	0	1	2	3
6	*	*	*	0	*	*	*
5	*	*	*	*	4D	*	*
4	*	*	*	16U	*	11D	*
3	*	*	16U	*	20U	*	16D
2	*	22U	*	20D	*	23D	*
1	*	*	27D	*	26D	*	*
0	*	*	*	29D	*	*	*

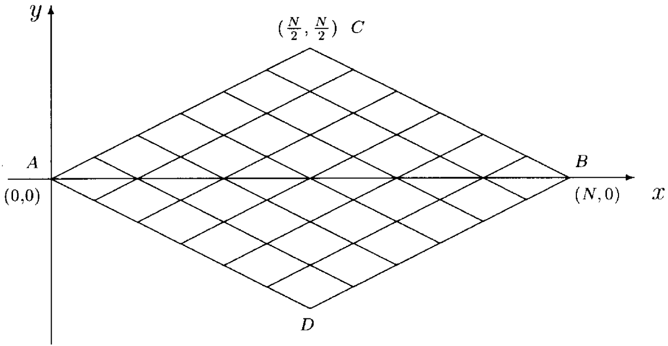
Optimal path: $(0,0,1)29D \to (1,-1,1)27D \to (2,-2,1)22U \to (3,-3,0)13U \to (4,-2,0)9U \to (5,-1,0)3U \to (6,0,0)$

Computational efficiency

In the following figure, where N is assumed to be even, if the shortest path is to be found by using DP approach as in Example 1, find the number of additions and comparisons are required for the dynamic programming solution.

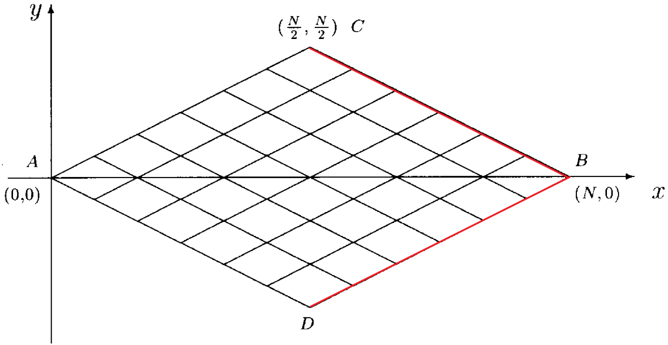


Computational efficiency



$$S(x,y) = \min \left\{ \begin{array}{l} a_u(x,y) + S(x+1, y+1), \\ a_d(x,y) + S(x+1, y-1) \end{array} \right\}$$

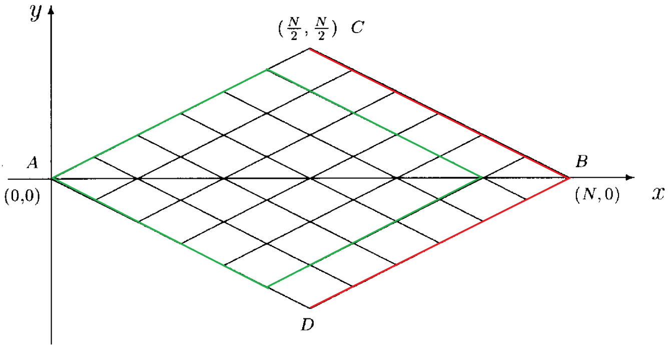
Computational efficiency



$$S(x, y) = \min \left\{ \begin{array}{l} a_u(x, y) + S(x+1, y+1), \\ a_d(x, y) + S(x+1, y-1) \end{array} \right\}$$

(i) there are N vertices (those on the line CB and DB, excluding B) at which one addition and no comparisons are required.

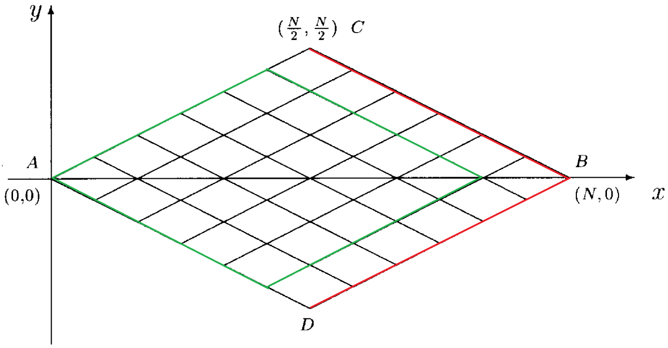
Computational efficiency



$$S(x, y) = \min \left\{ \begin{array}{l} a_u(x, y) + S(x+1, y+1), \\ a_d(x, y) + S(x+1, y-1) \end{array} \right\}$$

(ii) there are $(\frac{N}{2})^2$ remaining vertices at which two additions and one comparison are required.

