Pricing Related Goods

Jonathan Stokes

September 28, 2021

Be forewarned that while I think I have some pretty numerical results, the analytical side is still a bit rough. For instance I would love to be able to prove the concavity of the profit equations, perhaps another day. Also this entire set of notes is a rough draft, for what I'm unsure. A Java implementation of the profit optimization algorithm can be found here `https://github.com/sophist0/opt_profit`

Two Good Model

Assume we have two goods labeled G_1 and G_2 where G_2 is a sequel to G_1. For instance G_1 is Home Alone and G_2 is Home Alone 2. These have a potential audience A. The fraction of the audience that wants to good G_1 first is $f_{0,1}$. The faction that wants to see movie M_2 first is $f_{0,2}$. The fraction of viewers that see G_1 first and then want to see G_2 is $f_{1,2}$ and the fraction of viewers that see G_2 first and then want see G_1 is $f_{2,1}$. These fractions are subject to the following restriction $0 \leq f_{i,j} \leq 1$ for $i \in [0, 1, 2]$, $j \in [1, 2]$, and $i \neq j$. Additionally no sum of fractions leaving an audience node may be greater than 1 which in this case implies that

$$0 \leq f_{0,1} + f_{0,2} \leq 1 \tag{1}$$

or generally

$$0 \leq \sum_j f_{i,j} \leq 1 \text{ for all } i \tag{2}$$

These fractions $f_{i,j}$ are often empirically available to online marketplaces such as YouTube Movies.

The possible orders in which the audience can purchase G_1 and G_2 can be represented as a graph.

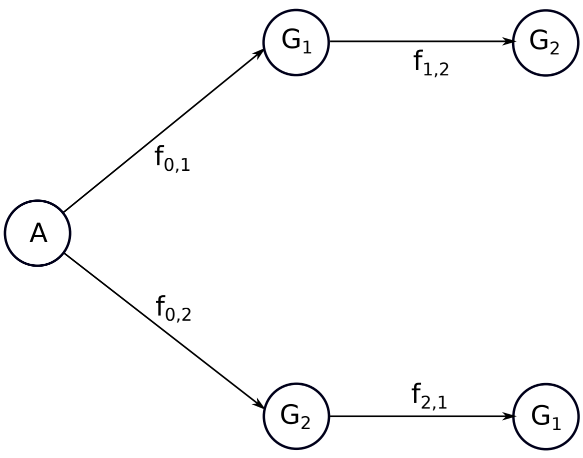

Figure 1: Two Good Purchase Order Model

To this model we can add the price p_i of each good. But before doing so lets normalize the min and max pricing of the goods such that at $p_i = 0$ everyone who wants good G_i purchases it and at $p_i = 1$ no one who wants good G_i purchases it. As with the fractions $f_{i,j}$ the actual min and max prices of a good can be estimated empirically. At this point I need to make and important clarification. I assume the min and max prices of all goods are the same. This makes sense in the case of two movies, but is harder to justify if the goods in question were a pound of coffee and a pound of sugar.

Define p_i as having domain $p_j \in [0, 1]$. Finally I assume that every member of the audience is subject to the same linear price sensitivity function $s_j = (1 - p_j)$ so if $f_{i,j} = 1$ for all i at $p_j = 1/2$ half the members of the audience who want to purchase good G_j do so. I have no reason to think that this price sensitivity function reflects actual consumer behavior, I chose it for its simplicity. The resulting two good pricing model is given below,

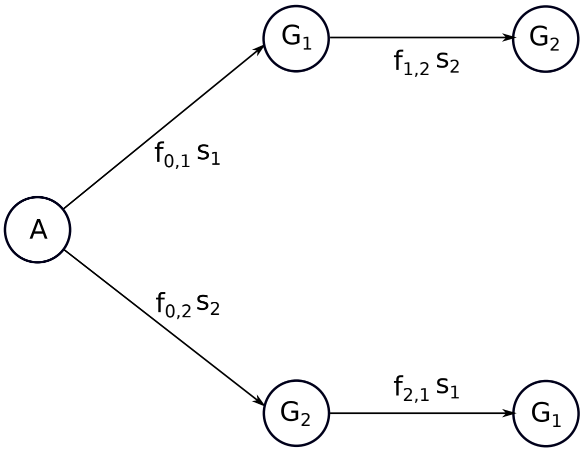

Figure 2: Two Good Pricing Model

If we do not assume that all goods have the same min and max price I think the sensitivity function could be formulated as

$$s_i = 1 - \frac{p_i - p_i^-}{p_i^+ - p_i^-} \tag{3}$$

where p_i^+ is the max price of good i and p_i^- the min price. But I have not worked out this case.

The model in the figure above can be collapsed to the following graph,

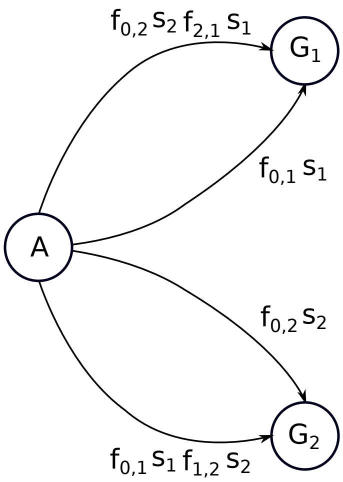

Figure 3: Collapsed Two Good Pricing Model

Two Good Profit Equation

If the audience size is N its clear from the model above that the profit earned by both goods can be computed as

$$\begin{aligned}
profit &:= f(p_1, p_2) && (4)\\
&= N[(f_{0,1}s_1 + f_{0,2}s_2 f_{2,1}s_1)p_1 + (f_{0,2}s_2 + f_{0,1}s_1 f_{1,2}s_2)p_2] && (5)\\
&\propto (f_{0,1}s_1 + f_{0,2}s_2 f_{2,1}s_1)p_1 + (f_{0,2}s_2 + f_{0,1}s_1 f_{1,2}s_2)p_2 && (6)\\
&= [(f_{0,1}(1-p_1) + f_{0,2}f_{2,1}(1-p_1)(1-p_2)]p_1 + [(f_{0,2}(1-p_2) + f_{0,1}f_{1,3}(1-p_1)(1-p_2))]p_2 && (7)\\
&= (p_1 - p_1^2)(f_{0,1} + f_{0,2}f_{2,1}(1-p_2)) + (p_2 - p_2^2)(f_{0,2} + f_{0,1}f_{1,2}(1-p_1)) && (8)
\end{aligned}$$

