Optimal Port Pricing and Expansion

Jack Devanney
Center for Tankship Excellence, USA, djw1@c4tx.org

Version: 2010-01-21
Unfinished draft for discussion and comment.

1 Introduction

In 1975, Devanney and Tan developed an algorithm for determining the societal income maximizing port pricing and expansion policies for a port faced with growing demand and large indivisible expansion alternatives. Their main goal was to demonstrate that not only was there no inconsistency between short-run allocative efficiency (marginal cost pricing) and long-run allocative efficiency (attracting the proper level of capital) but that the two are intimately and necessarily tied together.

Devanney and Tan (DandT) considered a single commodity port which every so often, call it a year, has a single investment opportunity: add or don't add a berth. All berths have identical marginal costs. If the decision is to add a berth, that berth comes on-line at the the next decision point. Finally, while the port faces a growing demand surface, the port is willing to treat demand as if it were constant between decision periods.

For such a port, Devanny and Tan developed a dynamic program based on the following two principles of economic efficiency:

1. In any short run, situation, the port must charge marginal social cost for its services.
2. The port should expand as soon as the capital (the resources) required for the expansion is more valuably employed in the port than elsewhere.

DandT show that a port following their policy will not need a subsidy despite charging marginal costs throughout no matter how large the berth fixed costs.[1] They point out that essentially what their algorithm does is simulate the competitive market dynamic in markets involving large fixed investments such as the tanker market.

The purpose of this paper is to present an algorithm, also a dynamic program, which improves on the Devanney and Tan algorithm in two respects.

1. It is slightly "more optimal".
2. Much more importantly, it generalizes to a wide range of more realistic port models, most of which cannot be handled by the DandT algorithm.

2 Consumers' and Producers' Surplus

Our algorithm will be based on the concept of consumers' and producers' surplus. In the short-run, a period short enough so that both the demand curve and and supply curve can be considered fixed, the consumers' and producers' surplus is the area below the short-run demand curve and above the short-run marginal cost curve integrated up to the short-run price. We will call this area, simply the *surplus*. It is easy to show that, in any short run situation, the surplus is maximized by setting price equal to marginal cost.

We extend this short-run concept to the dynamic situation. That is, we need an algorithm which will maximize the present value of the surplus net of fixed costs. The motivation here is Pareto-efficiency. If the port's pricing and expansion policy maximizes the sum of the consumers' and producers' surplus, it will be impossible to improve the consumers' surplus without decreasing the producers' surplus, and vice versa.[2] We can't increase the size of the pie because the pie is already as big as it can get.

[1] With one totally pathological exception discussed in Section 4.4.

[2] To be truly Pareto-optimal, all the prices exogenous to the port would need to reflect marginal social costs. This of course is never the case. Some have argued that therefore a port (or anything else) should not attempt to be Pareto-efficient. This is

the counsel of despair, a cop-out. If there are mis-pricings exogenous to the port, then it is the job of tother entities to go after those mispricings and correct them. The point is not optimality. The point is: are we moving in the right dirction?

3 The Simplest Port

3.1 The Basic Model

With Devanney and Tan, we begin by considering the following extremely simple port:

1. The port offers a *single*, homogeneous cargo handling service. That is, we might imagine a completely specialized port which handles only one commodity. The amount of cargo-handling services performed by the port in some time period, n, can be measured by the throughput of this commodity in this period, x_n, in say, tons. The port's pricing policy through time can also be described by a single number, p_n, in, say, dollars per ton. For exposition's sake, we will assume the period in question is a year, although it could just as easily be a month or a season. Further, we will assume that the period is short enough so that the port is willing to act as if its demand curve is constant over this period.
2. The port consists of a number of berths which are identical in so far as their short-run costs. Each berth has a design capacity of ΔC. At discrete points in time, say once a year, the port has the opportunity to expand. That is, it can order $0, 1, 2, \ldots, K$ new berths, where K should be chosen to be large enough so that we can be sure that the optimal expansion policy will not involve ordering more than this many berths at a single time. We will assume that, if the port decides to expand at the beginning of the nth period, t_n, the new berths will become available at the end of that period. We will also begin by assuming that any such investment will last forever.

   With these assumptions, the state of the port at any decision point t_n is completely described by the number of already installed berths, i. Let the the fixed cost of expanding at t_n be $EC(k, i, t_n)$ where k is the number of berths committed to at t_n. More precisely, the fixed expansion cost, $EC(k, i, t_n)$, is the present value of the time stream of expenses to which the port commits itself when it decides to make the expansion, including any maintenance costs that are independent of throughput. In most real world cases, the fixed cost of a individual berth will increase with both i and k, since the earlier berths will be given the best locations. But the algorithm does not require this.
3. Let $vc(x_n, i)$ be the throughput-dependent expenses associated with handling a quantity x_n in period n given i berths installed at that time. We will assume that vc is a non-increasing function of i and that its derivative with respect to x_n, $mc(x_n, i)$ is a non-decreasing function of x_n. For ports, *for a given* i, marginal cargo handling cost is generally constant up to some level, whereupon it increases sharply, finally becoming vertical at the point where it is impossible to further increase throughput. At this point, the marginal cost to the port of handling a unit of cargo becomes the maximum that a turned-away unit of cargo would have been willing to pay for this service. Thus, our concept of marginal cost includes the "congestion cost" of Allais (1964).
4. Finally, we will assume that the demand for the port's service in period n, $D(p_n, t_n)$ is known and a function only of the price in that period and time.

3.2 A dynamic program for obtaining the optimal pricing and expansion policy

We will assume that the port's cost of capital is constant at $r\%$ per annum and denote the associated discount factor by ρ. We will assume further that the port is willing to act as if demand is constant through an individual period — a year in our case – that is, the port is willing to act as if demand makes a discrete shift to the right at the end of each period and then remains constant through the ensuing period.[3] This implies that the short-run surplus maximizing price will be constant through an individual period.

At the beginning of the nth period, t_n, the port's current situation is completely described by the number of berths in operation, i. At t_n, in this situation, the port has two decisions to make:

1. How much should it charge for its service for the period t_n to t_{n+1}?.
2. Should it order another berth at t_n or not and if so how many?

Given a particular i *at* t_n, the two decisions can be separated, for any new expansion ordered at t_n will not become available until t_{n+1}. In this short run situation, we know the port will maximize society's short run surplus by setting price equal to marginal cost that is, by solving the equation

$$D^{-1}(x^*(i, t_n), t_n) = mc(x^*(i, t_n), i) \tag{1}$$

for $x^*(i, t_n)$ where D^{-1} is the inverse of the demand function. $D^{-1}(x^*(i, t_n))$ is the surplus maximizing price, $p^*(i, t_n)$ in this situation, and the resulting maximum surplus for the period (t_n, t_{n+1}) is

$$S^*(i, t_n) = \int_0^{x^*(i,t_n)} D^{-1}(x^*(i, t_n), t_n) - mc(x^*(i, t_n), i)dx \tag{2}$$

[3] This assumption is made for the purposes of exposition and is easily relaxed. See Section 8.1.

Equation 2 holds whether or not the port decides to expand at t_n because of the construction delay.

Turning to the long-run decision at t_n, let $V_n(i)$ be the maximum present value surplus net of fixed costs obtainable from t_n on, if at t_n the port has i berths in operation. $V_n(i)$ is the present value of future surplus *present valued back to* t_n.

If the port decides not to expand at t_n given i, then the maximum present valued surplus obtainable through the future present valued back to t_n is

$$S^*(i, t_n) + \rho V_{n+1}(i) \tag{3}$$

for at t_{n+1} the port will still have i berths operating, and it will want to follow an optimal policy from that point on.

If on the other hand, the port chooses to add k berths at t_n, then the present value of future surplus assuming optimal operation from t_{n+1} on is

$$S^*(i, t_n) - EC(k, i, t_n) + \rho V_{n+1}(i + k) \tag{4}$$

for at t_{n+1} the port will have $i + k$ berths on-line.

The surplus maximizing port will choose the maximum of these $K+1$ options. Hence we have the following recursion relation,

$$V_n(i) = \max_k \{S^*(i, t_n) - EC(k, i, t_n) + \rho V_{n+1}(i + k) \tag{5}$$

where the maximum is over $k = 0, 1, 2, \ldots, K$ and $EC(0, i, t_n) = 0$. This equation holds for all possible values of installed berths, i, and for all possible $n = 0, 1, 2, 3, ...$; that is, for all possible decision points, t_n.

In order to be able to numerically solve this set of equations, we need a boundary condition on V_n at some time in the future. One such boundary condition follows from supposing that at some time in the relatively distant future, t_N, demand will cease to grow, in which case, under the assumption of infinite investment life, it will be optimal not to order any expansion after t_N, nor will the profit maximizing price change.

Let $S^*(i, t)$ be the resulting surplus obtainable in any period for which $t > t_N$ given that the number of installed berths from t_N on is i. Since i and demand are constant from t_N on, the present value from t_N on given i is

$$V_N(i) = \frac{S^*(i, t_N)}{1 - \rho} \tag{6}$$

yielding the boundary condition at time t_N in the future for all i. Starting with this boundary condition and employing backwards recursion, we can solve for the optimal value function for all $V_n(i)$ and the corresponding surplus maximizing expansion, $k_n^*(i)$, and pricing policy, until we work out way back to $n = 0$, which is usually the present. At this point is a simple matter, to work our way forward starting with the number of already installed berths at t_0 (zero for a new port) picking out the optimal expansion policy, and re-computing the optimal prices for the policy.

4 A Sample Problem involving Linear Demand Curves

4.1 The Devanney and Tan Problem

A computer program implementing the above dynamic program has been written. We have exercised it on Devanney and Tan's sample problem.

1. Demand linear in price with exponentially decreasing growth

$$D(p, t) = (1 - e^{-\gamma t})(\alpha - \beta p) \tag{7}$$

   In the sample, linear exercises in this paper, we have held the demand surface constant, setting $\alpha = 10^6$ tons, $\beta = 10^4$ tons per dollar, and $\gamma = 0.1$. This demand surface is shown in Figure 1. For this demand surface, price can run between \$100 per ton and \$0 per ton, and the resulting throughput will be between 0 tons and a number which is 0 at $t = 0$ but fairly rapidly approaches 1,000,000 tons per year as t approaches 40 or so. For this demand surface, we have taken t_N to be 50 since practically all the growth has taken place by this time.
2. The fixed cost of a single berth, EC is the same for all berths.
3. Marginal costs of each berth are identical and quadratic in throughput. Given identical marginal costs, the optimal port will distribute its throughput, x, evenly among each of the berths. Thus, at any time, x/i tons of cargo will be flowing through each berth. The marginal cost function which was used in the sample problems was

$$mc(x, i) = \frac{3EC(1 - \rho)}{2\Delta C^3}(x/i)^2 \tag{8}$$

   where the constant $1.5EC(1-\rho)/\Delta C^3$ has been chosen to make the average cost curve minimum when throughout equals design capacity.[4]
4. $K = 1$. The DandT algorithm only allowed the port the option of ordering at most one new berth at each decision point.

The results of these sample calculations for $\rho = 0.9$, $EC = 10^6$ and $\Delta C = 10^5$ tons per year, $50,000$ tons per year and $25,000$ tons per year are shown in Figures 2, 4 and 6 respectively. The corresponding DandT results are shown in Figures 3, 5 and 7 respectively.

The results are quite similar. However, the surplus maximizing port sometimes brings on its capacity slightly sooner and in the case of the two lower ΔC's ends up with one more berth.[5] The present value profit for the surplus maximizing port tends to be lower than that for the DandT efficient port and can be negative. DandT accepted as a constraint that no berth would be ordered until that berth had a positive net prevent value. Given the discrete nature of the investment alternatives, this is not necessarily optimal from the point of view of society as a whole. The surplus maximizing port relaxes that constraint but still ends up with an overall NPV close to zero *as long as its expansions can keep up with demand growth.*

In Figures 4 and 5 and especially in Figures 6 and 7 the port can't keep up with demand growth due to the constraint that it can only bring on one new berth per year. As a result, in both cases, the port makes a substantial profit. From the point of view of societal income, this is a bad sign. It indicates that, if we were to relax the one berth per year constraint, the economically efficient port would take advantage of this relaxation. However, given the constraint, the port must allocate its scarce resources efficiently. Hence the high prices.

In the next six figures, we re-ran these three cases multiplying EC by 10. A berth now costs \$10,000,000. Once again the results are quite similar; but with the big increase in fixed costs, the relaxation of the never-expand-until-berth-NPV-positive constraint assumes a bit more importance. The surplus maximizing port brings on capacity noticeably sooner which means a bigger discrepancy in overall port NPV.