Computational efficiency



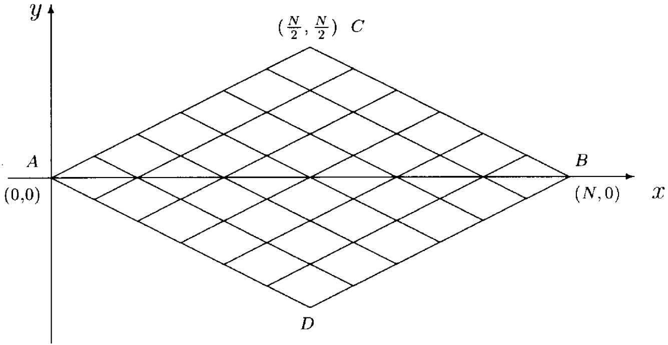
(i) there are N vertices (those on the line CB and DB, excluding B) at which one addition and no comparisons are required.

(ii) there are $(\frac{N}{2})^2$ remaining vertices at which two additions and one comparison are required.

$\Rightarrow$ A total of $N^2/2 + N$ additions and $N^2/4$ comparisons are needed for the dynamic-programming solution.

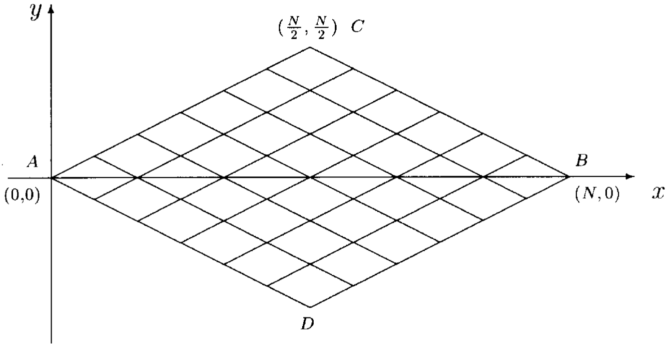
Enumeration method

Consider the number of additions and comparisons using enumeration method for an N-stage problem.



Enumeration method

Consider the number of additions and comparisons using enumeration method for an N-stage problem.



- There $\binom{N}{N/2}$ admissible paths.
- Each path requires $N-1$ additions, and all but the first one evaluated require a comparison in order to find the best path.
- Total $\binom{(N-1)N}{N/2}$ additions and $\binom{N}{N/2-1}$ comparisons.

Computational efficiency

For N-stage shortest path problem:

- DP: $N^2/2 + N$ additions and $N^2/4$ comparisons
- Enumeration: $(N-1)\binom{N}{N/2}$ additions and $\binom{N}{N/2} - 1$ comparisons

If $N = 6$:

- DP: 24 additions and 9 comparisons
- Enumeration: 100 additions and 19 comparisons

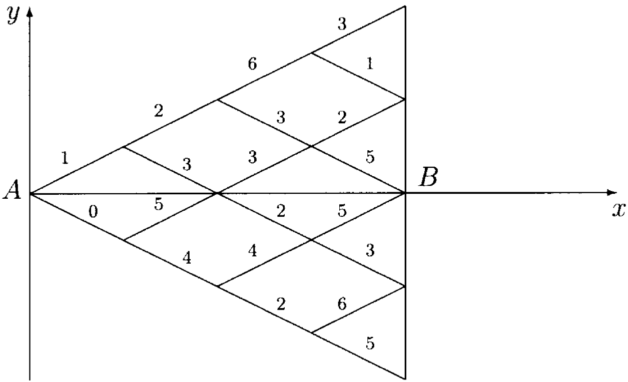
If $N = 20$:

- DP: 220 additions and 100 comparisons
- Enumeration: more than 3 million additions and 184,000 comparisons

$\Rightarrow$ In general, the lager the problem, the more impressive the computational advantage of DP.

