Math 5331 Lecture:

Down the Road - DP Models and Applications

Speaker: **Dr. W, Huang**

Example 1 (SPP). Consider the shortest path problem from Node A to Node J in the following network:

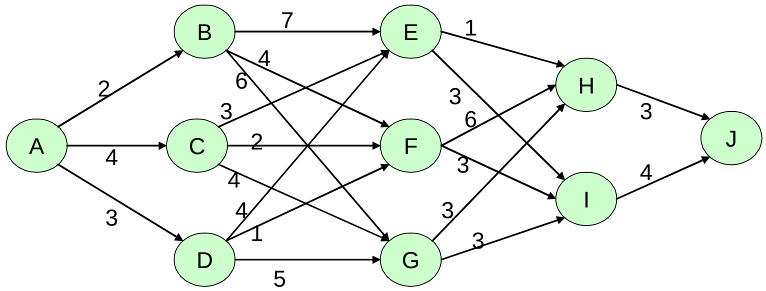

Figure 1: An example of SPP

We will introduce Dynamic Programming (concept and method) based on this example.

An optimization problem can be solved by using dynamic programming approach should have the following characteristics.

- The problem can be divided into T **stages**, say $S_1, S_2, \ldots, S_T$, with a decision to be made at each stage (except for the last stage, slight variation we might see in different textbooks).
- At each stage $t = 1, 2, \ldots, T$, there are N_t **states** (nodes), say $s_1^t, s_2^t, \ldots, s_{N_t}^t$.

  In our example, we have 5 stages:

$$\text{Stage 1}: S_1 = \{s_1^1\} = \{A\}, N_1 = 1;$$
$$\text{Stage 2}: S_2 = \{s_1^2, s_2^2, s_3^2\} = \{B, C, D\}, N_2 = 3;$$
$$\text{Stage 3}: S_3 = \{s_1^3, s_2^3, s_3^3\} = \{E, F, G\}, N_3 = 3;$$
$$\text{Stage 4}: S_4 = \{s_1^4, s_2^4\} = \{H, I\}, N_4 = 2;$$
$$\text{Stage 5}: S_5 = \{s_1^5\} = \{J\}, N_5 = 1.$$

- Therefore we have decision "variables" $x_1, x_2, \ldots, x_{T-1}$, where x_t is a mapping from S_t to S_{t+1}. For example

$$x_3 : \{E, F, G\} \to \{H, I\}.$$

- A solution to the problem is a vector of "functions" $(x_1, x_2, \ldots, x_{T-1})$. For a given solution, we can construct a "path" of states from s_1^1 to s_1^T:

$$s_1^1 \to x_1(s_1^1) \to x_2(x_1(s_1^1)) \to x_3(x_2(x_1(s_1^1))) \to \cdots \to x_{T-1}(\cdots x_2(x_1(s_1^1))) = s_1^T.$$

  Associate with this "path", we are able to calculate the "objective function value", that needs to be minimized or maximized. For example, we may have a solution

$(x_1, x_2, \ldots, x_{T-1})$ of the following form

$$\begin{aligned}
\text{At Stage 1} &: \mathbf{x}_1(A) = \mathbf{B}, \\
\text{At Stage 2} &: \mathbf{x}_2(B) = \mathbf{E}, \\
&: x_2(C) = F, \\
&: x_2(D) = G, \\
\text{At Stage 3} &: \mathbf{x}_3(E) = \mathbf{H}, \\
&: x_3(F) = H, \\
&: x_3(G) = H, \\
\text{At Stage 4} &: \mathbf{x}_4(H) = \mathbf{J}, \\
&: x_4(I) = J.
\end{aligned}$$

The state "path" is

$$A \to B \to E \to H \to J.$$

- Principle of optimality: Given a current state, the optimal decision for each of the remaining stages must **not depend** on previously reached states or previously chosen decisions.
- If the states for the problem have been classified into one of the T stages, there must be a **recursion** that relates the cost or reward earned during stages $t, t = 1, \ldots, T$ to the cost or reward from stages $t+1, t+2, \ldots, T$. More specifically, if we define

  $$\begin{aligned} c_t(x_t, s^t) &= \text{cost paid during Stage } t \text{ from state } s^t \in S_t \text{ using decision } x_t; \\ f_t(s^t) &= \text{minimum cost required from state } s^t \in S_t \text{ to the end } s_1^T. \end{aligned}$$

  The recursion formula is

  $$f_t(s^t) = \min_{x_t(s^t) \in S_{t+1}} [c_t(x_t(s^t), s^t) + f_{t+1}(x_t(s^t))].$$

  In the last example, this recursion formula is

  $$f_t(i) = \min_{j \in S_{t+1}} [c(j,i) + f_{t+1}(j)],$$

  where $c(j,i)$ is the distance from i to j. $f_t(i)$ is the minimum length from node i at Stage t to the destination J.
- The backward DP method. In many case, the problem is considered solved if we have $f_1(s^1)$ for all $s^1 \in S_1$. In our example, $f_1(A)$ gives the length of the shortest path from A to J. When using the backward recursion method, it is usually easy to find $f_{T-1}(s^{T-1}), s^{T-1} \in S_{T-1}$. If not, use a dummy as the destination. Then us the recursion equation to iterate on the backward order until we get to Stage 1:

  $$\begin{aligned} &f_{T-1}(s^{T-1}), \quad s^{T-1} \in S_{T-1}, \\ &f_{T-2}(s^{T-2}), \quad s^{T-2} \in S_{T-2}, \ldots \end{aligned}$$