Lets simplify the notation using the following mapping.

$$p_1 \to x,\ p_2 \to y,\ f_{0,1} \to a,\ f_{0,2} \to b,\ f_{0,1}f_{1,2} \to c,\ f_{0,2}f_{2,1} \to d \tag{9}$$

Under this mapping the equation simplifies to

$$g(x,y) = \frac{f(x,y)}{N} = (x - x^2)(a + d(1-y)) + (y - y^2)(b + c(1-x)) \tag{10}$$

Taking the first and second partial derivates of $g(x,y)$ gives

$$\begin{aligned}
\frac{\partial}{\partial x} g(x,y) &= (1-2x)(a + d(1-y)) - c(y - y^2) && (11)\\
\frac{\partial}{\partial y} g(x,y) &= (1-2y)(b + c(1-x)) - d(x - x^2) && (12)\\
\frac{\partial^2}{\partial x^2} g(x,y) &= -2(a + d(1-y)) && (13)\\
\frac{\partial^2}{\partial y^2} g(x,y) &= -2(a + d(1-x)) && (14)\\
\frac{\partial^2}{\partial x \partial y} g(x,y) &= \frac{\partial^2}{\partial y \partial x} g(x,y) = 2d\left(x - \frac{1}{2}\right) - 2c\left(y - \frac{1}{2}\right) && (15)\\
& && (16)
\end{aligned}$$

Therefore the Hessian matrix of $g(x,y)$ is

$$Hg(x,y) = \begin{bmatrix} \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial x \partial y} \\ \frac{\partial^2}{\partial y \partial x} & \frac{\partial^2}{\partial y^2} \end{bmatrix} = \begin{bmatrix} -2[a + d(1-y)] & 2[d(x - 1/2) - c(y - 1/2)] \\ 2[d(x - 1/2) - c(y - 1/2)] & -2[b + c(1-x)] \end{bmatrix} \tag{17}$$

Results for the Two Good Profit Equation

The squared terms in the first order partial derivatives of $g(x,y)$ make finding a closed form solution for the critical points of $g(x,y)$ difficult in terms of arbitrary parameterizations of $f_{0,1}$, $f_{0,2}$, $f_{1,2}$, and $f_{2,1}$. Therefore before solving for these points numerically I set the values of these parameters as follows,

$$f_{0,1} = \frac{9}{10},\ f_{0,2} = \frac{1}{10},\ f_{1,2} = \frac{3}{4},\ f_{2,1} = \frac{1}{4} \tag{18}$$

Practically these fractions could be found empirically by setting $p_1 = 0$ and $p_2 = 0$. Using these values under the mapping defined above gives

$$a = \frac{9}{10},\ b = \frac{1}{10},\ c = \frac{27}{40},\ d = \frac{1}{40} \tag{19}$$

Plugging these values into the partial derivatives of $g(x,y)$ gives

$$\begin{aligned}
\frac{\partial}{\partial x} g(x,y) &= (1-2x)\left(\frac{9}{10} + \frac{1}{40}(1-y)\right) - \frac{27}{40}(y - y^2) = 0 && (20)\\
\frac{\partial}{\partial y} g(x,y) &= (1-2y)\left(\frac{1}{10} + \frac{27}{40}(1-x)\right) - \frac{1}{40}(x - x^2) = 0 && (21)
\end{aligned}$$

Solving this system of equations with Sympy's nsolve and initial points $x, y = 1/2$, I found the critical point for $g(x, y)$ is $x^* = 0.408$ and $y^* = 0.494$ rounding to the thousandths place.

Plugging the values into the Hessian matrix we can determine if this solution is a saddle point or a local maxima using the partial derivative test.

$$Hg(x^*, y^*) = \begin{bmatrix} -1.8253 & 0.0035 \\ 0.0035 & -0.9992 \end{bmatrix} \tag{22}$$

The computing eigenvalues of $Hg(x^*, y^*)$ with numpy gives $\lambda_1 = -1.8253$ and $\lambda_2 = -0.9992$. Since all the eigenvalues of the Hessian are negative, the Hessian is a negative definite matrix. Therefore $g(x, y)$ achieves a local maxima at point $(0.408, 0.494)$.

Is this solution better than if we fixed $p_1 = p_2$ in the Two Good Pricing Model above? If we set $x = y$, $g(x, y)$ reduces to

$$g(x) = x^3(c + d) - x^2(a + b + 2(c + d)) + x(a + b + c + d) \tag{23}$$

and the first and second derivatives of $g(x)$ are

$$\frac{d}{dx}g(x) = 3x^2(c + d) - 2x(a + b + 2(c + d)) + (a + b + c + d) \tag{24}$$

$$\frac{d^2}{dx^2}g(x) = 6x(c + d) - 2(a + b + 2(c + d)) \tag{25}$$

Setting the first derivative of $g(x)$ to zero and solving for the two critical points gives $x^* \in 0.438, 1.848$. Since $x := p_1$ by definition $0 \leq x < 1$ and it follows that $x^* = 0.438$.

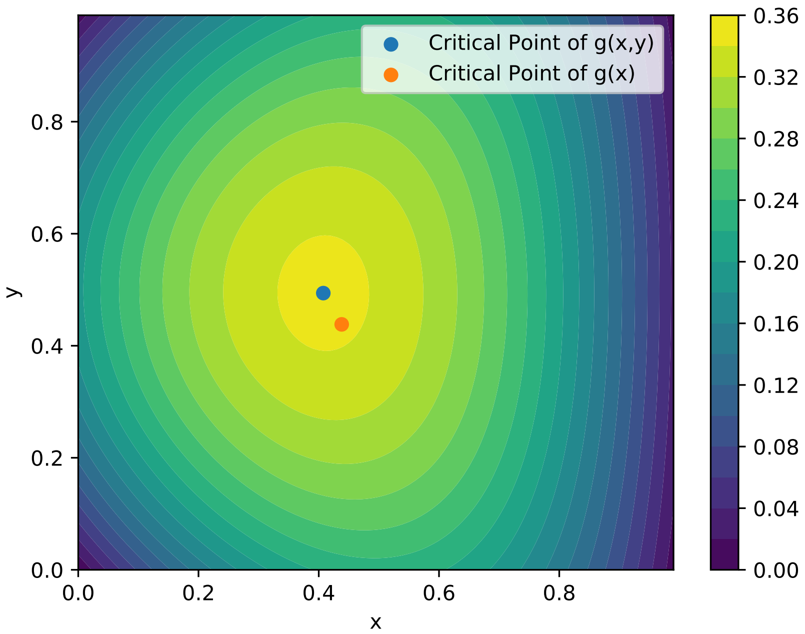

Figure 4: Contour Plot of $g(x, y)$

Now looking at the second derivative of $g(x)$ at x^* we find that

$$\frac{d^2}{dx^2}g(x^*) = -0.560 < 0 \tag{26}$$

Therefore $g(x^*)$ is a local maxima by the second derivative test and the only maxima in the domain of x.

