

F

Linear Programming Algorithm without Discounting

Optimizes the "average", i.e., expected, cost or return per period in steady state.



Linear Programming Algorithm with Discounting

Optimizes the present value of all future expected costs

@Dennis Bricker, U. of Iowa, 1998

LP model of MDP

Assume that, using the optimal policy, a steady state distribution exists.

Define "randomized" or "mixed" strategies:

 X_i^k = joint probability, in steady state, of being in state i and selecting action k ε K_i



@Dennis Bricker, U. of Iowa, 1998

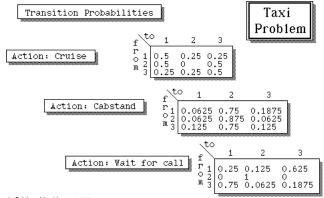


Maximize $\sum_{i \in S} \sum_{k \in K_i} C_i^k X_i^k$

Cost Matrix

@Dennis Bricker, U. of Iowa, 1998

name



© 2 0 1 0 .75 0 SDennis Bricker, U. of Iowa, 1998

Cruise 78 716 77
Cabstand 72.75 715 74
Wait for call 74.25 999 74.5

(Rows ~ actions, Columns ~ states)

A value of 999 above signals an infeasible action in a state.

Expected returns for each i&k

Taxi

Problem

@Dennis Bricker, U. of Iowa, 1998

LP Tableau

Initial basic feasible solution

basic: ★ ★ ★

k: 1 2 3 1 2 1 2 3 R

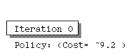
i: 1 1 1 1 2 2 3 3 3 3 3 8

Min 0 2.1 5.01667 0 74.95 0 70.566667 3.23333 9.2

1 1.45 1.36667 0 0.35 0 0.0333333 70.616667 0.4
0 70.4 0.1 1 0.3 0 70.4 0.15 0.2
0 70.05 70.466667 0 0.35 1 1.36667 1.46667 0.4

Iteration 0

Initial policy: in each city, select "cruise" ("greedy" policy)



| DASIC ₂ | $\begin{cases} X_1^1 = 0.4 \\ X_2^1 = 0.2 \end{cases}$ |
|--------------------|--|
| solution | $X_3^2 = 0.2$ |

| State | Action | P{i} |
|----------|----------|------|
| 1 Town A | 1 Cruise | 0.4 |
| 2 Town B | 1 Cruise | 0.2 |
| 3 Town C | 1 Cruise | 0.4 |

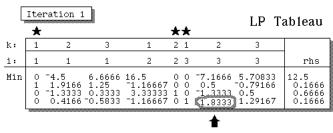
Initial policy: in each city, select "cruise" ("greedy" policy)

@Dennis Bricker, U. of Iowa, 1998

| | Iteration 1 ★★ | | | | | | | LP T | ableau |
|-----|------------------|-------------------------------------|-------------------------------------|---------------------|------------------|---|-------------------------------------|---------------------------------------|------------------------------------|
| k: | 1 | 2 | 3 | 1 | 2 | 1 | 2 | 3 | |
| i: | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | rhs |
| Min | 0 1 0 0 | -4.5 1.9166 -1.3333 0.4166 | 6.6666 1.25 0.3333 -0.5833 | -1.16667 3.33333 | 0 0 1 0 | 0 | -7.1666 0.5 -1.3333 1.8333 | 5.70833 -0.79166 0.5 1.29167 | 12.5 0.1666 0.6666 0.1666 |

$$\begin{array}{l} \textit{basic} \\ \textit{solution} \end{array} \begin{cases} \begin{array}{l} \chi_1^1 = \ \frac{1}{6} \\ \chi_2^2 = \ \frac{2}{3} \\ \chi_3^1 = \ \frac{1}{6} \end{array} \end{cases}$$