4.2 The Distribution of the Surplus

But the important point to notice is that, in every case in which the surplus maximizing port is free to keep up with demands, the port's NPV is near zero. To put it another way, the consumer gets (almost) all the surplus and the port (nearly) breaks even. At first glance, this seems almost magical. Equation 5 makes no distinction between consumers' and producers' surplus. Unlike DandT's algorithm, there is nothing explicit in our dynamic program that would seem to "favor" the consumer. However, an inspection of the figures reveals what is happening. The algorithm works hard to keep throughput close to design capacity. There

[4] This is a purely expositional convenience. The entire line of reasoning does not depend on the concept of "design capacity" in any fundamental way. Nor does it depend on "average cost", which strictly speaking applies only to the steady-state situation.

[5] The "Total throughput" shown is slightly misleading. It is the "Total throughput" between 0 and t_N. It does not include the throughput at steady state which will be slightly higher for the surplus maximizing port for the small design capacities.

is a substantial penalty for not doing so. But at design capacity, average cost is at a minimum at which point average cost equals marginal cost. But our port always charges marginal cost. Therefore to the extent that the port is able to keep throughput at design capacity, it will be charging average cost; but a port that charges average cost will just break even.

Very loosely speaking, this can be regarded as a manifestation of the Second Fundamental Theorem of Welfare Economics. Our port's expansion and pricing policy is Pareto-optimal. Therefore, by the 2nd Theorem, it can be realized by a competitive market equilibrium. But in a competitive market equilibrium, the producers compete away their surplus to the consumer.[6]

4.3 Results Insensitive to The Shape of the Demand Surface

Another possibly surprising feature of the algorithm is that the results are insensitive to the shape of the demand surface in the sense that ***any demand surface that results in the same intersections of demand and marginal cost through time will produce the same expansion policy***.

It is obvious from Equation 2 that the shape of the demand curve (and marginal cost curve) to the right of price equals marginal cost is irrelevent since this portion of the curve never enters into the calculation. It is only slightly less easy to see that adding or subtracting area under the demand curve to the left of the optimal price for any given t_n will have no effect on the allocation. The reason is that adding or subtracting a constant to every V_n will not change the optimal choice of k in Equation 5 as long as the consumer's surplus stays positive.[7] So if you knew what the optimal pricing was through time, you could cut off the demand curve for that time — make it horizontal — at a point a little above that price.

From a naieve point of view, this feature of the algorithm might be a little hard to swallow. After all if there is this tremendous amount of willingness to pay associated with a demand curve with tremendously large area to the left of the optimal price point wouldn't society somehow be better off if we brought some berths on sooner. The algorithm — and classical theory — say no. Bringing berths on sooner pushes the marginal cost ***down***, so the only thing that determines the ***additional*** consumer surplus associated with this change is the shape of the demand curve ***below*** the old price and above the new price. Marginal analysis rules.

This observation is important practically for in the real world the port's hardest job will be to try and estimate the demand surface it is facing. If we are in a situation in which we know the optimal policy will result in prices that won't stray too far from average cost, then the port need only estimate its demand curves through time in the vicinity of average cost.

Finally, a very important important corollary is that the algorithm won't build another berth unless that berth (nearly) breaks-even. Clearly, the algorithm won't add another berth unless the net surplus associated with that berth is positive. But since we can cut off demand at just above the intersection of demand and supply through time, we can generate a demand surface for this expansion with nearly zero consumers' surplus. This means the producers' surplus associated with the expansion needs to be positive or nearly so.

4.4 Marginal Cost Pricing's Supposed Requirement for Subsidies

If one keeps EC at 10^7 but increases the sample design capacity to 1,000,000 tons per year, the DandT port won't invest in any berths; but, if you drop the design capacity to 500,000 tons per year, the DandT algorithm invests in one berth and runs 37,000,000 tons of cargo through it in the first 50 years. DandT seems to imply that, if our cargo handling technology gets good enough, we don't handle any cargo. The problem is that with a ΔC of a million tons per year and the sample demand surface, it is impossible ***even with only one berth*** to get on the upside of the average cost curve, ***even when demand has reached its full growth***. Charging marginal costs, as it must, there is no way the port can pay for even the first berth, so under the DandT constraint, the first berth is never ordered. This situation — decreasing average costs ***after full growth has been achieved for the smallest possible investment*** — is handled correctly by the surplus maximizing port; but it will need a subsidy as Figure 14 makes clear. This is the only situation in these runs in which the surplus maximizing port requires a significant subsidy.

This requirement for a marginal cost pricing entity to not need a subsidy is much weaker than most static analyses indicate. Static analyses tend to equate large indivisible fixed investment with decreasing average costs, and then immediately argue that marginal cost pricing is not possible without subsidies in the face of large indivisible fixed investment, an argument that is cheerfully used to justify monopolistic pricing or taxpayer subsidies or the like by most port authorities. But viewed dynamically, we see that the situations in which marginal cost pricing actually requires a subsidy are far, far rarer than the static analyses would have

[6] Provided fixed costs are not increasing. See Section 6.

[7] The last caveat is needed because the program (and society) has the option of doing nothing, making no investment at all.

us believe. In fact, in the port context, it is almost impossible to imagine a real world situation in which we would have decreasing average costs for the smallest possible investment at full demand growth.