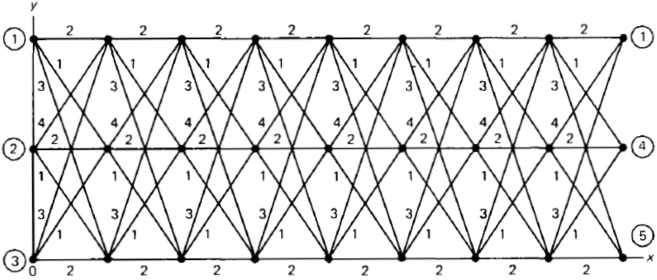
Exercise

How many additions and how many comparisons are required in the DP solution and in enumeration for an N-stage problem involving a network of the type shown in the following figure? Evaluate your formulas for $N = 20$.

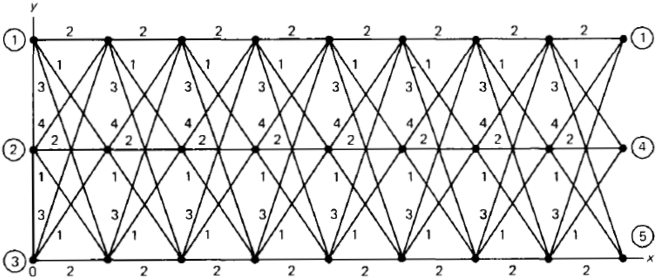


The Doubling-up procedure

Assume all arcs point diagonally to the right, and assume that while as usual the arc costs depend on their initial and final vertices, they do not depend on the stage (i.e., the x coordinate). Such a repeating cost pattern is called stage-invariant and when the stage is often time, it means that costs do not vary with time, only with the nature of the decision. An eight-stage example of such a network with terminal costs as well as arc costs is shown in the following figure.



Goal: Devise a procedure for doubling at each iteration the duration of the problem solved.

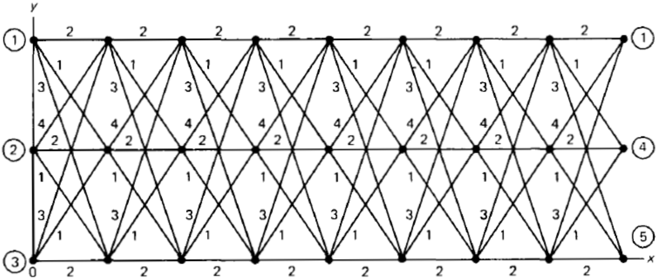


OPTIMAL VALUE FUNCTION:

$S(y_1, y_2, k)$ =the cost (ignoring terminal costs) of the minimum-cost path of length$k$ stages connecting $y = y_1$ and $y = y_2$

We obtain a RECURRENCE RELATION for this function by seeking the optimal value of y at the midpoint of a path of duration $2k$ stages connecting y_1 and y_2. The formula is therefore

$$S(y_1, y_2, 2k) = \min_{y=0,1,2} [S(y_1, y, k) + S(y, y_2, k)]$$



BOUNDARY CONDITION:

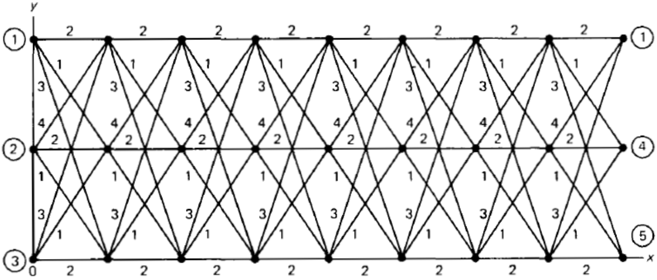
$$S(y_1, y_2, 1) = a(y_1, y_2)$$

where $a(y_1, y_2)$ is the cost of the single arc directly connecting y_1 to y_2 in one step.

ANSWER TO BE SOUGHT:

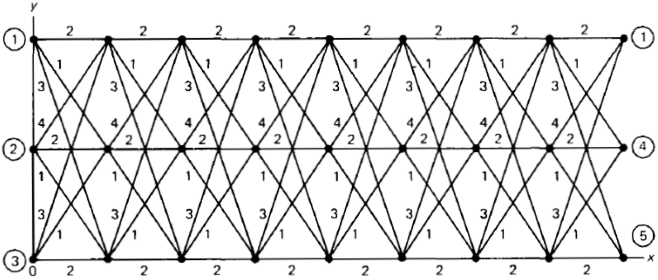
$$\min_{y_1, y_2 = 0,1,2} [t_0(y_1) + S(y_1, y_2, 8) + t_8(y_2)]$$

where the terminal costs are denoted by $t_0(y)$ for $x = 0$ and $t_8(y)$ for $x = 8$.