What is the difference in profit achieved using $g(x, y)$ vs $g(x)$ to optimize pricing? Well

$$g(x^*) = 0.34299 \tag{27}$$
$$g(x^*, y^*) = 0.34532 \tag{28}$$
$$\frac{g(x^*, y^*)}{g(x^*)} = 1.00678 \tag{29}$$

or about a 0.68% increase as a result of price optimizing via $g(x, y)$. But recall this is the normalized profit per potential audience member. If the maximum price anyone is willing to pay for G_1 and G_2 is $p^+ = \$20$ while the minimum is $p^- = \$0$, this sets $p_1 = \$8.16$, $p_2 = \$9.88$ under $g(x, y)$ and $p_1 = p_2 = \$8.76$ under $g(x)$. Now if the audience size is 10^8 the difference in profit using $g(x, y)$ to set prices vs $g(x)$ is about $\$13,560,000$ which is not trivial.

n Good Model

From the small example given in Fig. 2 its clear that to compute the profit over a set of goods purchased in sequence requires looking at every permutation of that sequence. The figure below gives a tree representing these permutations for a sequence of n goods.

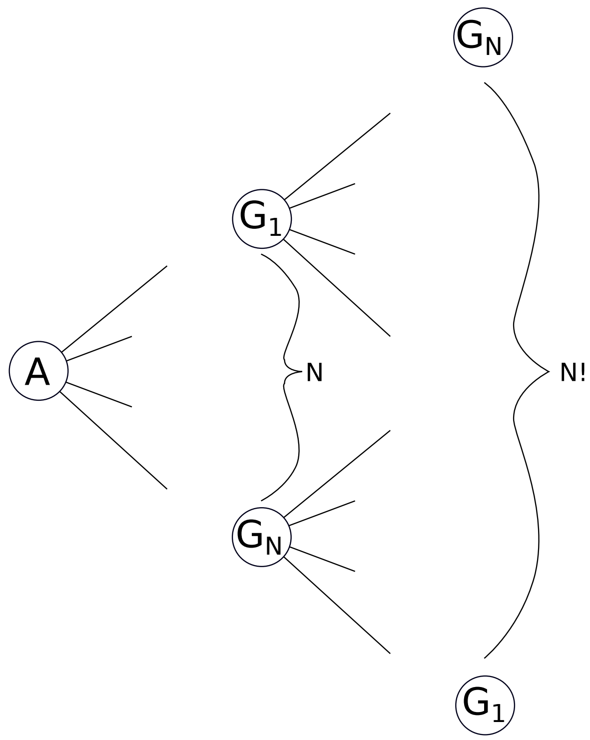

Figure 5: n Good Model

This tree has the following statistics.

- The n goods have $n!$ permutations and therefore this tree has $n!$ leaves.
- The number of edges in the tree can be computed as

$$edges = n + n(n-1) + n(n-1)(n-2) + \dots \tag{30}$$
$$= \sum_{i=0}^{n-1} \left(\prod_{j=0}^{i} (n-j) \right) \tag{31}$$

- There are $2\binom{n}{2} + n$ unique parameters representing fractions $f_{i,j}$. Here the final term captures the fraction of the audience that wants to purchase each good first i.e. fractions $f_{0,j}$.

If the graph of the n Good Model is collapsed as we collapsed the graph of the Two Good Model the result is n graphs of the form below each with $n!$ weighted edges from the audience to a good G_l for $l \in [1, \ldots, n]$.

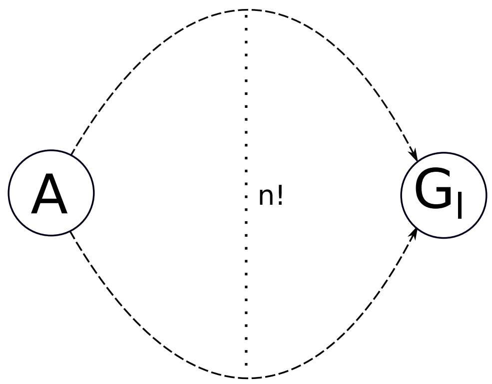

Figure 6: Collapsed n Goods Model

Each of these $n!$ edges is weighted by the product of between 1 and n terms of the form $f_{x,y}(1-p_j)$ depending on where in the permutation G_l is and each of these weighted edges is multiplied by p_l the price of good G_l giving equation for calculating the profit of good G_l along the k_{th} edge the form,

$$g_{l,k}(p_j,\ldots,p_k) = f_{x,y}(1-p_j)f_{q,r}(1-p_l)\cdots f_{w,z}(1-p_k)p_l \tag{32}$$

Therefore the equation for the total profit takes the form

$$profit = f(p_1,\ldots,p_n) = \sum_{l=1}^{n} g_l(p_1,\ldots,p_n) = \sum_{l=1}^{n}\sum_{k=1}^{n!} g_{l,k}(p_j,\ldots,p_k) \tag{33}$$

Taking the second partial derivatives of $g_{l,k}(p_1,\ldots,p_n)$ gives

$$\frac{\partial^2}{\partial p_l^2} g_{l,k}(p_j,\ldots,p_k) = -2f_{x,y}(1-p_j)\cdots f_{u,v}(1-p_{l-1})f_{q,r}f_{e,o}(1-p_{l+1})\cdots f_{w,z}(1-p_k) \tag{34}$$

$$\frac{\partial^2}{\partial p_i \partial p_j} g_l(p_j,\ldots,p_k) = 0 \text{ for } i \neq l \tag{35}$$

$$\frac{\partial^2}{\partial p_l \partial p_j} g_{l,k}(p_j,\ldots,p_k) = -f_{x,y}f_{q,r}(1-2p_l)\cdots f_{w,z}(1-p_k) \tag{36}$$