Figure 1: Linear Demand Surface

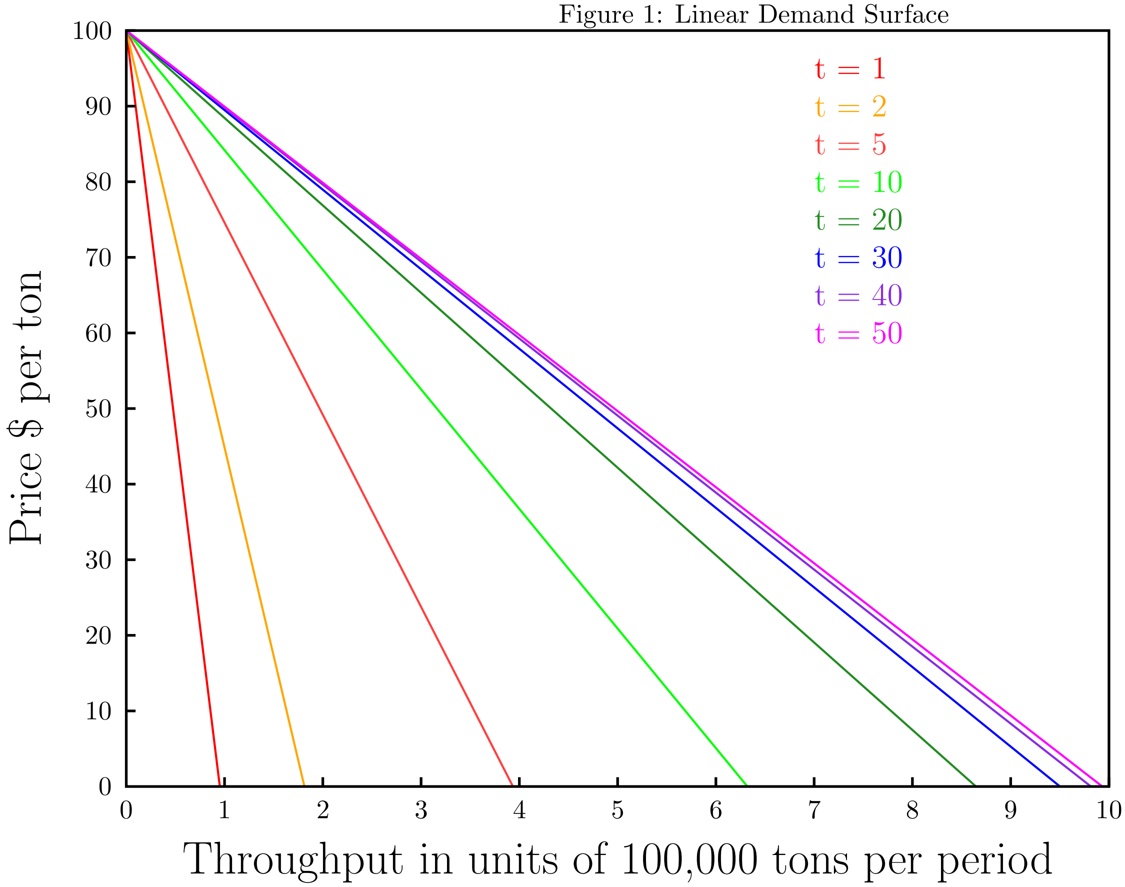


Figure 2: Surplus Maximizing Policy, $\Delta C = 10^5$, $EC = 10^6$

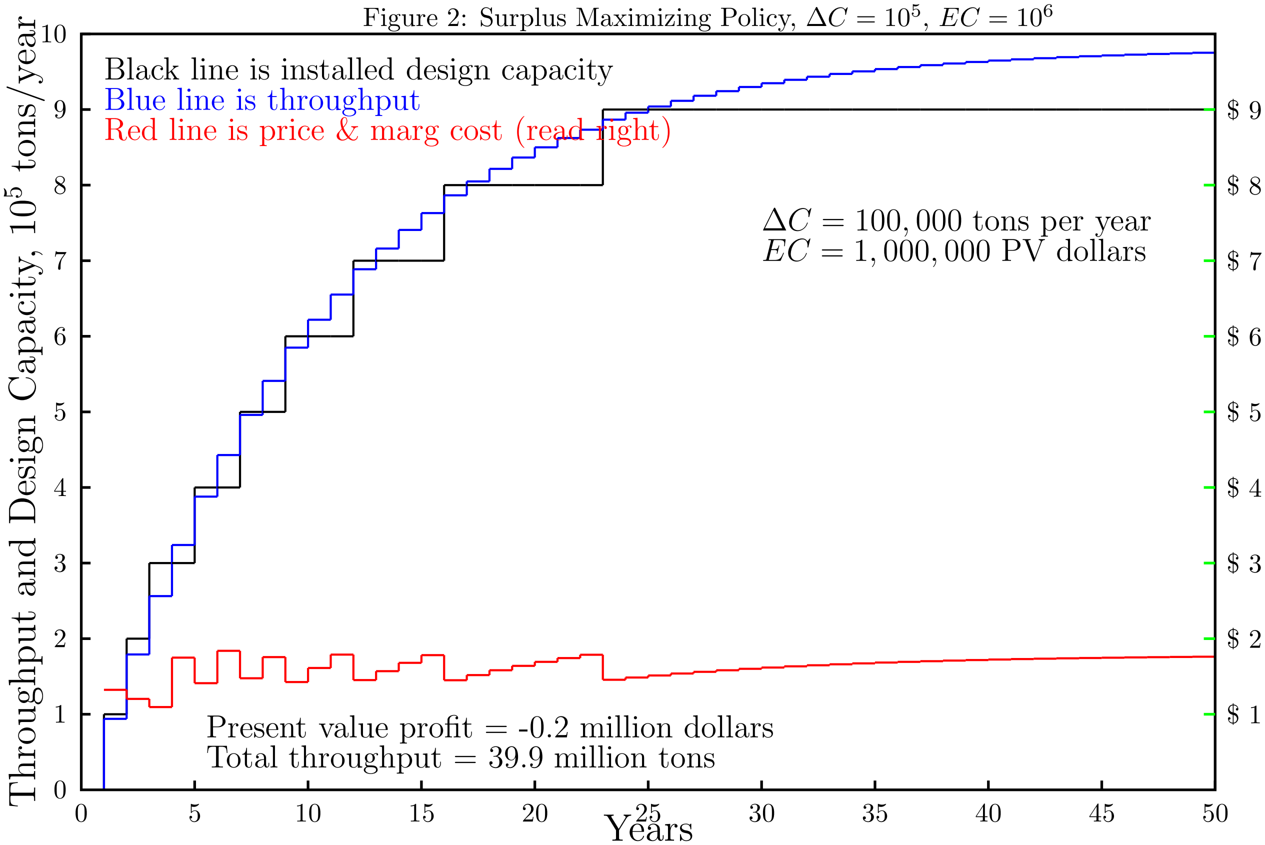


Figure 3: DandT Efficient Policy, $\Delta C = 10^5$, $EC = 10^6$

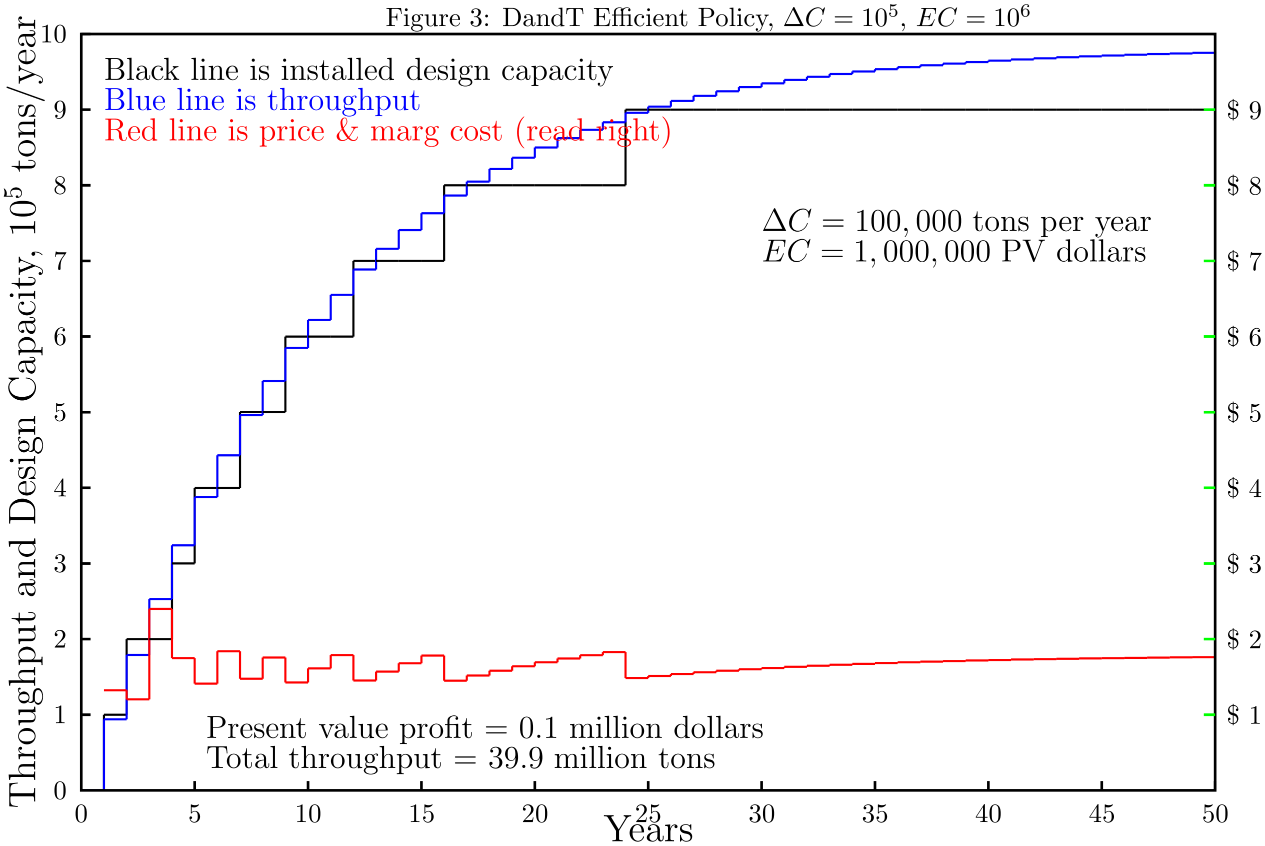


Figure 4: Surplus Maximizing Policy, $\Delta C = 50,000$, $EC = 10^6$

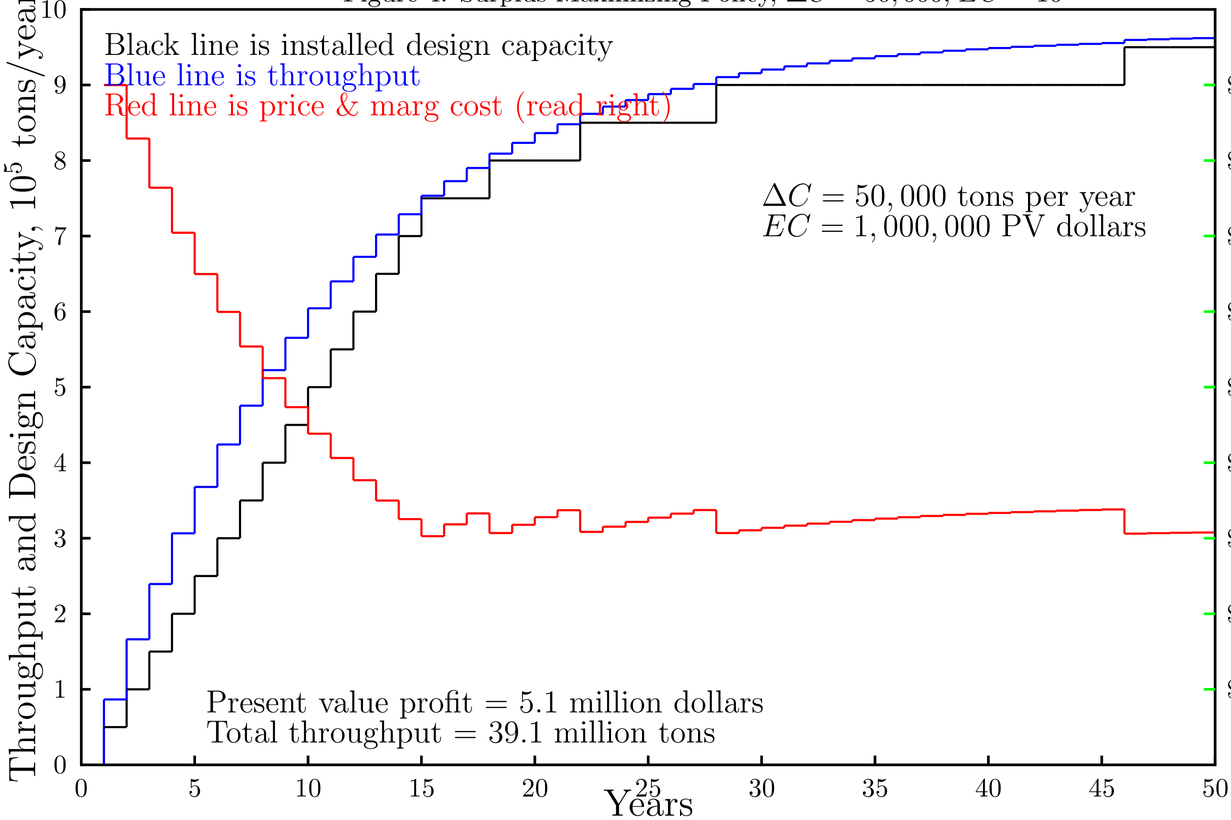
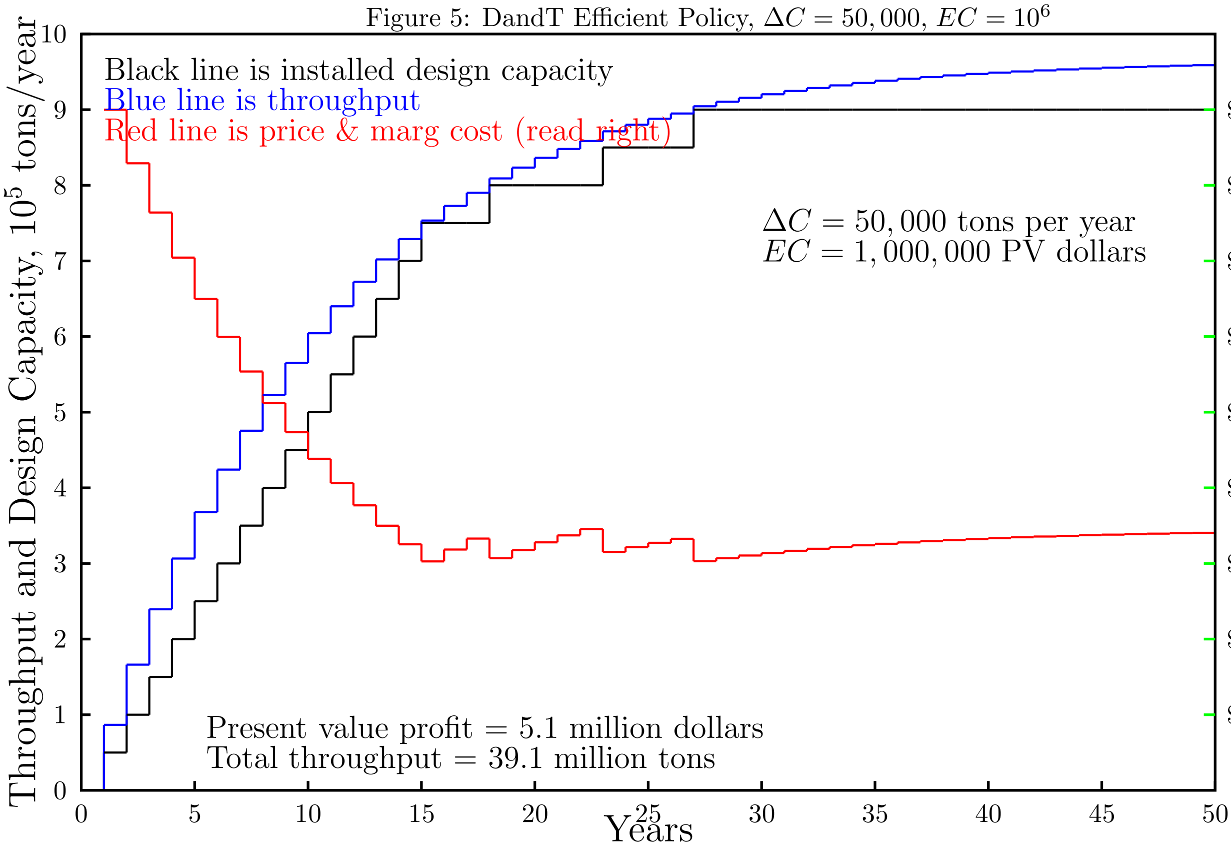


Figure 5: DandT Efficient Policy, $\Delta C = 50,000$, $EC = 10^6$



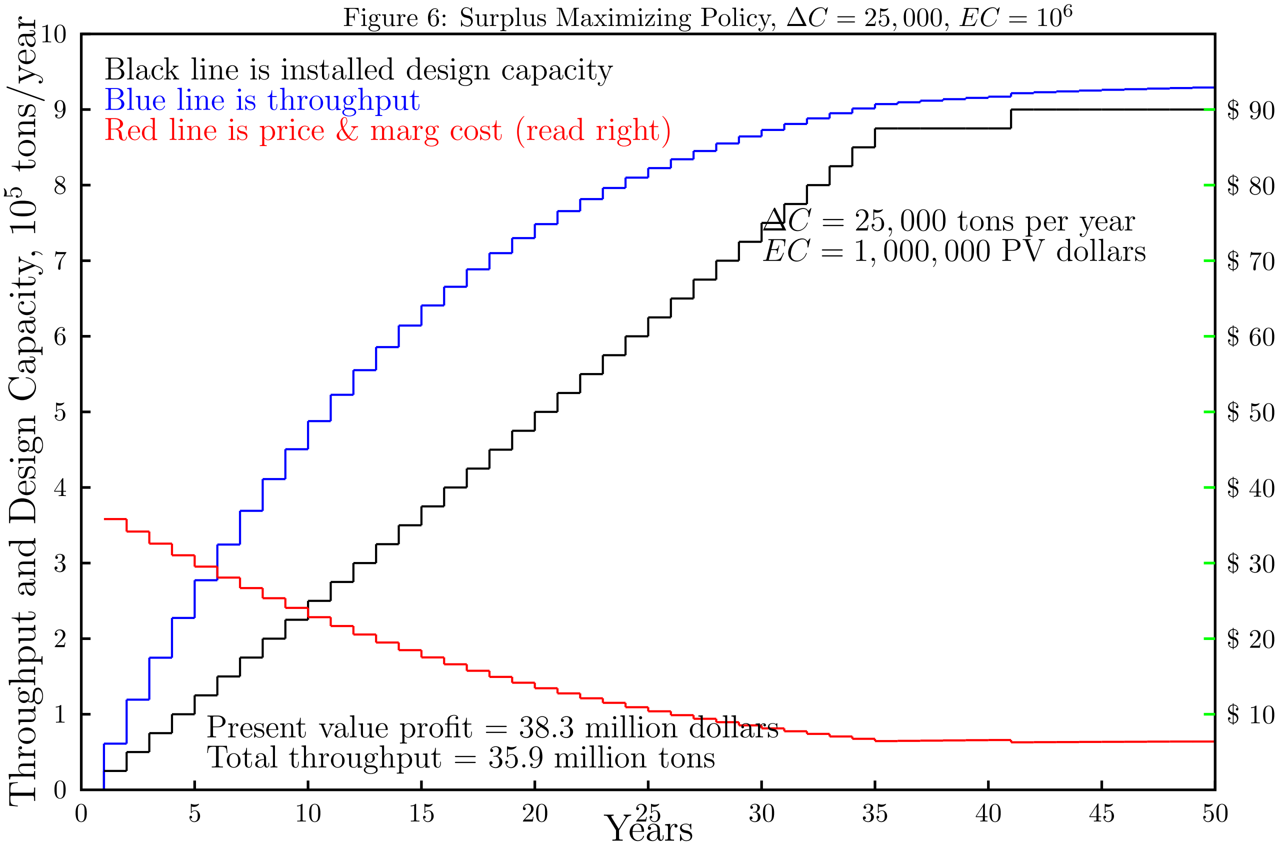


Figure 6: Surplus Maximizing Policy, $\Delta C = 25,000$, $EC = 10^6$
Black line is installed design capacity
Blue line is throughput
Red line is price & marg cost (read right)
$\Delta C = 25,000$ tons per year
$EC = 1,000,000$ PV dollars
Present value profit = 38.3 million dollars
Total throughput = 35.9 million tons
Throughput and Design Capacity, 10^5 tons/year
Years
0
1
2
3
4
5
6
7
8
9
10
0
5
10
15
20
25
30
35
40
45
50
$ 10
$ 20
$ 30
$ 40
$ 50
$ 60
$ 70
$ 80
$ 90