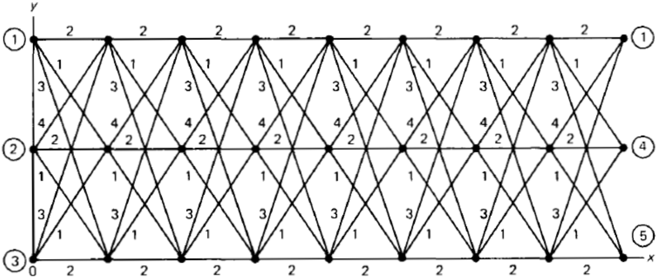
OPTIMAL POLICY FUNCTION:

$P(y_1, y_2, 2k) =$ the midpoint on the best path of duration $2k$ stages connecting y_1 and y_2.



By the Boundary condition, we can have

$S(y_1, y_2, 1)$	y_2		
y_1	0	1	2
0	2	1	3
1	1	2	4
2	3	1	2



Use $S(y_1, y_2, 1)$ and the recurrence relation to obtain the optimal two-stage solution for all pair of end points.

$S(y_1, y_2, 1)$	y_2		
y_1	0	1	2
0	2	1	3
1	1	2	4
2	3	1	2

$$S(y_1, y_2, 2k) = \min_{y=0,1,2}[S(y_1, y, k)+S(y, y_2, k)]$$

$$\begin{aligned} S(0,0,2) &= \min[S(0,0,1)+S(0,0,1), S(0,1,1) \\ &\quad + S(1,0,1), S(0,2,1)+S(2,0,1)] \\ &= \min[4,2,6] = 2, \quad P(0,0,2) = 1 \end{aligned}$$

$$S(y_1, y_2, 2k) = \min_{y=0,1,2} [S(y_1, y, k) + S(y, y_2, k)]$$

$$\begin{aligned}
S(0,0,2) &= \min[S(0,0,1) + S(0,0,1), S(0,1,1) + S(1,0,1), \\
&\quad S(0,2,1) + S(2,0,1)] \\
&= \min[4,2,6] = 2, \quad P(0,0,2) = 1 \\
S(0,1,2) &= \min[S(0,0,1) + S(0,1,1), S(0,1,1) + S(1,1,1), \\
&\quad S(0,2,1) + S(2,1,1)] \\
&= \min[3,3,4] = 3, \quad P(0,1,2) = 0 \text{ or } 1 \\
S(0,2,2) &= \min[2+3, 1+4, 3+2] = 5, \quad P(0,2,2) = 0, 1, \text{ or } 2; \\
S(1,0,2) &= \min[3,3,7] = 3, \quad P(1,0,2) = 0 \text{ or } 1; \\
S(1,1,2) &= \min[2,4,5] = 2, \quad P(1,1,2) = 0; \\
S(1,2,2) &= \min[4,6,6] = 4, \quad P(1,2,2) = 0; \\
S(2,0,2) &= \min[5,2,5] = 2, \quad P(2,0,2) = 1; \\
S(2,1,2) &= \min[4,3,3] = 3, \quad P(2,1,2) = 1 \text{ or } 2; \\
S(2,2,2) &= \min[6,5,4] = 4, \quad P(2,2,2) = 2;
\end{aligned}$$

Using the recurrence relations with $k = 2$ to solve all four-stage problems.

$$\begin{aligned}
S(0,0,4) &= \min[S(0,0,2)+S(0,0,2), S(0,1,2)+S(1,0,2),\\
&\qquad S(0,2,2)+S(2,0,2)]\\
&= \min[2+2, 3+2, 5=2] = 4, \quad P(0,0,4) = 0\\
S(0,1,4) &= \min[2+3, 3+2, 5+3] = 5, \quad P(0,1,4) = 0;\\
S(0,2,4) &= \min[7,7,9] = 7, \quad P(0,2,4) = 0 \text{ or } 1;\\
S(1,0,4) &= \min[5,5,6] = 5, \quad P(1,0,4) = 0 \text{ or } 1;\\
S(1,1,4) &= \min[6,4,7] = 4, \quad P(1,1,4) = 1;\\
S(1,2,4) &= \min[8,6,8] = 6, \quad P(1,2,4) = 1;\\
S(2,0,4) &= \min[4,6,6] = 4, \quad P(2,0,4) = 0;\\
S(2,1,4) &= \min[5,5,7] = 5, \quad P(2,1,4) = 0 \text{ or } 1;\\
S(2,2,4) &= \min[7,7,8] = 7, \quad P(2,2,4) = 0 \text{ or } 1;
\end{aligned}$$