@Dennis Bricker, U. of Iowa, 1998

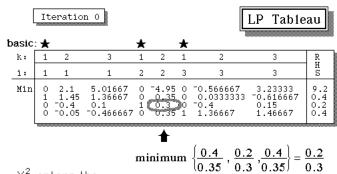


minimum $\left\{\frac{0.166}{0.5}, \frac{0.1666}{1.833}\right\} = \frac{0.1666}{1.8333}$

 X_3^2 enters the basis, replacing X_3^1

Iteration 2 Policy: (Cost= ~13.1515)

| State | Action | P{i} |
|----------|------------|-----------|
| 1 Town A | 1 Cruise | 0.121212 |
| 2 Town B | 2 Cabstand | 0.787879 |
| 3 Town C | 2 Cabstand | 0.0909091 |



X₂² enters the basis, replacing X₂¹
^{®Dennis Bricker, U. of Iowa, 1998}

Iteration 1

Policy: (Cost= ~12.5)

| State | Action | P{i} |
|----------|------------|----------|
| 1 Town A | 1 Cruise | 0.166667 |
| 2 Town B | 2 Cabstand | 0.666667 |
| 3 Town C | 1 Cruise | 0.166667 |

@Dennis Bricker, U. of Iowa, 1998

| | Iteration 2 | | | | | | | | Γab1eau |
|-----|------------------|--|--|---|---|---|---|--|---------|
| | * | | | | * | | * | | |
| k: | 1 | 2 | 3 | 1 | 2 | 1 | 2 | 3 | |
| i: | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | rhs |
| Min | 0 1 0 0 | -2.8712 1.8030 -1.0303 0.2272 | 4.3863 1.4090 -0.0909 -0.3181 | | 0 | | 0 | 10.7576 -1.1439 1.4393 0.7045 | 0.1212 |

Note that for every state, there is a variable in the basis for only one action!

@Dennis Bricker, U. of Iowa, 1998

LP Tableau

| | * * * * | | | | | LP Ta | ıb1eau | | |
|-----|---|---|---|------------------------------------|---|---------------------------------------|---------|---------------------------------------|--|
| k: | 1 | 2 | 3 | 1 | 2 | 1 | 2 | 3 | |
| i: | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | rhs |
| Min | 1.59244 0.55462 0.57142 -0.12605 | 0 | 6.63025 0.78151 0.71428 -0.49579 | 10.5882 -0.4705 2 -0.5294 | 0 | 3.4747 -0.1512 0.5714 0.5798 | 0 0 0 1 | 8.9359 -0.6344 0.7857 0.8487 | 13.3445 0.06722 0.85714 0.07563 |

Reduced costs are all nonnegative... the optimality condition is satisfied!

Optimal Policy

Iteration 3

Policy: (Cost= -13.3445)

| State | Action | P{i} |
|----------|------------|-----------|
| 1 Town A | 2 Cabstand | 0.0672269 |
| 2 Town B | 2 Cabstand | 0.857143 |
| 3 Town C | 2 Cabstand | 0.0756303 |

The optimal policy found by the simplex LP algorithm is deterministic, not randomized, i.e., for each state, only one action is specified.



@Dennis Bricker, U. of Iowa, 1998

LP Algorithm for MDP with discounting

Determining a policy which minimizes the *present value* of all future costs over an infinitely long planning horizon.

Note: existence of a steady state distribution is *not* assumed!



@Dennis Bricker, U. of Iowa, 1998

Decision variables

 $\lambda_i^k(n) \text{ = Joint probability that} \\ \text{and} \\ \text{action } k \in K_i \text{ is selected}$

Note that this definition of the decision variables does not assume that the same policy is optimal for every stage!

The present value of future costs (i.e., the discounted future costs) will depend upon the initial state of the system.