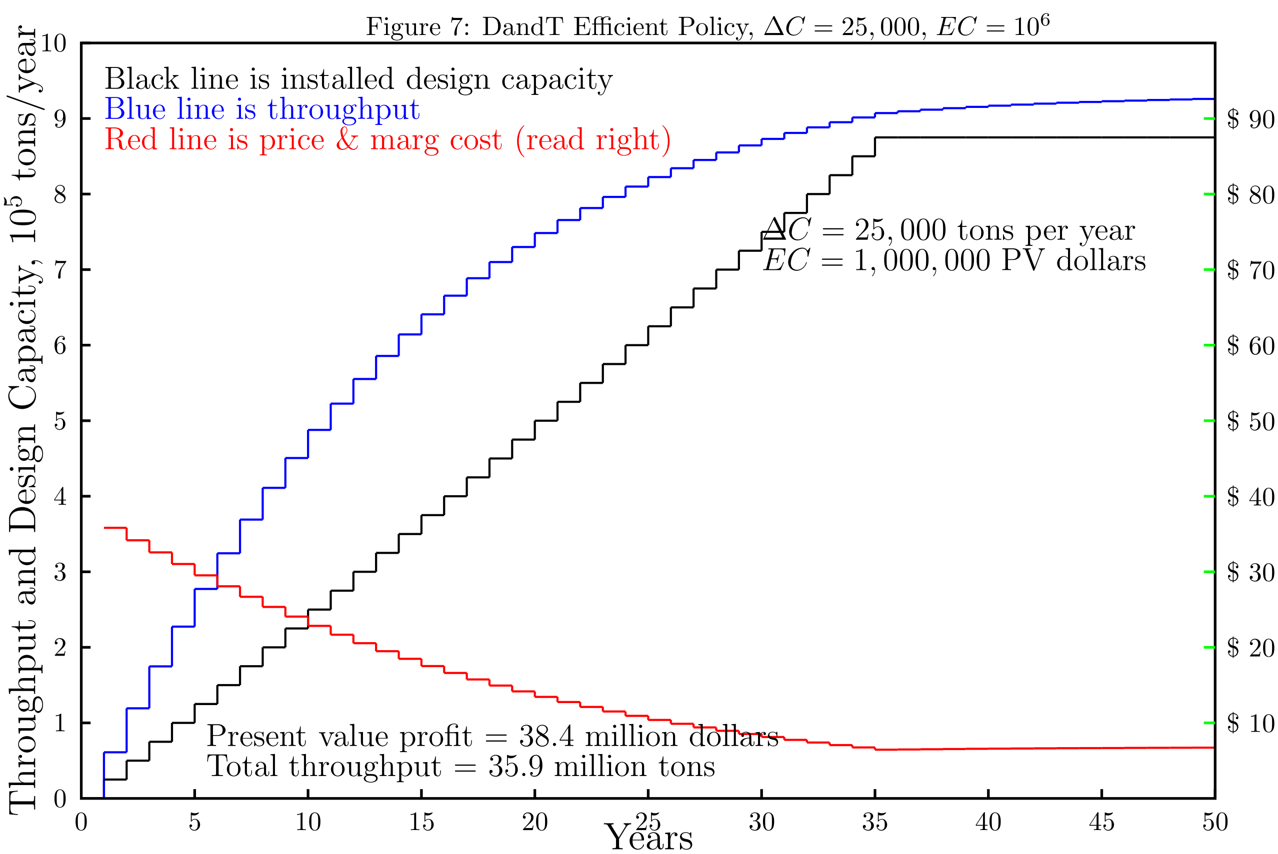


Figure 7: DandT Efficient Policy, $\Delta C = 25,000$, $EC = 10^6$
Black line is installed design capacity
Blue line is throughput
Red line is price & marg cost (read right)
$\Delta C = 25,000$ tons per year
$EC = 1,000,000$ PV dollars
Present value profit = 38.4 million dollars
Total throughput = 35.9 million tons
Throughput and Design Capacity, 10^5 tons/year
Years
0
1
2
3
4
5
6
7
8
9
10
0
5
10
15
20
25
30
35
40
45
50
$ 10
$ 20
$ 30
$ 40
$ 50
$ 60
$ 70
$ 80
$ 90

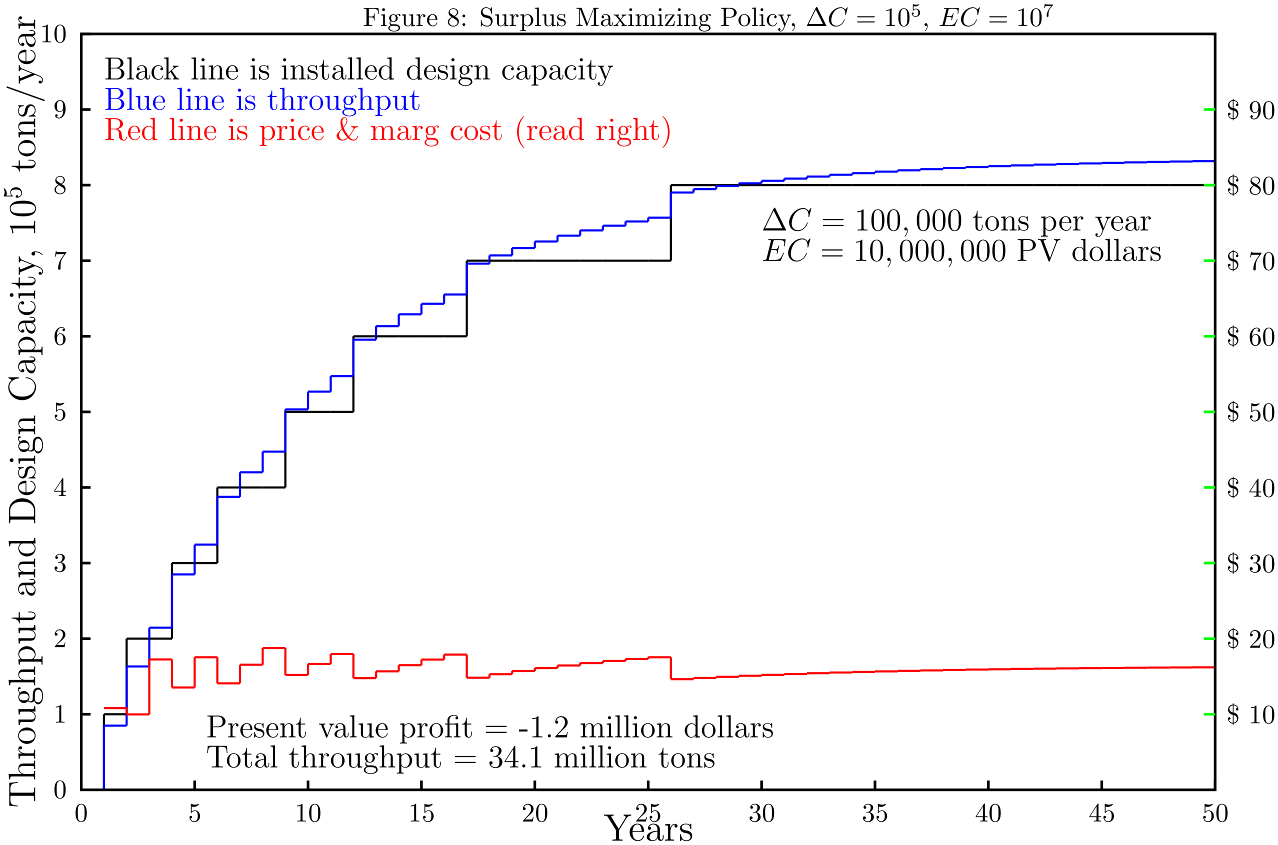


Figure 8: Surplus Maximizing Policy, $\Delta C = 10^5$, $EC = 10^7$
Black line is installed design capacity
Blue line is throughput
Red line is price & marg cost (read right)
$\Delta C = 100,000$ tons per year
$EC = 10,000,000$ PV dollars
Present value profit = -1.2 million dollars
Total throughput = 34.1 million tons
Throughput and Design Capacity, 10^5 tons/year
Years
0
5
10
15
20
25
30
35
40
45
50
0
1
2
3
4
5
6
7
8
9
10
$ 10
$ 20
$ 30
$ 40
$ 50
$ 60
$ 70
$ 80
$ 90

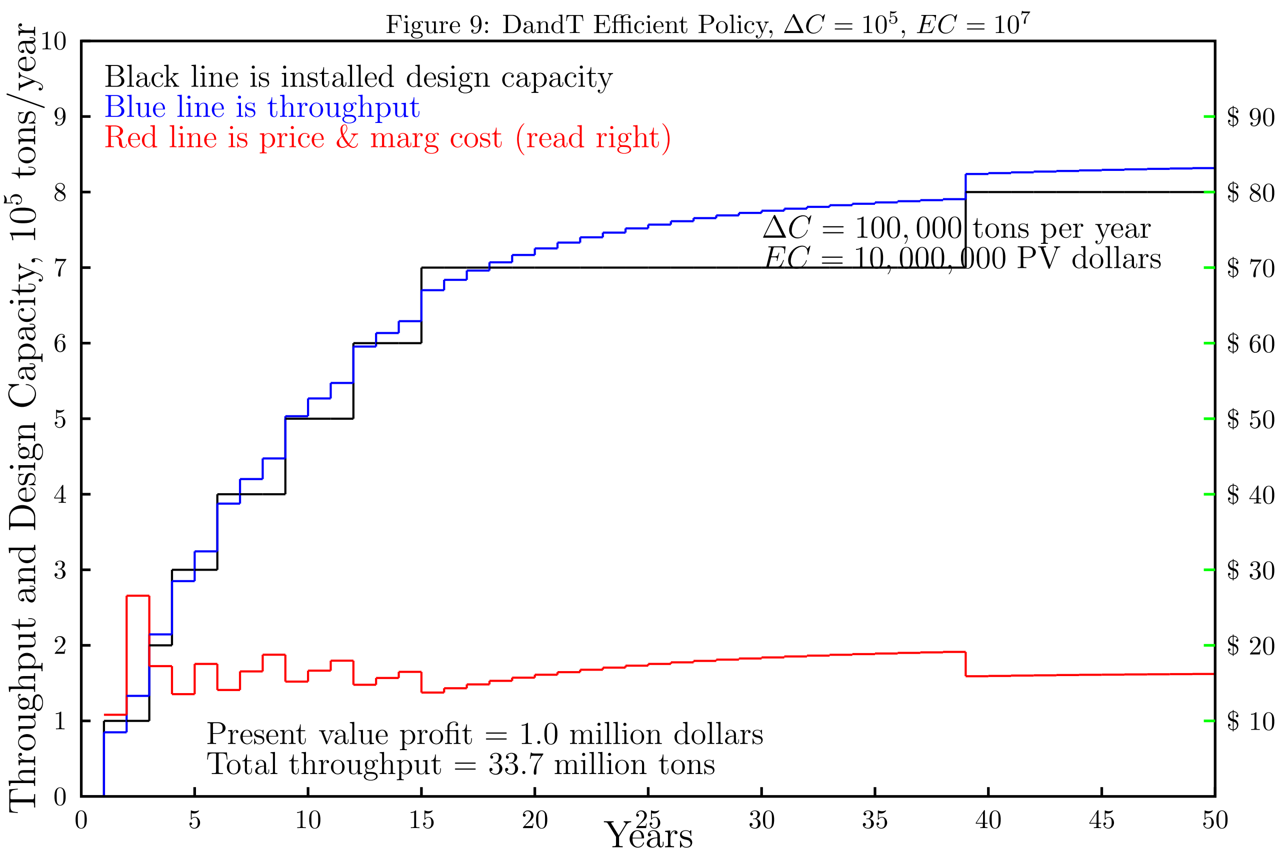


Figure 9: DandT Efficient Policy, $\Delta C = 10^5$, $EC = 10^7$
Black line is installed design capacity
Blue line is throughput
Red line is price & marg cost (read right)
$\Delta C = 100,000$ tons per year
$EC = 10,000,000$ PV dollars
Present value profit = 1.0 million dollars
Total throughput = 33.7 million tons
Throughput and Design Capacity, 10^5 tons/year
Years
0
5
10
15
20
25
30
35
40
45
50
0
1
2
3
4
5
6
7
8
9
10
$ 10
$ 20
$ 30
$ 40
$ 50
$ 60
$ 70
$ 80
$ 90