  In our example, we need to evaluate

$$f_5(J) = 0;$$

$$f_4(H) = \min_{j\in S_5}[c(j,H)+f_5(j)] = c(J,H)+f_5(J) = 3+0 = 3, \text{ and we find } x_4(H) = J;$$

$$f_4(I) = \min_{j\in S_5}[c(j,I)+f_5(j)] = c(J,I)+f_5(J) = 4+0 = 4, \text{ and we find } x_4(I) = J;$$

$$f_3(E) = \min_{j\in\{H,J\}}[c(j,E)+f_4(j)] = 4, \text{ and we find } x_3(E) = H;$$

$$f_3(F) = \min_{j\in\{H,J\}}[c(j,F)+f_4(j)] = 7, \text{ and we find } x_3(F) = I;$$

$$f_3(G) = \min_{j\in\{H,J\}}[c(j,G)+f_4(j)] = 6, \text{ and we find } x_3(G) = H;$$

$$f_2(B) = \min_{j\in\{E,F,G\}}[c(j,B)+f_3(j)] = 11, \text{ and we find } x_2(B) = F \text{ (ties are broken arbitrarily)},$$

$$f_2(C) = \min_{j\in\{E,F,G\}}[c(j,C)+f_3(j)] = 7, \text{ and we find } x_2(C) = E,$$

$$f_2(D) = \min_{j\in\{E,F,G\}}[c(j,D)+f_3(j)] = 8, \text{ and we find } x_2(D) = F \text{ (ties are broken arbitrarily)},$$

$$f_1(A) = \min_{j\in\{B,C,D\}}[c(j,A)+f_2(j)] = 11, \text{ and we find } x_1(A) = C \text{ (ties are broken arbitrarily)}.$$

We now know that the minimum length from A to J is 11. The shortest path can be constructed forwardly

$$A \to x_1(A) = C \to x_2(C) = E \to x_3(E) = H \to x_4(H) = J,$$

or

$$A \to C \to E \to H \to J.$$

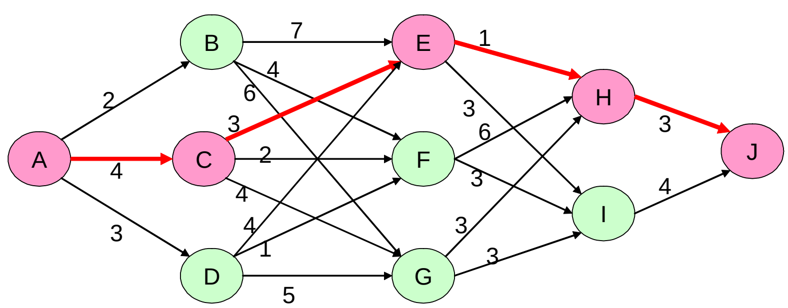

Figure 2. The optimal solution to the SPP

In textbooks, applications of DP include: general SPP, Knapsack problem, Inventory problems, Resource allocation problems, Equipment-replacement problems.

Example 2 (Fishery problem) The owner of a lake must decide how many bass to catch and sell each year. If she sells x bass during year t, then a revenue $r(x)$ is earned. The cost of catching x bass during the year is a function $c(x,b)$, where b is the total number of bass in the lake at the beginning of the year. Reproduction rate of bass is 20% per year. There are $10,000$ bass in the lake at the beginning of year 1.

Develop a DP recursion that can be used to maximize the owner's net profit over a

T-year horizon.

Stages: Let years be the stages (naturally), because decisions are to be made at the beginning of every year, $t = 1, 2, 3, \ldots, T$.

States: What information or "natural condition" at each stage that will help us to make decision?

- the number of bass in the lake at the beginning of the year;
- catching cost for each year if variable;
- market price of bass, if variable.

Let b be the number of bass in the lake at the beginning of the year.

Decision variables: What is the decision variable, or controllable variable? (Our action) the number of bass to catch during the year. Let $x_t(b)$ be the number of bass to catch during year t if there are b bass in the lake at the beginning of the year. After the decision is made, we are to be transferred to the next stage with calculable number of bass in the lake.

Objective: Maximize the total net profit over the T years.

Recursion equation? Let

$f_t(b) =$ the maximum net profit that can be obtained from year t to T if there are b basses in the lake at the beginning of year t

$p_t(x, b) =$ net profit from catching and selling x bass during year t with b basses in the lake at the beginning of the year.

Then for $t = 1, 2, \ldots, T$, we have

$$f_t(b) = \max_{x \in [0,b]} \{p_t(x, b) + f_{t+1}(1.2(b - x))\}$$
$$\max_{x \in [0,b]} \{r(x) - c(x, b) + f_{t+1}(1.2(b - x))\}.$$

Solve the problem backward: starting from solving $f_T(b), b \geq 0.$ And we need find the value of $f_1(10000)$.

- At stage T: to find $f_T(b)$, for all $b \geq 0$, we solve
$$\max_{x \in [0,b]} \{r(x) - c(x, b)\},$$
we may denote the optimal solution by $x_T(b)$, where b can be considered as a parameter. Maximum possible b is $10000(1.2)T$. Many!!
- At stage $T - 1$: to find $f_{T-1}(b)$, for all $b \geq 0$, we solve

$$\max_{x\in[0,b]}\{r(x)-c(x,b)+f_T(1.2(b-x)\},$$

We need to write a computer program!!