To simplify notation lets define this second partial derivatives of $g_l(p_j,\ldots,p_k)$ and $g_{l,k}(p_j,\ldots,p_k)$ as

$$g''_{l,k}(l,j) := \frac{\partial^2}{\partial p_l \partial p_j} g_{l,k}(p_j,\ldots,p_k) \tag{37}$$

$$g''_l(l,j) := \frac{\partial^2}{\partial p_l \partial p_j} g_l(p_j,\ldots,p_k) \tag{38}$$

where $l, j \in [1,\ldots,n]$. It follows from the sum rule for derivatives that

$$g''_l(l,j) = \sum_{k=1}^{n!} g''_{l,k}(l,j) \tag{39}$$

Notice that the functions $g''_l(l,j)$ are the l,j entries of the Hessian matrix for the profit function $g(p_1,\ldots,p_n)$ such that

$$Hg(p_1,\ldots,p_n) = \begin{bmatrix} g''_1(1,1) & g''_1(1,2) & \ldots & g''_1(1,n) \\ g''_2(2,1) & g''_2(2,2) & \ldots & g''_2(2,n) \\ \vdots & \vdots & \ldots & \vdots \\ g''_n(n,1) & g''_n(n,2) & \ldots & g''_n(n,n) \end{bmatrix} \tag{40}$$

Which in theory could be computed for given a set of prices and audience fraction parameters for an arbitrary n. This Hessian matrix could then be used to determine if a critical point on the surface of the normalized profit function $g(p_1, \ldots, p_n)$ is a local maxima.

The Profit Algorithm

Assume we can construct a tree T_n as shown in the n-Good Model with node set $\mathcal{T}_n$ and root node v_0. Let $\mathbf{k}_i$ be a vector encoding the location of each node $v_{\mathbf{k}_i}$ such that

$$\mathbf{k}_i = [1, 2, \ldots, i-1, i] \tag{41}$$

and the F be a hashmap of factors for each non-root node where $F_{\mathbf{k}_i} = f_{\mathbf{k}_i}$ is the factor corresponding to node $v_{\mathbf{k}_i}$. Finally let $\mathbf{p}_i = p_i$ for $i \in [1, \ldots, n]$ where p_i is the price of good i.

---

profit $= 0$
for $v_{\mathbf{k}_i} \in \mathcal{T}_n \backslash \{v_0\}$ **do**
  $c = 1$
  for $j \in \mathbf{k}_i$ **do**
    $c = c(1 - \mathbf{p}_j)F_{\mathbf{k}_j}$
  end for
  $profit = c\mathbf{p}_i$
end for
return $profit$

Using stochastic gradient ascent (SGA) to optimize the Two Good Model with the parameters f_1, f_2, f_3, f_4 given and computing the gradient of profit using the algorithm above as a function of the prices $\mathbf{p}$. The SGA algorithm initiates all prices to 1/2, the learning rate is $\nu = 0.01$, and it iterates 1000 times, resulting in $profit = 0.34302$ for $x = 0.434$, $y = 0.446$ which is the same to the hundredths place as the local maxima of $g(x, y)$ found above. Perhaps this algorithm would perform a bit better with an adaptive learning rate.

Results for the n Good Profit Equation

The SGA optimization algorithm can be used to optimize larger sets of goods and is limited only by the difficulty of computing the profit over the $n!$ permutations of the goods. This difficulty can be mitigated if its assumed most these permutations do not occur. I think the number of permutations that are likely to occur can be quantified using typical sets. But I have not done any pruning of the permutation space here. If we assume the fractions of the audience who would like to purchase the n goods are defined as follows.

$$f_{0,i} = 2^{-i} \tag{42}$$

$$f_{i,i+1} = \frac{9}{10} \text{ for } i > 0 \tag{43}$$

$$f_{i,j} = \frac{1}{100} \text{ for } i > 0 \text{ and } j \neq i+1 \tag{44}$$

We can look at how the optimal prices vary with n as computed by the SGA in the table below where the gradient is estimated using $\delta = 0.001$, 1000 iterations, and $\nu = 0.01$. This model assumes that almost everyone purchases goods in order and the optimized pricing suggests its best to make most of the profit on the first good purchased and make less profit on the remaining goods which might be purchased. The

n	p_1^*	p_2^*	p_3^*	p_4^*	p_5^*	p_6^*	$g(p_1, \dots p_n)$	p^*	$g(p^*)$	$g(p_1^*, \dots p_n^*)/g(p^*)$
1	0.500						0.1250	0.500	0.1250	1.000
2	0.617	0.183					0.2700	0.451	0.2472	1.092
3	0.605	0.274	0.220				0.3796	0.413	0.3437	1.104
4	0.593	0.293	0.239	0.313			0.4572	0.368	0.4162	1.100
5	0.569	0.301	0.254	0.294	0.375		0.5109	0.341	0.4700	1.087
6	0.556	0.319	0.260	0.298	0.356	0.427	0.5456	0.334	0.5100	1.070

Table 1

apparent reduction in the advantage of letting prices vary independently as n increases actually appears to be a function of the number of SGA iterations being fixed and difficulty of the optimization problem increasing with n.

Pruning Permutations

Consider the following tree which generates all permutations of $\{1, 2, 3\}$ prepended with a 0. Lets call this tree the source $\mathbf{X}$. Let X_i be the i_{th} permutation generated by $\mathbf{X}$ and Y_j the j_{th} value in a permutation sometimes specified as $\mathsf{Y}_{i,j}$ if it is required to indicate that its the j_{th} value of the i_{th} permutation. The fractions in this tree were chosen independently at random according to the following distributions.

$$Pr(\mathsf{f}_{0,j} = \frac{1}{3}) = 1, \qquad Pr(\mathsf{f}_{0,j} = \frac{1}{6}) = 0; \qquad Pr(\mathsf{f}_{i,j} = \frac{1}{3}) = \frac{1}{3}, \qquad Pr(\mathsf{f}_{i,j} = \frac{1}{6}) = \frac{2}{3} \tag{45}$$