Figure 10: Surplus Maximizing Policy, $\Delta C = 50,000$, $EC = 10^7$

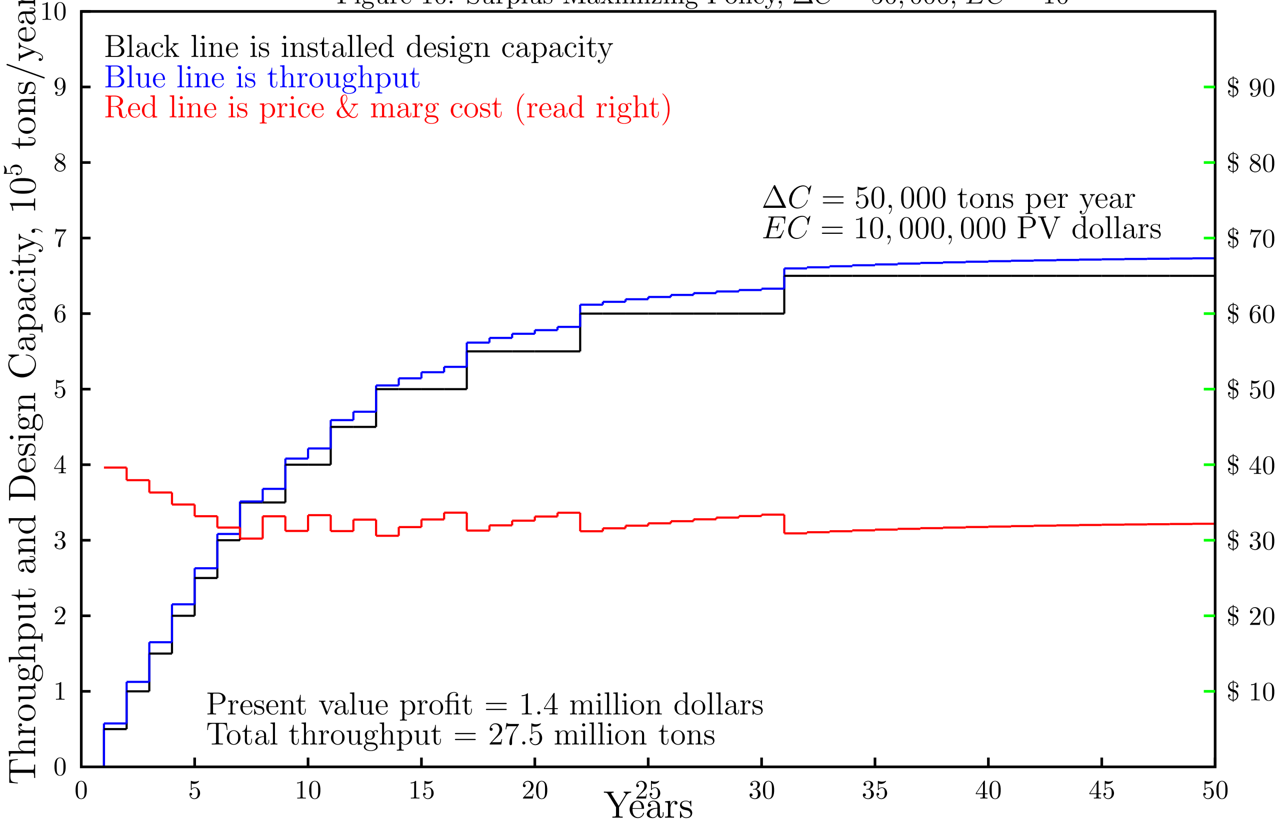


Figure 11: DandT Efficient Policy, $\Delta C = 50,000$, $EC = 10^7$

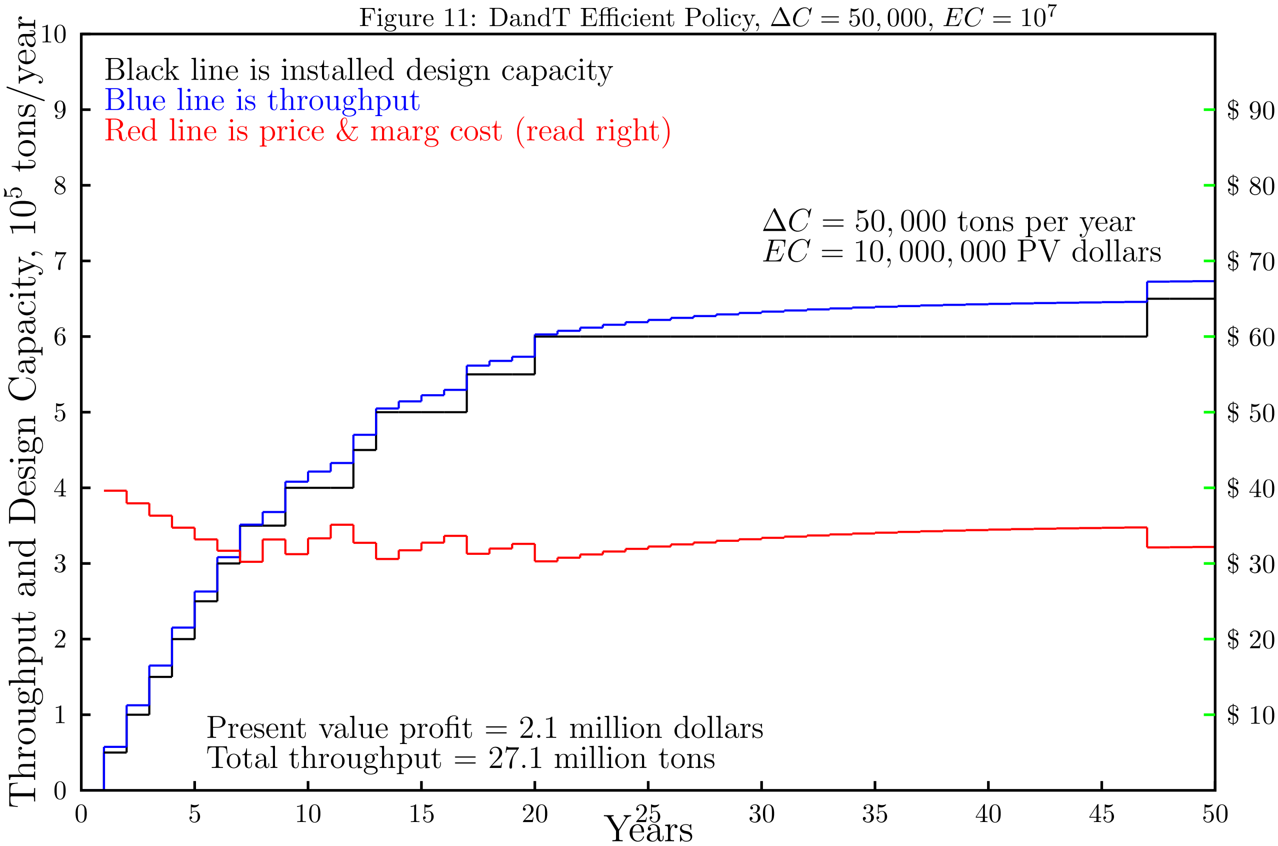


Figure 12: Surplus Maximizing Policy, $\Delta C = 25,000$, $EC = 10^7$

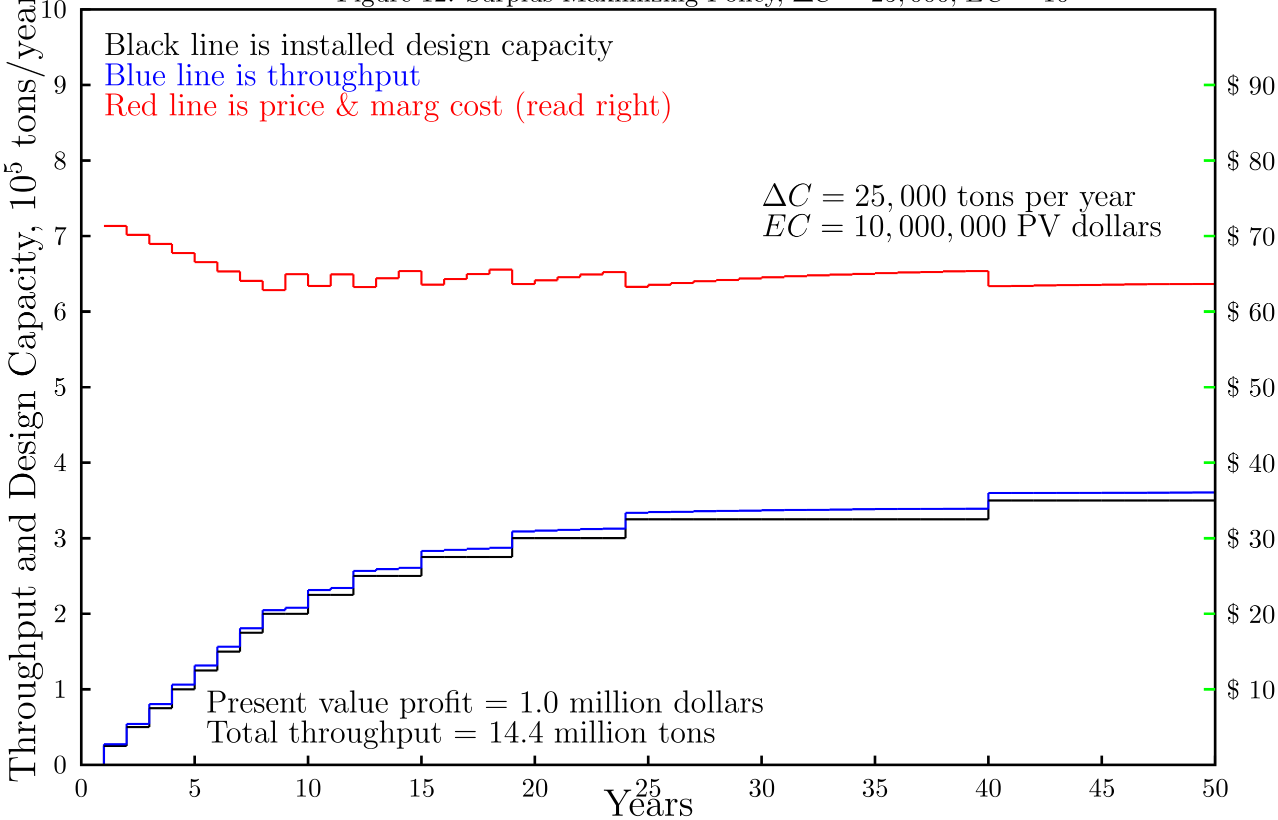


Figure 13: DandT Efficient Policy, $\Delta C = 25,000$, $EC = 10^7$

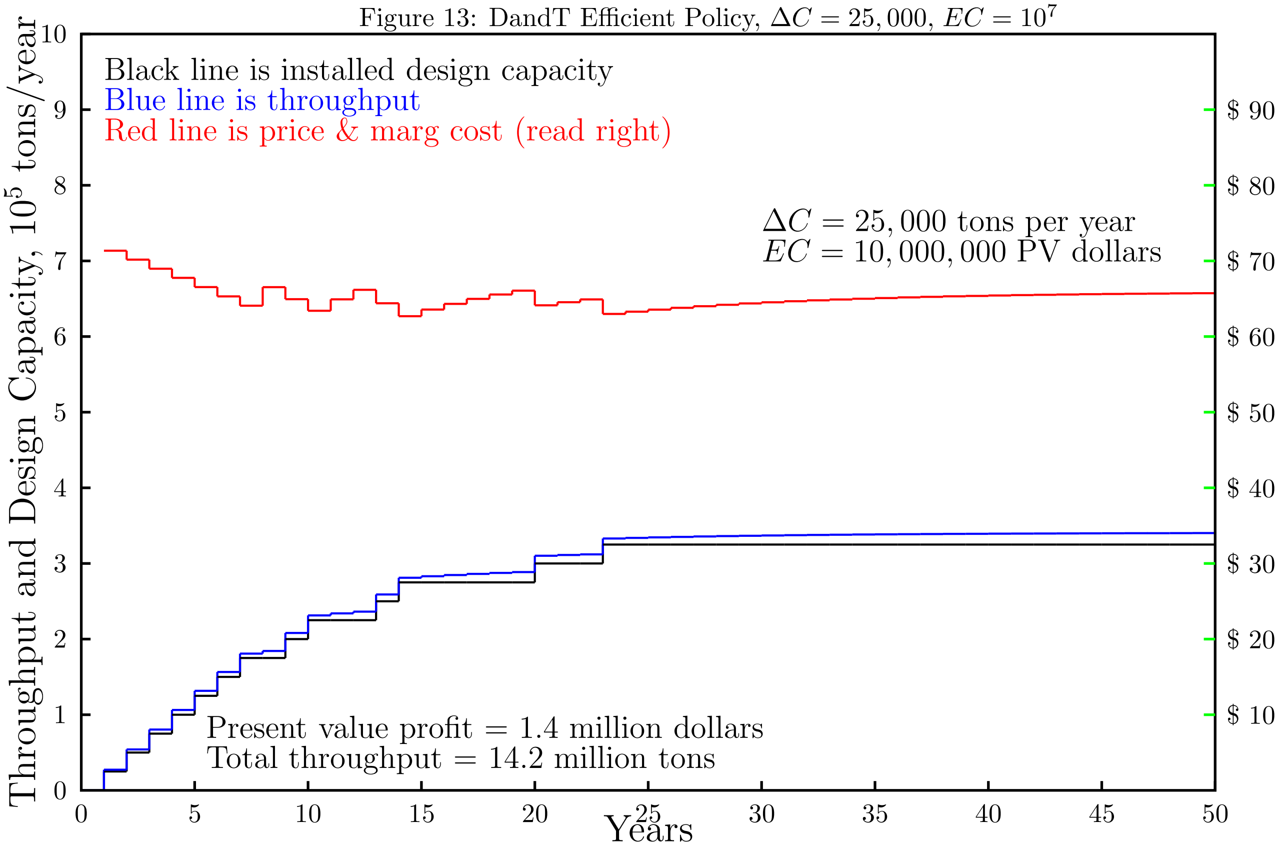
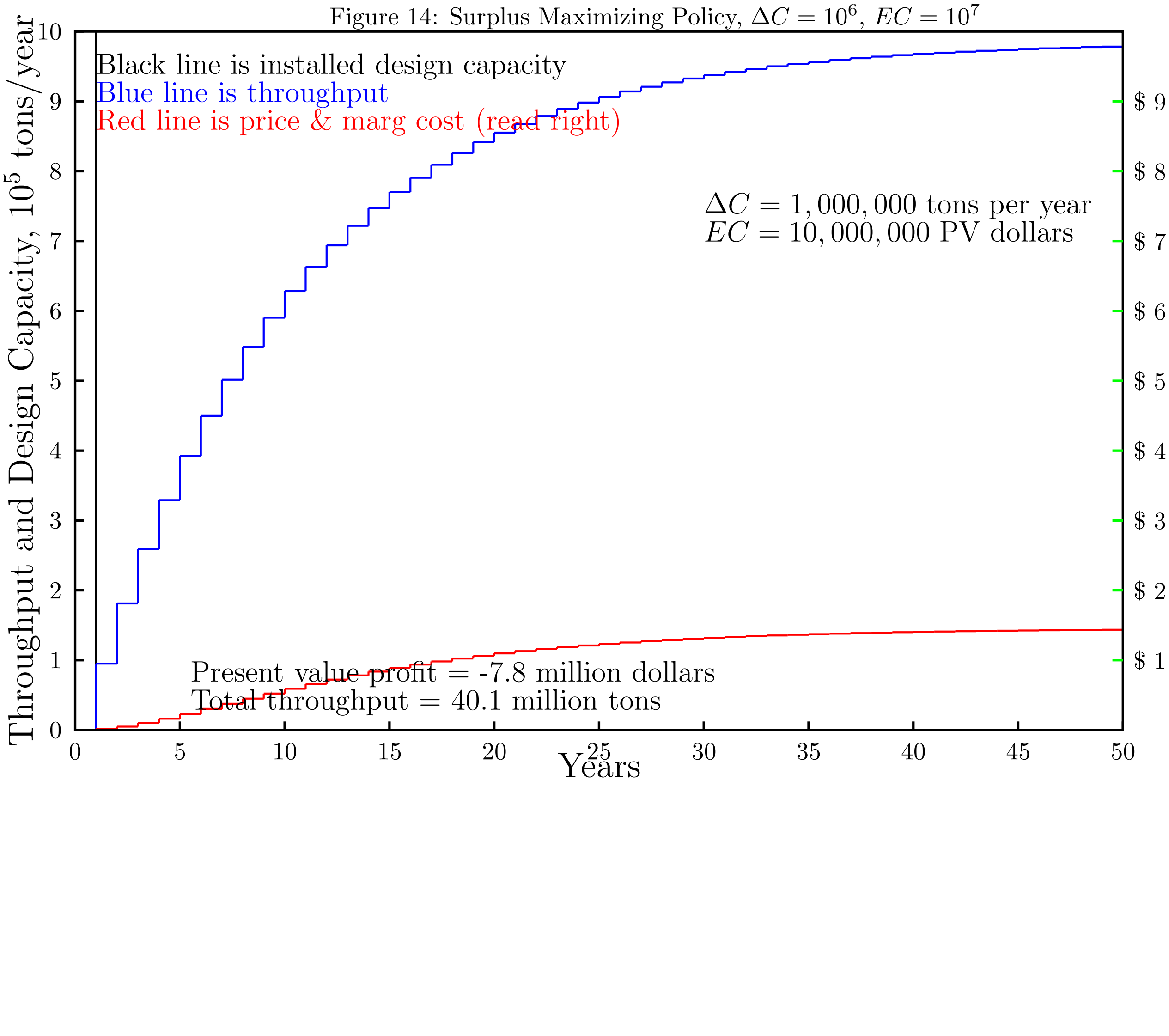


Figure 14: Surplus Maximizing Policy, $\Delta C = 10^6$, $EC = 10^7$



5 Constant Elasticity Demand Surface

To study how the algorithm reacts to non-linear demand curves, we examined the demand surface shown in Figure 15. Once again we are assuming exponentially decreasing growth. But, for any given t_n, the demand curve has a constant elasticity of 2.[8] Notice that these demand curves become nearly horizontal for prices below about \$10/ton. This is totally unrealistic in the port context, but serves to provide a tough test for the algorithm. This demand surface has the following equation.

$$D(p,t) = (1 - e^{-\gamma t})(10^8 p^{-2}) \tag{9}$$

For this set of exercises, we assumed the marginal costs of each berth are identical and cubic in throughput.

$$mc(x,i) = \frac{4EC(1-\rho)}{3\Delta C^4}(x/i)^3 \tag{10}$$

where the constant $1.25EC(1-\rho)/\Delta C^4$ has been chosen to make the average cost curve minimum when throughout equals design capacity.

The results for $\rho = 0.9$, $EC = 10^7$ and $\Delta C = 10^5$ tons per year, $50,000$ tons per year and $25,000$ tons per year are shown in Figures 16, 17 and 18 respectively.

Once again we note that the optimal port alternates periods of over-supply and under-supply in a manner that ends up with the port nearly breaking even in present valued terms. In Figure 18 where the minimum average cost is over \$60 per ton, this means that the optimal port can invest in only one berth given the demand surface. Yet despite the very high fixed cost, \$400 per ton/year of capacity, the marginal cost charging port essentially breaks even. To do so, it has to stay in the far left hand portion of Figure 15.