Define

 α_j = probability that system in initially in state j

Note: If the initial state is known, then $\alpha = [0, 0, ..., 0, 1, 0, ..., 0]$

@Dennis Bricker, U. of Iowa, 1998

Define

$$\beta$$
 = discount factor = $\frac{1}{1+r}$ where r = rate of return per stage

Then the present value of a cost Y which is incurred 1 period hence is βY 2 periods hence is $\beta^2 Y$ 1 n periods hence is $\beta^n Y$

@Dennis Bricker, U. of Iowa, 1998

@Dennis Bricker, U. of Iowa, 1998

 $If \quad C_j^k = cost \ of \ action \ k \ in \ state \ j$

then

$$\sum_{j} \sum_{k \in K_{j}} C_{j}^{k} \lambda_{j}^{k}(n) \quad \text{= expected cost during} \\ \text{stage (period) } n$$

and

$$\sum_{n=0}^{\infty} \beta^n \sum_j \sum_{k \in K_j} C_j^k \lambda_j^k(n) = \text{present value of} \\ \text{all costs in periods} \\ \text{n=0, 1, 2,}$$

Our objective is therefore to minimize the discounted future expected costs:

$$\sum_{j} \sum_{k \in K_{j}} \left[\sum_{n=0}^{\infty} \, \beta^{n} \, \, C_{j}^{k} \, \lambda_{j}^{k} \left(n \right) \right]$$

Constraints

For each state j at stage n=0: $\sum_{k \in K_i} \lambda_j^k(0) = \alpha_j$

For each state j at stage n, n=1,2,...

$$\underbrace{\sum_{k \in K_i} \lambda_j^k(n)}_{} = \underbrace{\sum_{i} \sum_{k \in K_i} p_{ij}^k \lambda_i^k(n-1)}_{}$$

Probability that system is in state j at stage n

Probability that system makes transition from state i in stage n-1 to state j in stage n

Note that there is an infinite number of as infinitely many constraints, as well

©Dennis Bricker, U. of Iowa, 1998

For each pair of state i and action k, consider the sequence of probabilities

$$\left\{\lambda_{j}^{k}(n)\right\}_{n=0}^{\infty}$$

Its z-transform is $F(z) = \sum_{i=1}^{\infty} z^{i} \lambda_{i}^{k}(n)$

Define a new set of decision variables

$$\mathbf{x}_{j}^{k} = \sum_{n=0}^{\infty} \beta^{n} \lambda_{j}^{k}(n)$$

i.e., the z-transform of $\{\lambda_i^k(n)\}$ evaluated at B

@Dennis Bricker, U. of Iowa, 1998

Constraints

In order to reduce the set of constraints to a finite number

(with finitely many variables), perform the following operations:

• For each pair j & n, multiply the corresponding constraint by Bn

$$\begin{cases} \beta^{o} \sum_{k \in K_{j}} \lambda_{j}^{k}(0) = \beta^{o} \alpha_{j} & \text{for each state } j \\ \\ \sum_{k \in K_{j}} \beta^{n} \lambda_{j}^{k}(n) = \beta \sum_{i} \sum_{k \in K_{i}} p_{ij}^{k} \beta^{n-1} \lambda_{i}^{k}(n-1) & \text{for each state } j \\ & & \& n \ge 1 \end{cases} \Rightarrow \sum_{n=0}^{\infty} \sum_{k \in K_{j}} \beta^{n} \lambda_{j}^{k}(n) = \alpha_{j} + \beta \sum_{n=1}^{\infty} \sum_{i} \sum_{k \in K_{i}} p_{ij}^{k} \beta^{n-1} \lambda_{i}^{k}(n-1)$$

@Dennis Bricker, U. of Iowa, 1998

· Rearrange the order of summation in this new

$$\sum_{n=0}^{\infty} \sum_{k \in K_j} \beta^n \lambda_j^k(n) = \alpha_j + \beta \sum_{n=1}^{\infty} \sum_i \sum_{k \in K_i} p_{ij}^k \beta^{n-1} \lambda_i^k(n-1)$$

$$\implies \sum_{k \in K_i} \sum_{n=0}^{\infty} \beta^n \lambda_j^k(n) = \alpha_j + \beta \sum_i \sum_{k \in K_i} \sum_{n=1}^{\infty} p_{ij}^k \beta^{n-1} \lambda_i^k(n-1)$$

$$\implies \frac{\sum_{k \in K_{j}} x_{j}^{k} = \alpha_{j} + \beta \sum_{i} \sum_{k \in K_{i}} x_{i}^{k}}{\sum_{k \in K_{i}} x_{i}^{k}} \quad \text{for all } j$$

since
$$\sum_{n=1}^{\infty} p_{ij}^k \beta^{n-1} \ \lambda_i^k (n-1) = \ \sum_{n=0}^{\infty} \beta^n \lambda_i^k (n)$$

In order to reduce the size of the LP to finite proportions, we will utilize the z - transform.