Using the recurrence relations with $k = 4$ to solve all eight-stage problems.

$$\begin{aligned}
S(0,0,8) &= \min[S(0,0,4)+S(0,0,4), S(0,1,4)+S(1,0,4), \\
&\quad S(0,2,4)+S(2,0,4)] \\
&= \min[8,10,11] = 8, \quad P(0,0,8) = 0 \\
S(0,1,8) &= \min[9,9,12] = 9, \quad P(0,1,8) = 0 \text{ or } 1; \\
S(0,2,8) &= \min[11,11,14] = 11, \quad P(0,2,8) = 0 \text{ or } 1; \\
S(1,0,8) &= \min[9,9,10] = 9, \quad P(1,0,8) = 0 \text{ or } 1; \\
S(1,1,8) &= \min[10,8,11] = 8, \quad P(1,1,8) = 1; \\
S(1,2,8) &= \min[12,10,13] = 10, \quad P(1,2,8) = 1; \\
S(2,0,8) &= \min[8,10,11] = 8, \quad P(2,0,8) = 0; \\
S(2,1,8) &= \min[9,9,12] = 9, \quad P(2,1,8) = 0 \text{ or } 1; \\
S(2,2,8) &= \min[11,11,14] = 11, \quad P(2,2,8) = 0 \text{ or } 1;
\end{aligned}$$

Now, using

$$\min_{y_1, y_2 = 0,1,2} [t_0(y_1) + S(y_1, y_2, 8) + t_8(y_2)]$$

to obtain the value of the answers and the optimal choice of the initial and terminal points.

$$\begin{aligned}
\text{answer} = \min[&t_0(0) + S(0,0,8) + t_8(0), t_0(0) + S(0,1,8) + t_8(1),\\
&t_0(0) + S(0,2,8) + t_8(2), t_0(1) + S(1,0,8) + t_8(0),\\
&t_0(1) + S(1,1,8) + t_8(1), t_0(1) + S(1,2,8) + t_8(2),\\
&t_0(2) + S(2,0,8) + t_8(0), t_0(2) + S(2,1,8) + t_8(1),\\
&t_0(2) + S(2,2,8) + t_8(2)]\\
&= \min[3+8+5, 3+9+4, 3+11+1, 2+9+5, 2+8+4,\\
&2+10+1, 1+8+5, 1+9+4, 1+11+1]\\
&= 13 \text{ with } y_1 = 1, y_2 = 2, \text{ and } y_1 = 2, y_2 = 2 \text{ both yielding that value.}
\end{aligned}$$

Construct the optimal path:

$$1-0-1-0-1-0-1-0-2,2-1-0-1-0-1-0-0-2,$$
$$2-1-0-1-0-1-0-1-2,2-1-0-1-0-1-0-2-2,$$
$$2-1-0-1-0-0-1-0-2,2-1-0-1-0-1-1-0-2,$$
$$2-1-0-0-1-0-1-0-2,2-1-0-1-1-0-1-0-2,$$
$$2-1-1-0-1-0-1-0-2,2-2-1-0-1-0-1-0-2,$$

Computational efficiency

Assume the duration is 2^N stages and there are M states at each stage.

For doubling-up,

- Each doubling of k requires M additions for each of M^2 pairs (y_1, y_2).
- Double-up N times to solve the 2^N-stage problem (neglecting the terminal costs).
- So $N \cdot M^3$ additions are needed.

For the usual procedure,

- Each stage requires M^2 additions (M decisions at each of M points).
- So, roughly $2^N \cdot M^2$ additions are needed.

For $N = 3$ and $M = 3$, the usual one-state-variable procedure is slightly better.
But for $N = 4$, doubling-up dominates.
No matter what M is, for large enough N, doubling-up will dominate.

To solve using doubling-up, say 12-stage problem, one can combine $S(y_1, y_2, 8)$ and $S(y_1, y_2, 4)$ by the formula

$$S(y_1, y_2, 12) = \min_y [S(y_1, y, 8)] + S(y, y_2, 4)] \tag{1}$$

Generally, we have the formula

$$S(y_1, y_2, m+n) = \min_y [S(y_1, y, m)] + S(y, y_2, n)] \tag{2}$$

which raises some interesting questions about the minimum number of iterations to get to some given duration N.

Problem:
Using doubling-up and formulas like (1), how many iterations are needed for duration 27? Can you find a procedure using (2) that requires fewer iterations?