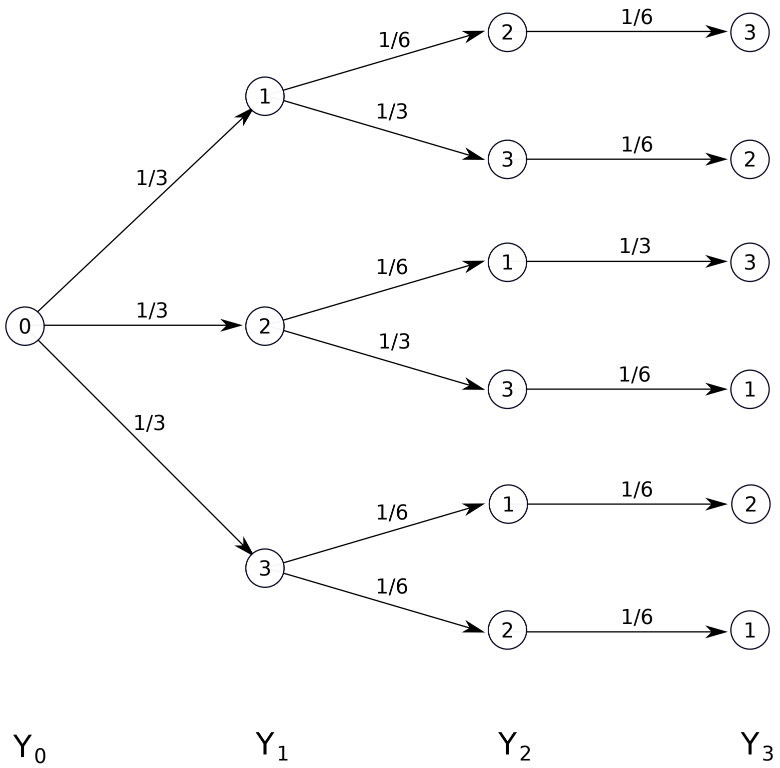

Figure 7: Source $\mathbf{X}$

The probabilities of Y_j taking on a value in a message X_i from $\mathbf{X}$ is given in the table below.
Examining the table above its clear that the distribution of Y_j is not equal to the distribution of Y_{j-1}, therefore Y_j is not a stationary stochastic process. Additionally examining the source notice that $P(\mathsf{Y}_3 = 3|\mathsf{Y}_2 = 2)$ may equal $\frac{1}{6}$ or 0, therefore Y_j is not a Markov process.

Y_0	Y_1	Y_2	Y_3
$P(\mathsf{Y}_0 = 0) = 1$	$P(\mathsf{Y}_1 = 0) = 0$	$P(\mathsf{Y}_2 = 0) = 0$	$P(\mathsf{Y}_3 = 0) = 0$
$P(\mathsf{Y}_0 = 1) = 0$	$P(\mathsf{Y}_1 = 1) = \frac{1}{3}$	$P(\mathsf{Y}_2 = 1) = \frac{2}{9}$	$P(\mathsf{Y}_3 = 1) = \frac{1}{36}$
$P(\mathsf{Y}_0 = 2) = 0$	$P(\mathsf{Y}_1 = 2) = \frac{1}{3}$	$P(\mathsf{Y}_2 = 2) = \frac{1}{9}$	$P(\mathsf{Y}_3 = 2) = \frac{1}{36}$
$P(\mathsf{Y}_0 = 3) = 0$	$P(\mathsf{Y}_1 = 3) = \frac{1}{3}$	$P(\mathsf{Y}_2 = 3) = \frac{2}{9}$	$P(\mathsf{Y}_3 = 3) = \frac{1}{27}$
$P(\mathsf{Y}_0 = \emptyset) = 0$	$P(\mathsf{Y}_1 = \emptyset) = 0$	$P(\mathsf{Y}_2 = \emptyset) = \frac{4}{9}$	$P(\mathsf{Y}_3 = \emptyset) = \frac{49}{54}$

Table 2

However if we consider the source $\mathbf{X}$ producing messages X_i the situation is very different. Each message X_i is independent and identically distributed according to distributions Y_j for $j \in [0, \dots, n]$. Denoting the distribution of X_i explicitly as F_{X_i} its also clear that the source $\mathbf{X}$ is strictly stationary since,

$$F_{\mathsf{X}_i} = F_{\mathsf{X}_{i+\tau}} \qquad \text{for all} \qquad i, \tau \in \mathbb{N} \tag{46}$$

Before continuing its necessary to define the convergence of a sequence of random variables $\mathsf{X}_1, \mathsf{X}_2, \dots$ to a random variable X in probability if for every $\epsilon > 0$, $Pr\{|\mathsf{X}_n - \mathsf{X}| > \epsilon\} \to 0$. Given the notion of convergence in probability the Asymptotic Equipartition Property (AEP) Theorem (Cover Thm. 3.1.1) can be stated as

$$-\frac{1}{n} \log p(\mathsf{X}_1, \mathsf{X}_2, \dots, \mathsf{X}_n) \to^{prob.} H(\mathsf{X}) \tag{47}$$

where $H(\mathsf{X})$ is the entropy of X.

Cover also defines a *typical set* $A_\epsilon^{(n)}$ for a sequence $(\mathsf{X}_1, \mathsf{X}_2, \dots, \mathsf{X}_n) \in \mathcal{X}^n$ with the property

$$2^{-n(H(\mathsf{X})+\epsilon)} \leq Pr(\mathsf{X}_1, \mathsf{X}_2, \dots, \mathsf{X}_n) \leq 2^{-n(H(\mathsf{X})-\epsilon)} \tag{48}$$

In Thm 3.1.2 Cover states that the typical set $A_\epsilon^{(n)}$ has among others the following properties

$$Pr\{A_\epsilon^{(n)}\} > 1 - \epsilon \qquad \text{for } n \text{ sufficiently large.} \tag{49}$$

$$|A_\epsilon^{(n)}| \leq 2^{n(H(\mathsf{X})+\epsilon)} \tag{50}$$

$$|A_\epsilon^{(n)}| \geq (1-\epsilon)2^{n(H(\mathsf{X})-\epsilon)} \qquad \text{for } n \text{ sufficiently large.} \tag{51}$$

Given our definition of the source $\mathbf{X}$ and the messages it produces satisfying the AEP property it follows that if $\mathbf{X}$ produces sequences of length m the number of these typical sequences is less than $2^{m(H(\mathsf{X})+\epsilon)}$.