If we reduce the berth fixed costs to a million dollars, then we are dealing with minimum average costs in the range of \$5.77 per ton for a design capacity of 25,000 tons per year to \$1.44 for $\Delta C = 100,000$ tons per year. A glance at the demand surface, Figure 15, we reveal that this will imply very large throughputs. Indeed the following table summarizes the results.

Table 1: Results for $EC = 10^6$ and constant elasticity = 2

	DC=25,000	DC=50,000	DC=100,000
Min Average Cost(\$/ton)	5.77	2.89	1.44
Berths at year 50	116	232	464
Thruput at year 50	2,978,613	11,919,151	47,656,302
Port NPV (millions)	+0.019	-0.006	-0.049

Since the optimal policy ends up with such a large throughput and a correspondingly large number of berths, it is possible for program to keep throughput very close to design capacity throughout. To put it another way, the smallest increment of fixed investment is small compared to the throughput. As a result, the port's overall net present value is very close to zero, never more than 5% the cost of an individual berth, even though we are talking about some enormously large gross numbers.

These runs offer further "empirical" evidence, that an optimal, marginal cost pricing port will (nearly) break even in present value terms, whether the investment increment is large or small relative to throughput, whether fixed costs are high or low, as long as the port can get on the up side of the average cost curve ***for the smallest possible investment at full demand growth.*** In the next section, we will find that we have to add one more requirement: non-decreasing fixed costs.

[8] Since we require a finite area under the curve, the elasticity has to be greater than 1 unless we chop off the demand curve at the high price end, which we are allowed to do by Section 4.3.

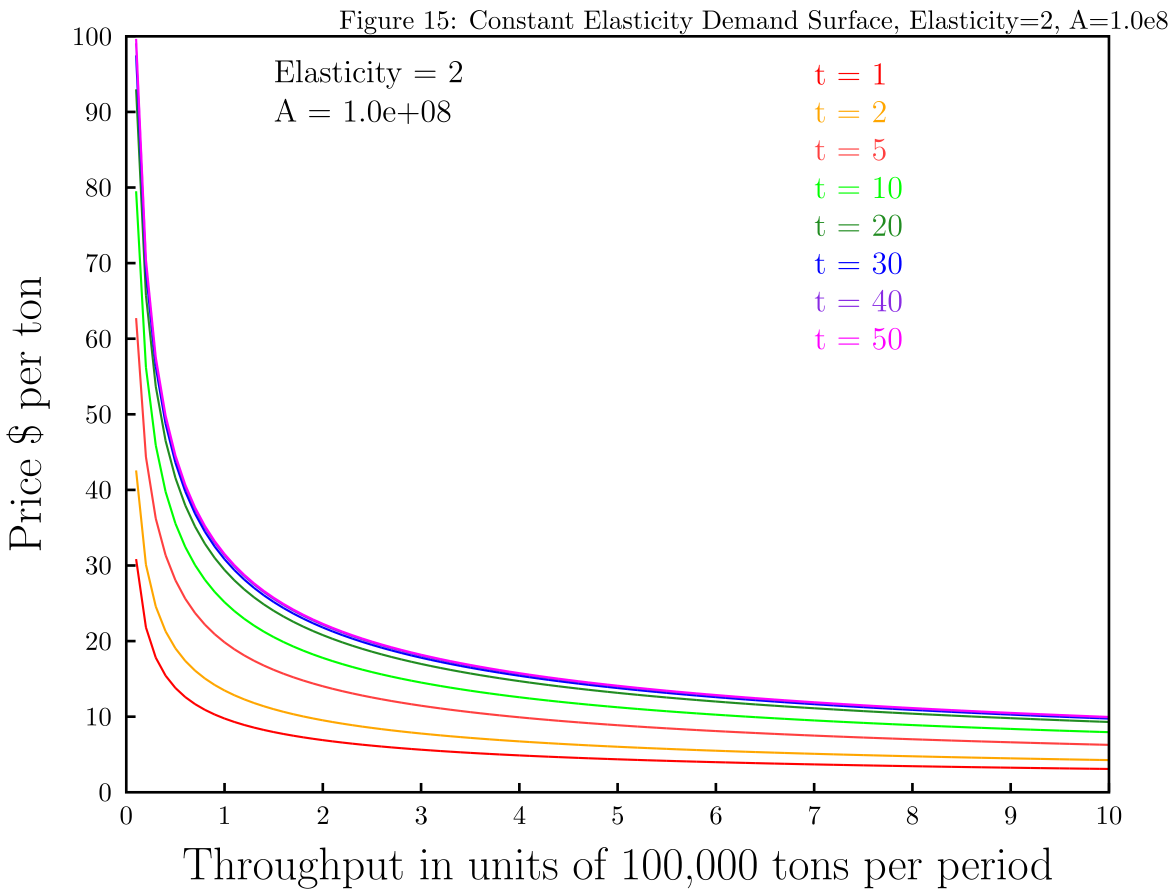


Figure 15: Constant Elasticity Demand Surface, Elasticity=2, A=1.0e8
Elasticity = 2
A = 1.0e+08
t = 1
t = 2
t = 5
t = 10
t = 20
t = 30
t = 40
t = 50
Price $ per ton
Throughput in units of 100,000 tons per period