The z-transform of the sequence $\{a_n\}_{n=0}^{\infty}$ is the function

$$F(z) = \sum_{n=0}^{\infty} z^n a_n$$

[See Queueing Systems, Vol. 1, Appendix 1 by L. Kleinrock]

Note that, given F, we can $a_n = \frac{1}{n!} \frac{d^n F(0)}{dz^n}$

@Dennis Bricker, U. of Iowa, 1998

We are then able to rewrite our objective function

$$\sum_{j} \sum_{k \in K_{j}} \left[\sum_{n=0}^{\infty} \, \beta^{n} \, C_{j}^{k} \, \lambda_{j}^{k}(n) \right]$$

with a finite number of terms:

$$\sum_j \sum_{k \in K_j} C_j^k x_j^k$$

where

$$\mathbf{x}_{j}^{k} = \sum_{n=0}^{\infty} \, \beta^{n} \, \lambda_{j}^{k}(n)$$

@Dennis Bricker, LL of Iowa, 1998

For each state j, sum the equations over n:

$$\begin{cases} \beta^o \sum_{k \in K_j} \lambda^k_j(0) = \beta^o \alpha_j & \text{for each state } j \\ \sum_{k \in K_j} \beta^n \lambda^k_j(n) = \beta \sum_i \sum_{k \in K_i} p^k_{ij} \beta^{n-1} \ \lambda^k_i(n-1) & \text{for each state } j \\ & \& \ n \geq 1 \end{cases}$$

$$\implies \sum_{n=0}^{\infty} \sum_{k \in K_i} \beta^n \lambda_j^k(n) = \alpha_j + \beta \sum_{n=1}^{\infty} \sum_i \sum_{k \in K_i} p_{ij}^k \beta^{n-1} \lambda_i^k(n-1)$$

@Dennis Bricker, U. of Iowa, 1998

LP Model

$$\label{eq:minimize} \text{Minimize } \sum_{j} \sum_{k \; \in \; K_{j}} C_{j}^{k} x_{j}^{k}$$

subject to

$$\sum_{k \in K_j} x_j^k = \alpha_j + \beta \sum_i \sum_{k \in K_i} p_{ij}^k x_i^k \quad \text{for all } j$$

$$x_i^k \ge 0$$

 $\mbox{Note that} \quad \begin{cases} \bullet \ \ sum \ of \ x \ is \ not \ specified \ to \ be \ 1 \\ \bullet \ \ no \ redundant \ constraint \ was \end{cases}$

eliminated from state equations

MDP LP Algorithm 1/28/98 page 5

Using the "Kronecker delta", i.e.,

$$\delta_{ij} \equiv \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

this LP model may be rewritten:

Minimize
$$\sum_{j} \sum_{k \in K_{j}} C_{j}^{k} x_{j}^{k}$$
 subject to
$$\sum_{i} \sum_{k \in K_{i}} \left(\delta_{ij} - \beta p_{ij}^{k} x_{i}^{k} \right) = \alpha_{j} \quad \text{for all } j$$

$$x_{j}^{k} \ge 0$$

If x^* is the optimal basic solution, then

$$x_i^{*k} > 0$$
 (i.e., basic)

implies that

the optimal policy is to select action k when in state j for every stage n=0,1,2,...

i.e., the optimal policy is stationary, same for every time period!



⊚Dennis Bricker, U. of Iowa, 1998

⊚Dennis Bricker, U. of Iowa, 1998