Now unfortunately this does not tell us the number of unique permutations X_i in the length m sequences that make up the typical set $A_\epsilon^{(m)}$. Clearly its $\leq m2^{m(H(\mathsf{X})+\epsilon)}$ but its not clear this is a tight enough bound to be useful. Especially given that there is no necessary relation between n the number of elements in the permutation X_i and m the length of permutation sequences. So perhaps we are at a dead end here and may need to take a different approach.

First notice that the random variables Y_j for the j_{th} element in a permutation follow a categorical distribution and thus if one has access to that distribution permutations can be sampled directly. This while interesting is not useful if finding the parameters of the categorical distribution requires generating every permutation? Probably not.

Say instead we set a threshold and say we want to compute the profit on any branch upto and including those with at least $\alpha = \frac{1}{54}$ probability. How does this effect the source $\mathbf{X}$

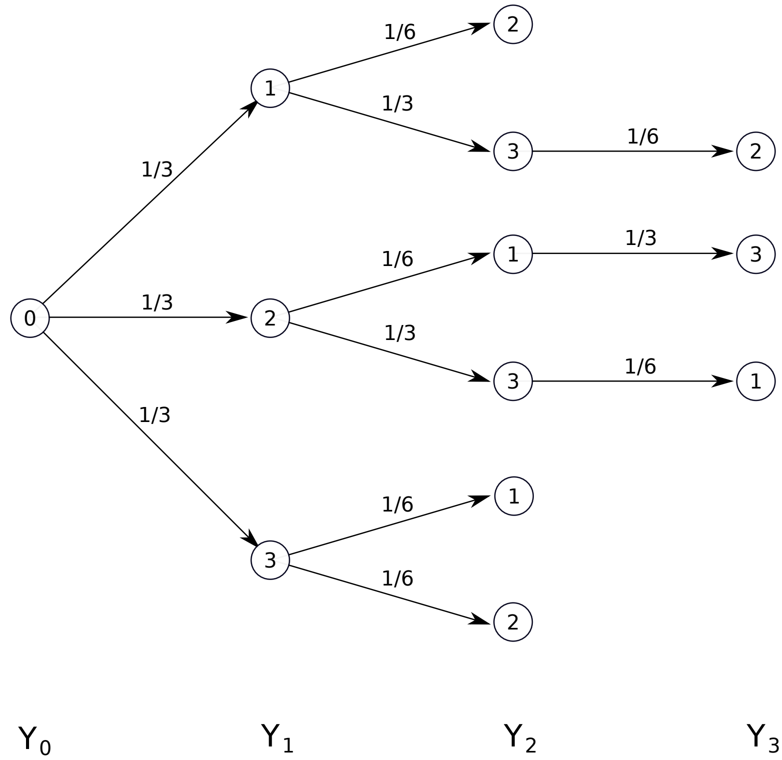

Figure 8: Truncated Source $\mathbf{X}$

Unfortunately picking such a large α results in little pruning of such a small tree. Nor does setting threshold α give us how confident we should be that we captured $\beta\%$ of permutations in way that two standard deviations is guaranteed by the Chebyshev inequality to capture 75% of observations. However if we sum the probability at each root of the sub-trees pruned from the truncated source we have the probability any message in the original source cannot be reproduced on the truncated source. To get at this quantity lets call $\mathcal{V}$ the set of node in the source tree $\mathbf{X}$ and v a node in this set. Additionally lets call $\mathbf{m}$ a message from source $\mathbf{X}$ where $\mathbf{m}$ is a vector of nodes $v \in \mathcal{V}$ and denote the roots of the sub-trees in the source $\mathbf{X}$ but not in the truncated source $\hat{\mathbf{X}}$ as set $\mathcal{R}$. $\mathcal{R}$ is the set of pruned sub-tree roots.

$$\beta = \sum_{v \in \mathcal{V}} \mathbb{1}\left\{v \in \mathcal{R}\right\} Pr(v \in \mathbf{m}) \tag{52}$$

The probability that a message $\mathbf{m}$ produced by source $\mathbf{X}$ cannot the reproduced by $\hat{\mathbf{X}}$ is equal to β since β is the probability that a message $\mathbf{m}$ terminates at a node in $\mathbf{X}$ but not in $\hat{\mathbf{X}}$. In the tree above $\beta = \frac{1}{36}$ so the truncated source $\hat{\mathbf{X}}$ correctly reproduces $\frac{35}{36}$ of the messages sent by source $\mathbf{X}$. (Note this is different from the fraction of the profit captured by $\hat{\mathbf{X}}$ verse $\mathbf{X}$.)

The figures below show how prices and the number of nodes vary as a function of the threshold α. The parameters used in the optimization are the same as those used to generate the results in Table 1.

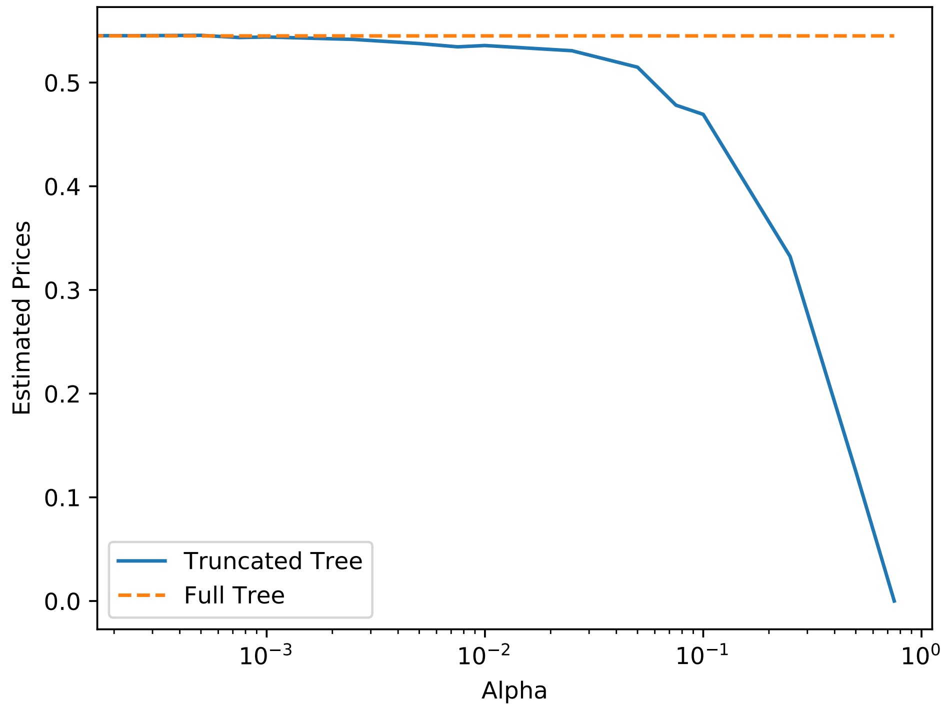

Figure 9: Optimized price $g(p_1^*, \ldots, p_n^*)$ as a function of α for $n = 6$.