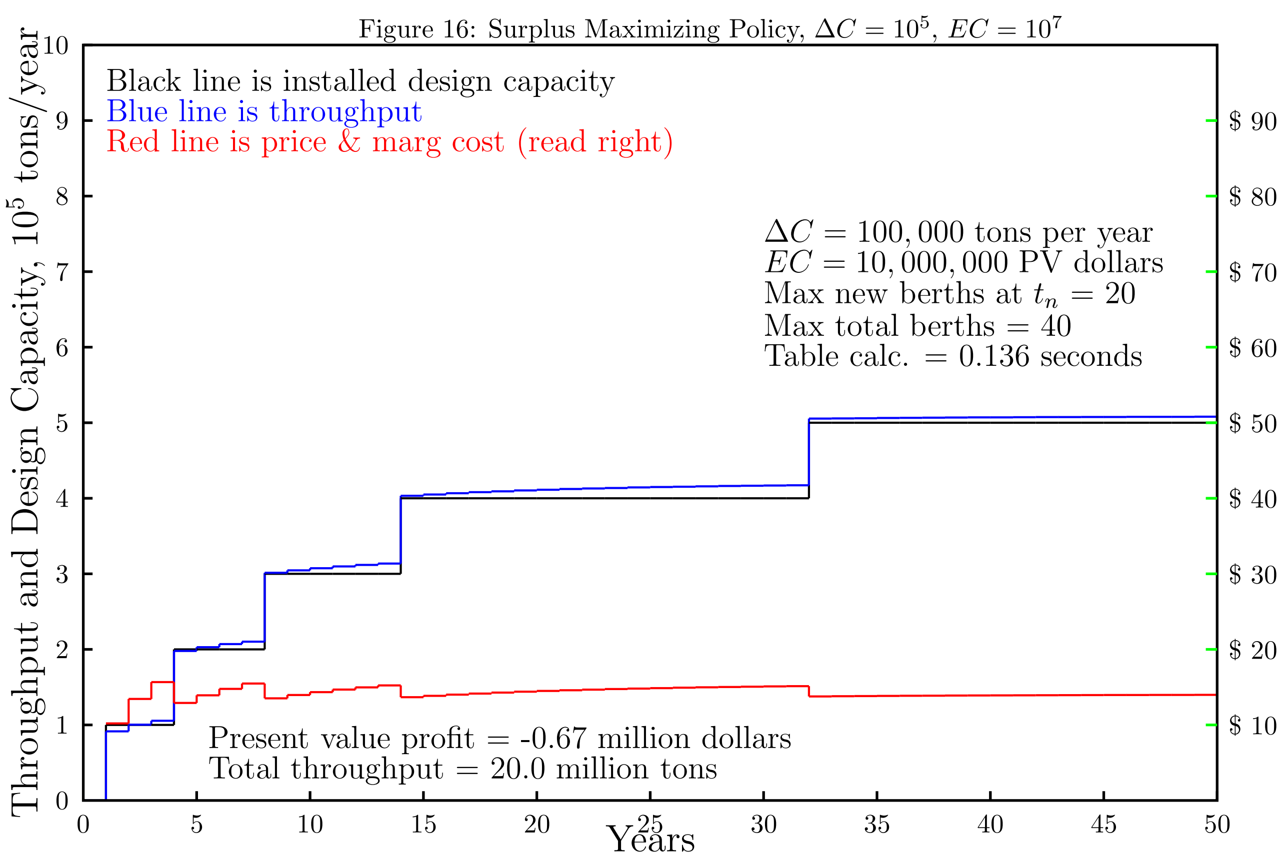


Figure 16: Surplus Maximizing Policy, $\Delta C = 10^5$, $EC = 10^7$
Black line is installed design capacity
Blue line is throughput
Red line is price & marg cost (read right)
$\Delta C = 100,000$ tons per year
$EC = 10,000,000$ PV dollars
Max new berths at $t_n = 20$
Max total berths = 40
Table calc. = 0.136 seconds
Present value profit = -0.67 million dollars
Total throughput = 20.0 million tons
Throughput and Design Capacity, 10^5 tons/year
Years

Figure 17: Surplus Maximizing Policy, $\Delta C = 50,000$, $EC = 10^7$

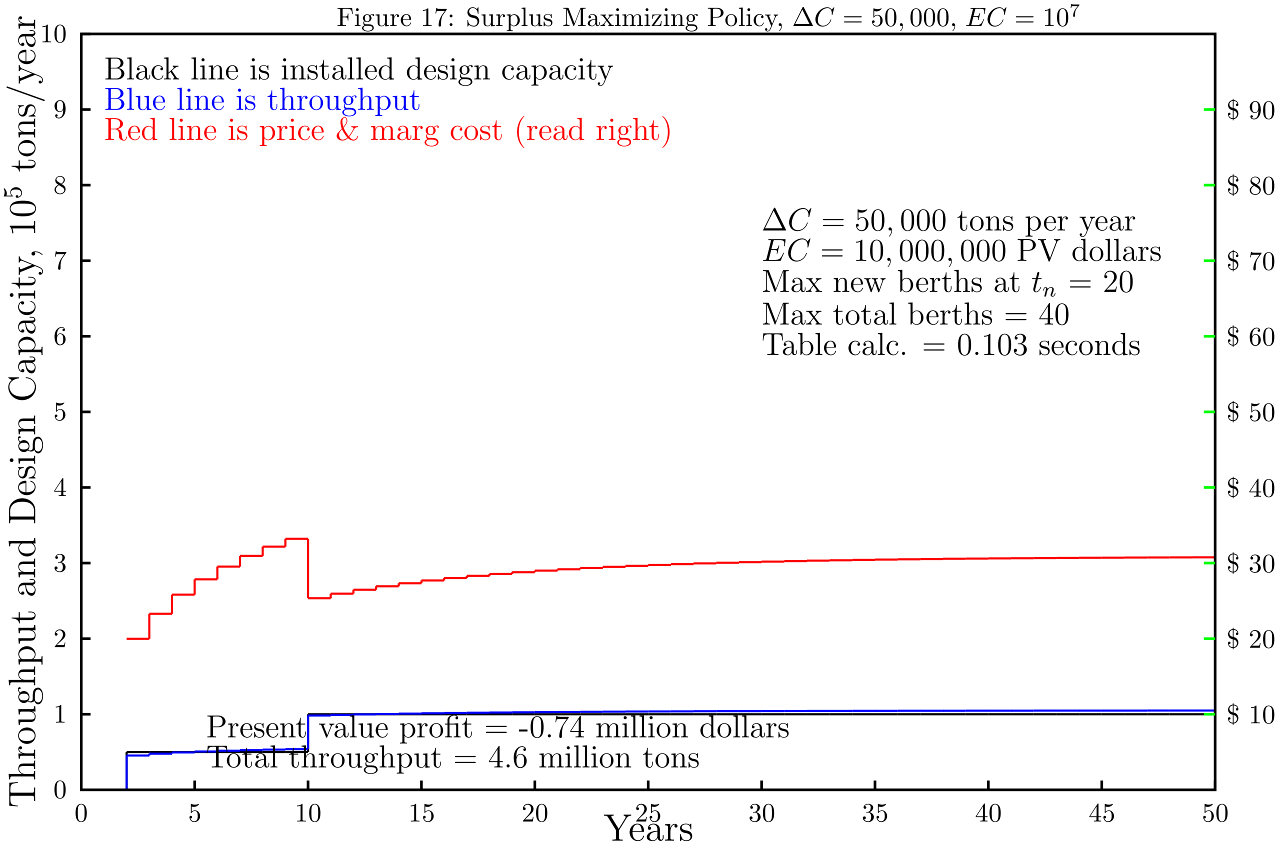
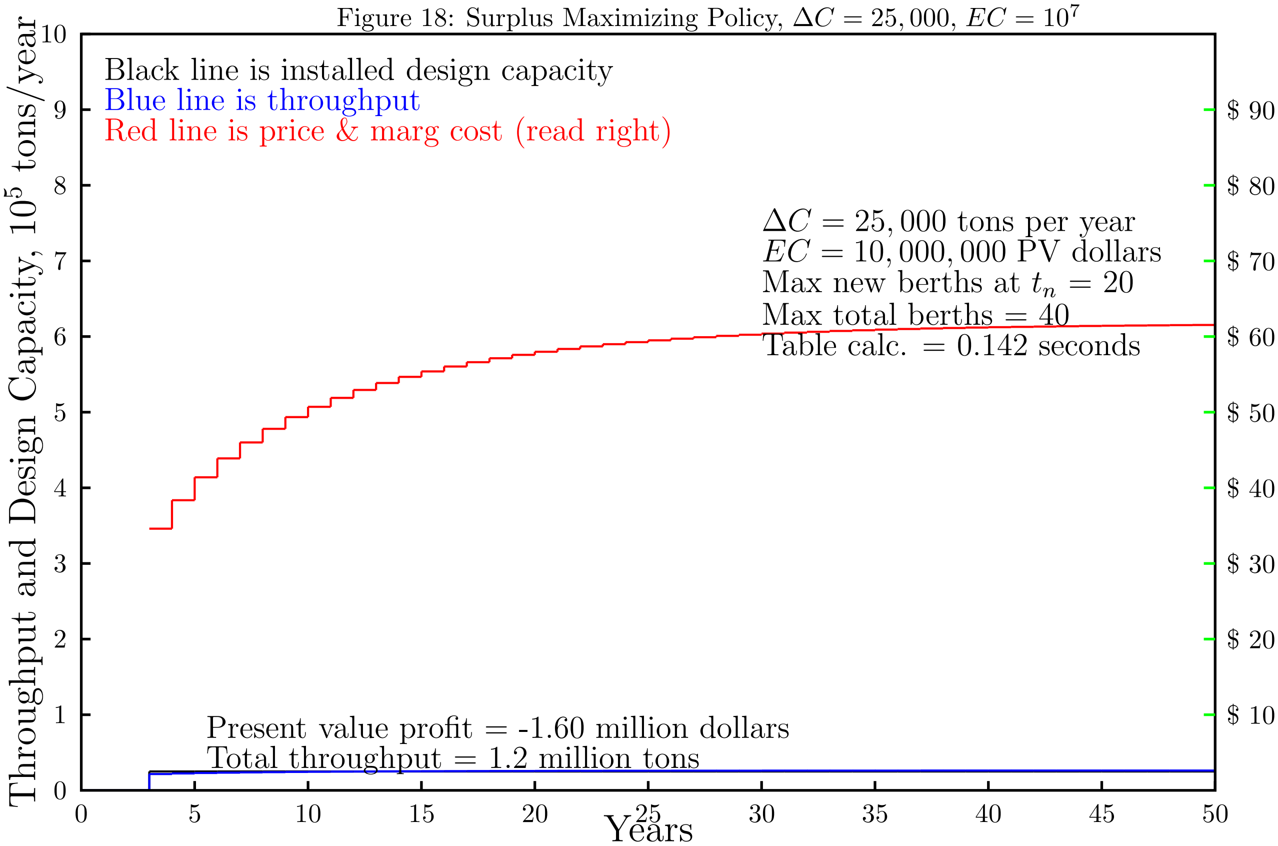