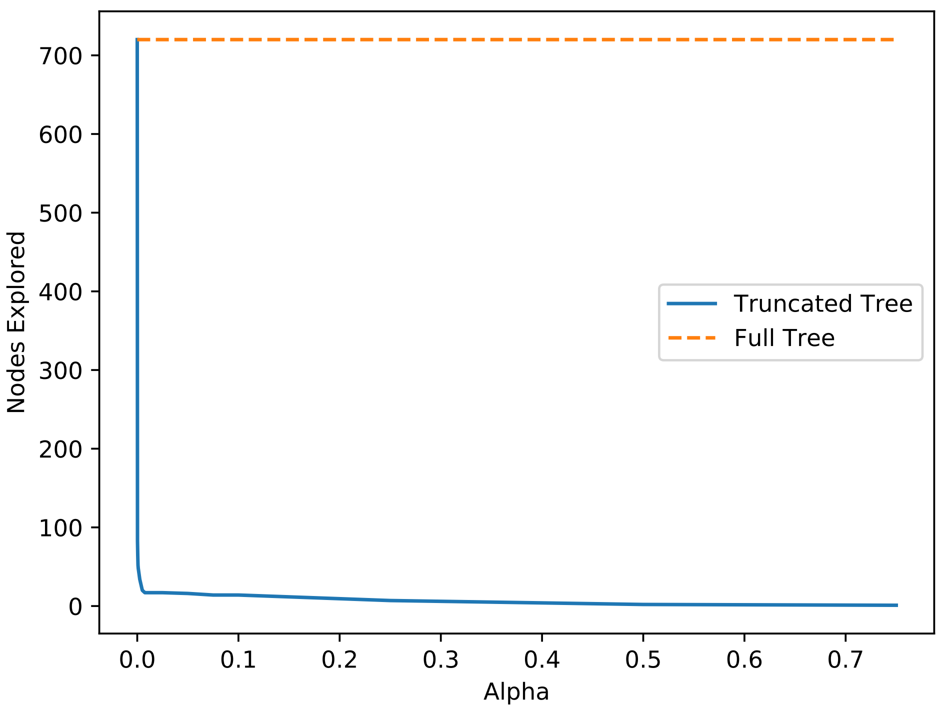

Figure 10: Nodes explored as a function of α for $n = 6$.

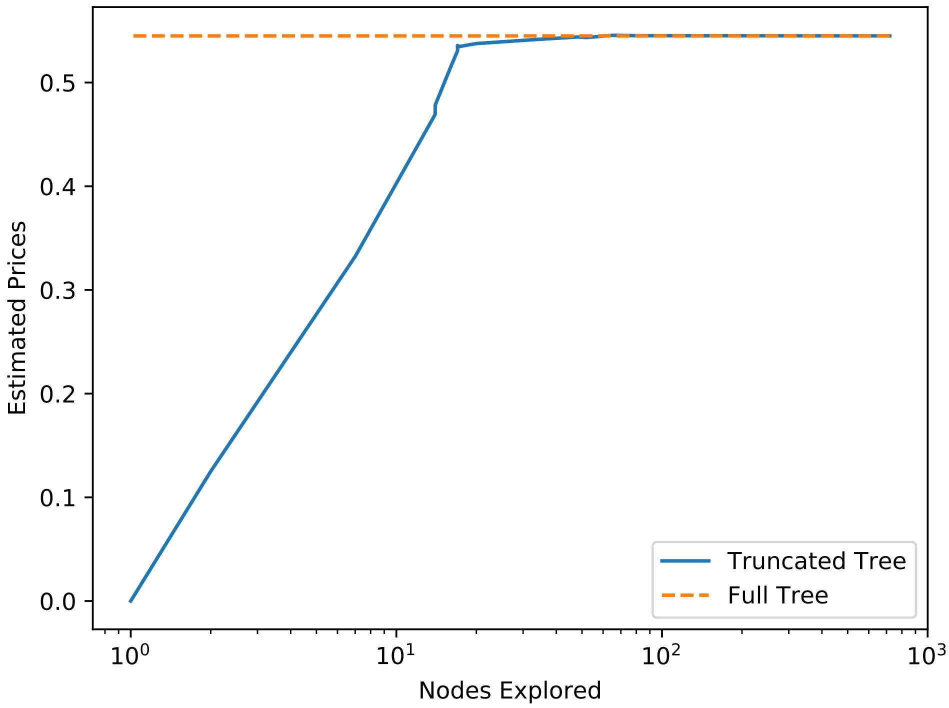

Figure 11: Optimized price $g(p_1^*, \ldots, p_n^*)$ as a function of nodes explored for $n = 6$

The actual results are given in Table 3 below. The takeaway is that one can reduce the number of nodes explored by at least a factor of 10 and still get close to the optimal price if all tree nodes are explored.

α	profit	nodes explored
0.0	0.545	1956
0.00025	0.545	107
0.0005	0.546	87
0.00075	0.543	72
0.001	0.544	68
0.0025	0.541	45
0.005	0.538	25
0.0075	0.534	21
0.01	0.536	21
0.025	0.531	20
0.05	0.515	18
0.075	0.478	15
0.1	0.469	14
0.25	0.332	7
0.5	0.125	1
0.75	0	0

Table 3

To construct a truncated source with a target β_t set α large and iteratively decrease α by ϵ until the difference between β of the truncated source and the target is acceptably small, $|\beta - \beta_t| < \delta$.

Next lets try to get a handle on the number of truncated permutations that exist for a given α without having to enumerate them. Let $\mathbf{f} = [f_v]$ be a vector containing the sum of the fraction weights leaving any node. So $f_r = 3(\frac{1}{3}) = 1$ the this sum for the root node. Define the following indicator function as

$$g(i, \alpha, \mathbf{f}) = \mathbb{1}\left\{\prod_{j=1}^{i} \mathbf{f}_j \geq \alpha\right\} \tag{53}$$

and

$$c(\alpha, \mathbf{f}) = \sum_{i=1}^{n} g(i, \alpha, \mathbf{f}) \tag{54}$$

Also lets use $\mathbf{f}^-$ to denote that the vectors elements are in decreasing order and $\mathbf{f}^+$ to denote that its elements are in increasing order.

An upper bound on the number of permutations ρ in the truncated tree can be given as

$$\rho \leq \prod_{k=0}^{c(\alpha,\mathbf{f}^-)-1} (n-k) \tag{55}$$

And a lower bound on the number of permutations in the truncated tree can be given as

$$\rho \geq \prod_{k=0}^{c(\alpha,\mathbf{f}^+)-1} (n-k) \tag{56}$$

The expected number or permutations takes additional work. Instead of considering the actual edge weights $f_{i,j}$ consider the random variables $\mathsf{f}_{i,j}$ representing the probabilities of those weights taking on specific values in our construction of the source. Since each permutation is prepended with a zero node representing the pool of people who may buy a good, $\mathbb{E}[\mathsf{f}_{0,j}]$ is likely different from $\mathbb{E}[\mathsf{f}_{i,j}]$ and in fact is different in our constructions. So we will define the two expectations separately as follows

$$\bar{f}_0 = \mathbb{E}[\mathsf{f}_{0,j}] = \sum_f f Pr(\mathsf{f}_{0,j} = f) \tag{57}$$

$$\bar{f}_{1+} = \mathbb{E}[\mathsf{f}_{i,j}] = \sum_f f Pr(\mathsf{f}_{i,j} = f) \tag{58}$$

and letting $\bar{\mathbf{f}} = [\bar{f}_0, \bar{f}_{1+}]$ the intermediate functions can be defined as

$$g(i, \alpha, \bar{\mathbf{f}}) = \mathbb{1}\left\{\bar{\mathbf{f}}_0(\bar{\mathbf{f}}_1)^{i-1} \geq \alpha\right\} \tag{59}$$

and

$$c(\alpha, \bar{\mathbf{f}}) = \sum_{i=1}^{n} g(i, \alpha, \bar{\mathbf{f}}) \tag{60}$$

Therefore it follows that

$$\mathbb{E}[\rho] = \prod_{k=0}^{c(\alpha,\bar{\mathbf{f}})-1} (n-k) \tag{61}$$

Now there is a caveat with this expectation. Equation (59) can be a reduced to a product of expectations because we chose the values of the random variables $\mathsf{f}_{0,j}$ and $\mathsf{f}_{i,j}$ independently. This is not generally the case for sources and if it is not the case equation (61) is only an approximation of the expectation and an unbounded one at that. So the best one can do is upper and lower bound the number of permutations in the truncated source.

To test these bounds lets construct a larger permutation tree than the one above. For $n = 6$ lets choose the fractions according to the following distribution

$$Pr(\mathsf{f}_{0,j} = \frac{1}{6}) = 1, \qquad Pr(\mathsf{f}_{0,j} = \frac{1}{12}) = 0; \qquad Pr(\mathsf{f}_{i,j} = \frac{1}{6}) = \frac{1}{6}, \qquad Pr(\mathsf{f}_{i,j} = \frac{1}{12}) = \frac{5}{6} \tag{62}$$

Clearly $\bar{\mathbf{f}} = [\frac{1}{6}, \frac{7}{72}]$. Sweeping α generates the following figure,

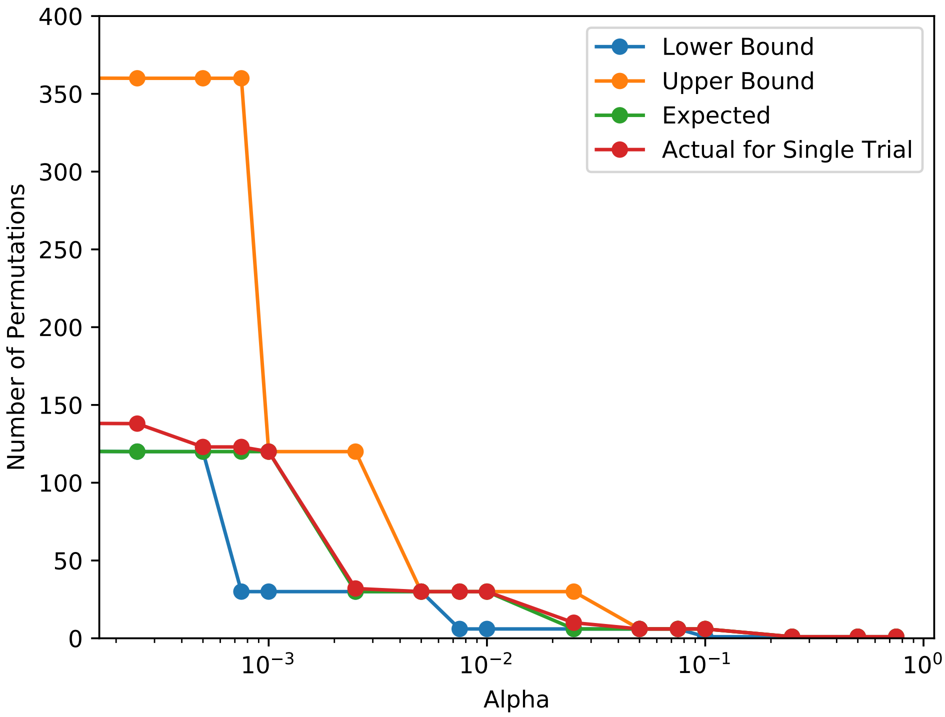

Figure 12

This figure could be improved by generating an expectation and standard deviation of the actual number of permutations via Monte Carlo simulation. Nonetheless even for the single trial shown the bounds and expectation seem correct.

References

Elements of Information Theory 2ed. Thomas Cover.

Final Thoughts

I have been doing a bit of reading to understand what the literature on this problem is. I found the following paper which solves the same problem under the constraint of having a discrete set of prices. Still they show their algorithm can scale to thousands of products and while in theory so could the above algorithm I'm not sure how effective the optimization would be in that case.

"Large-Scale Price Optimization via Network Flow". Shinji Ito and Ryohei Fujimaki.