Figure 18: Surplus Maximizing Policy, $\Delta C = 25,000$, $EC = 10^7$



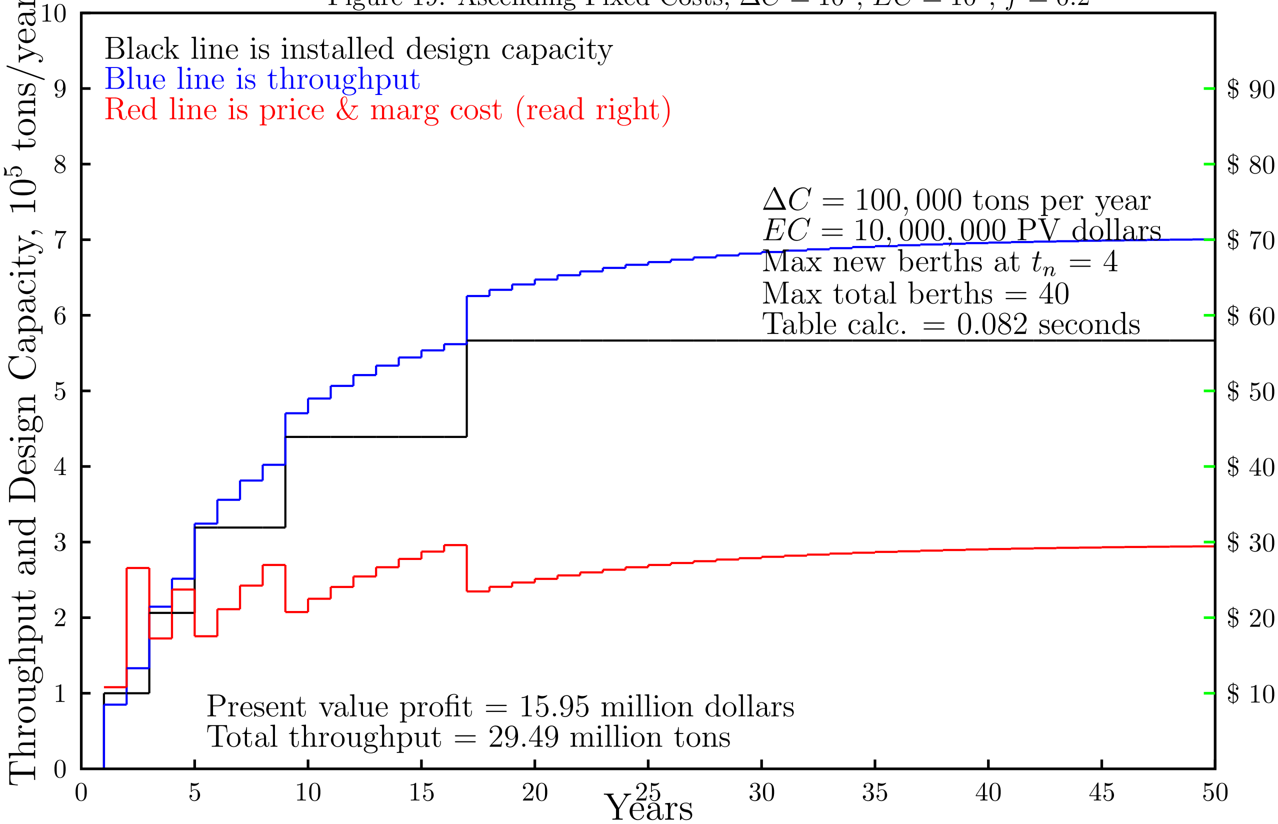
6 Ascending (and Decending) Fixed Costs

In the real world, even if the berths are identical operationally, the fixed cost of a berth will usually increase with the number of berths. The early berths will get the best locations. Each additional berth will require more dredging or fill than the last. For an example of such a problem, we will use the simplest port, and the linear demand surface, but assume that the cost of each berth is $(1.0 + f)$ times as large as the last. The recursion becomes

$$V_n(i) = \max_k \left\{ S^*(i, t_n) - \left(\sum_{\kappa=1}^{k} (1.0 + f)^{i+\kappa-1} EC \right) + \rho V_{n+1}(i + k) \right. \qquad (11)$$

where the sum term is zero if $k = 0$. Figure 19 shows the results for $f = 0.2$ but otherwise the same problem as Figure 8.

Figure 19: Ascending Fixed Costs, $\Delta C = 10^5$, $EC = 10^7$, $f = 0.2$



The first thing to notice about Figure 19 in that the step sizes are increasing. Under our escalating fixed cost assumption, The design capacity (min average cost throughput) of the ith berth is $(1.0 + f)^{i/3}\Delta C$.

The second thing to notice is that the societal surplus maximizing port makes a substantial profit in this situation. Table 2 shows that the higher the fixed cost escalator, f, is the more money the economically efficient port makes. Remember this port always charges marginal cost.

Table 2: Port and Last Berth NPV (million USD) as fixed costs escalate

f	Port NPV	Last Berth NPV
0.2	15.95	0.054
0.1	9.27	0.025
0.0	-1.22	-0.043

What's happening here is the algorithm will not build the last berth unless it has net positive surplus. But in order for that berth to have net positive surplus that berth must (nearly) break even with the port charging marginal cost. But if the last, most expensive port is (nearly) breaking even, then all the other berths are making money.

It is almost uncanny how the surplus maximizing port apes a competitive market. This is exactly what would happen in a competitive berth provision market.[9] The early berths get the best and cheapest locations. The later berths are further up the river. But they all act as price takers. Where will the price go at equilibrium? It will go to the marginal cost of the marginal berth. But the marginal berth will just break-even in present value terms. If it were to make money, it would pay somebody to build another berth, and it would no longer be the marginal berth. If it were to lose money, it would go out of business and would not be the marginal berth. For the marginal berth, marginal cost and average cost are the same.

We can see the same thing in the rightmost column of Table 2, which shows the NPV of the last berth installed. The NPV of the last berth is a small fraction of the cost of the berth, increasing a bit with the escalator, both because the cost of that berth is increasing, and because the number of berths being installed is decreasing, exacerbating the discreteness of the problem.

Misleading. NPV of last berth is small in large part because its looking at 0.9**16 = 0.19 discount factor.

By exactly the same argument, if fixed costs are decending — each berth costs less than the previous berth — then the surplus maximizing port will lose money as Table 3 shows. In this table, we have run a negative fixed cost escalator, so the last berth is cheaper than the earlier ones. The port still (nearly) breaks-event on the last berth, but loses money on the older ones.

Table 3: Port and Last Berth NPV (million USD) for descending fixed costs

f	Port NPV	Last Berth NPV
0.0	-1.22	0.072
-0.1	-12.71	-0.043

It is hard to imagine a port topography where the second berth would be cheaper than the first; but descending fixed costs can happen for other reasons. The most obvious one is improved technology. Given descending fixed costs, the surplus maximizing port will require a subsidy. But this is merely a signal that the older berths should be upgraded or, more likely, turned into condos. Unfortunately, in most real world ports, this signal was misread. The old berths could no longer make money, so they were turned over to a public monopoly which could charge monopolistic prices, thereby attempting to preserve longshoreman jobs, etc. The result was over-supply; and the port could rightly claim, if we charge marginal costs, we will lose money.

The answer should not have been: raise prices. The answer should have been: reduce supply until you can break-even charging marginal costs, which is exactly what competition would have done.

[9] And did come close to happening in many ports in the early break bulk days when a big port could have more than hundred privately owned berths.

6.1 Descending (Ascending) Variable Costs

Although it is difficult to imagine a real world situation in which the fixed costs of a berth decrease with increasing number of berths, it could happen with variable costs due to advances in technology. To study changing variable costs, we begin as usual with the simplest situation, everything the same as Section 3, except that now each berth j can have its own quadratic marginal cost curve, $K_j x_j^2$ where x_j is the throughput through berth j in the nth period. This leads to the to the following set of equations for the short run situation at t_n given i installed berths.

$$\alpha/\beta - 1.0/(\beta(1-\gamma t_n))\sum_{j=1}^{i} x_i = K_j x_j^2 \qquad j = 1, 2, \ldots, i \tag{12}$$

This set of equations merely says price must equal marginal cost for each berth. Since the left hand (demand) side is the same for all i equations, the solution takes the following very simple form.

$$k_1 x_1^2 + 1.0/(\beta(1-\gamma t_n))\left(\sum_{j=1}^{i}\sqrt{k_1/k_j}\right) x_1 - \alpha/\beta = 0 \tag{13}$$

$$x_j = \sqrt{k_1/k_j}\, x_1 \qquad j = 2, 3, \ldots, i \tag{14}$$

Equation 13 can be solved for x_1 by the quadratic formula, after which all the other throughputs follow from Equations 14.

Figure 20 shows the results for $EC = 10^7$, $\rho = 0.9$, $DC = 100,000$, $k_1 = 1.5EC(1.0-\rho)/\Delta C^3$ and $k_j = k_1/j$. The last assumption is contrived; but allows us to produce a problem with descending variable costs without increasing the size of the state space. Under this strange assumption, the design capacity of the ith berth is $\Delta C i^{0.333}$. In Figure 20 each step in design capacity is larger than the last.

In this situation, the surplus maximizing port operates mostly below design capacity. Everytime the throughput climbs up to design capacity, more capacity is ordered. One result is that the port loses money. The port will (nearly) break-even on the last berth, which means it will lose money on the earlier ones. This is a signal that the earlier berths should be upgraded or released to other uses.

Why is this optimal??

Figure 21 shows the results for the opposite situation in which the variable cost of the ith berth is proportional to i. Under this assumption, the design capacity of the ith berth is $\Delta C/i^{0.333}$. Ascending variable costs could easily happen in the real world; for example, in situation where the more unfavorable location of the last berth implies additional cargo shifting or in-land transportation costs. In this situation, the surplus maximizing port tend to operate above design capacity, delaying new expansions until even the last berth is making money.

Why is this optimal??

It should be pointed out that it is unlikely in the real world we would find such drastically descending/ascending variable costs as studied in Figures 20 and 21.

It should also be obvious that we are using the terms *ascending* and *descending* to mean something quite different than the normal use of *increasing* and *decreasing* when talking about costs. Usually when an author talks about increasing or decreasing costs, he is talking about moving along a single cost curve. As we have seen, this sort of movement loses importance when the problem is viewed dynamically. What is important is whether the fixed cost or variable cost curve for the next investment is higher or lower than the last. The concept of ascending/decending is comparing one cost curve with another.

Figure 20: Descending Variable Costs, $\Delta C = 10^5$, $EC = 10^7$

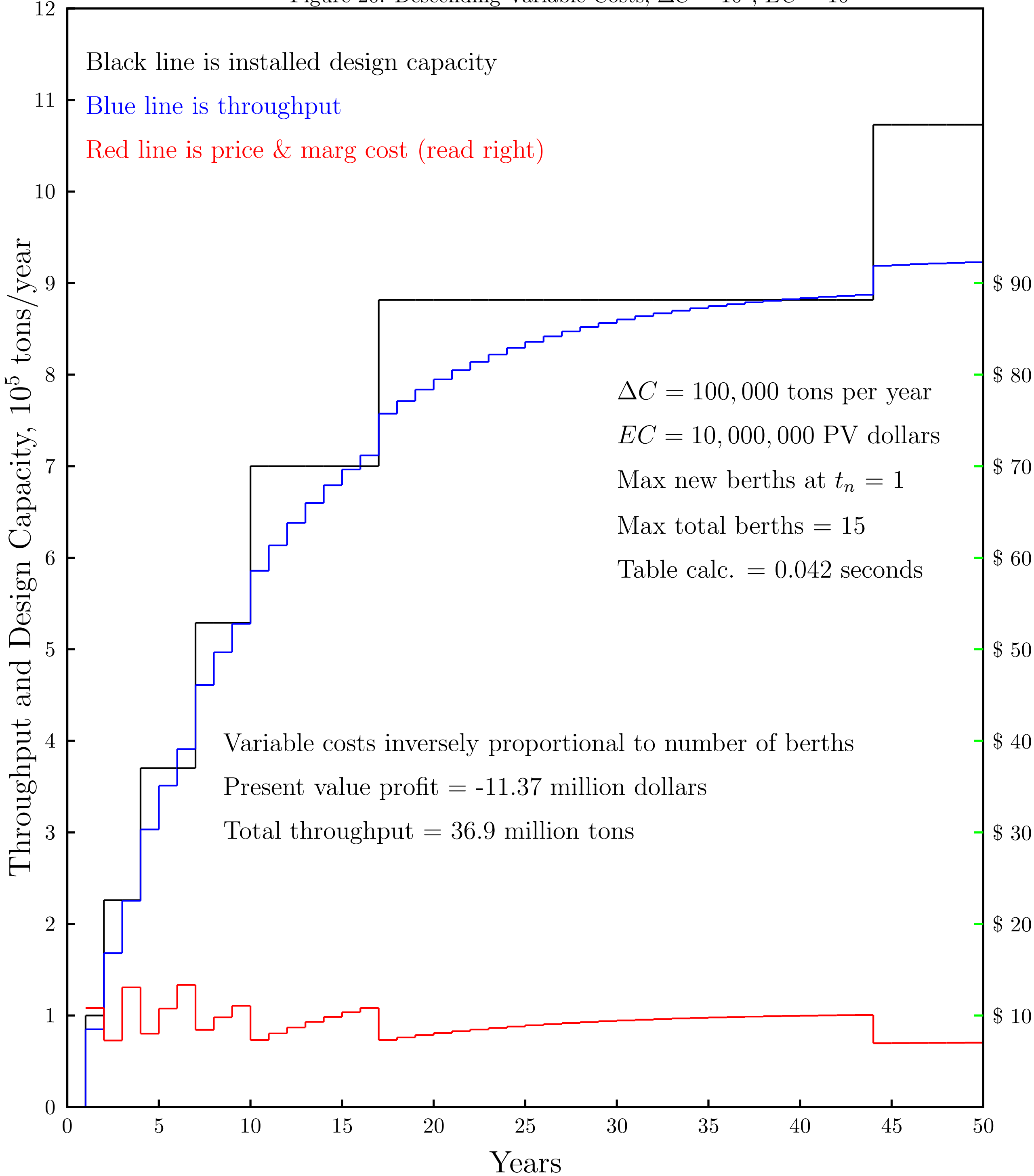
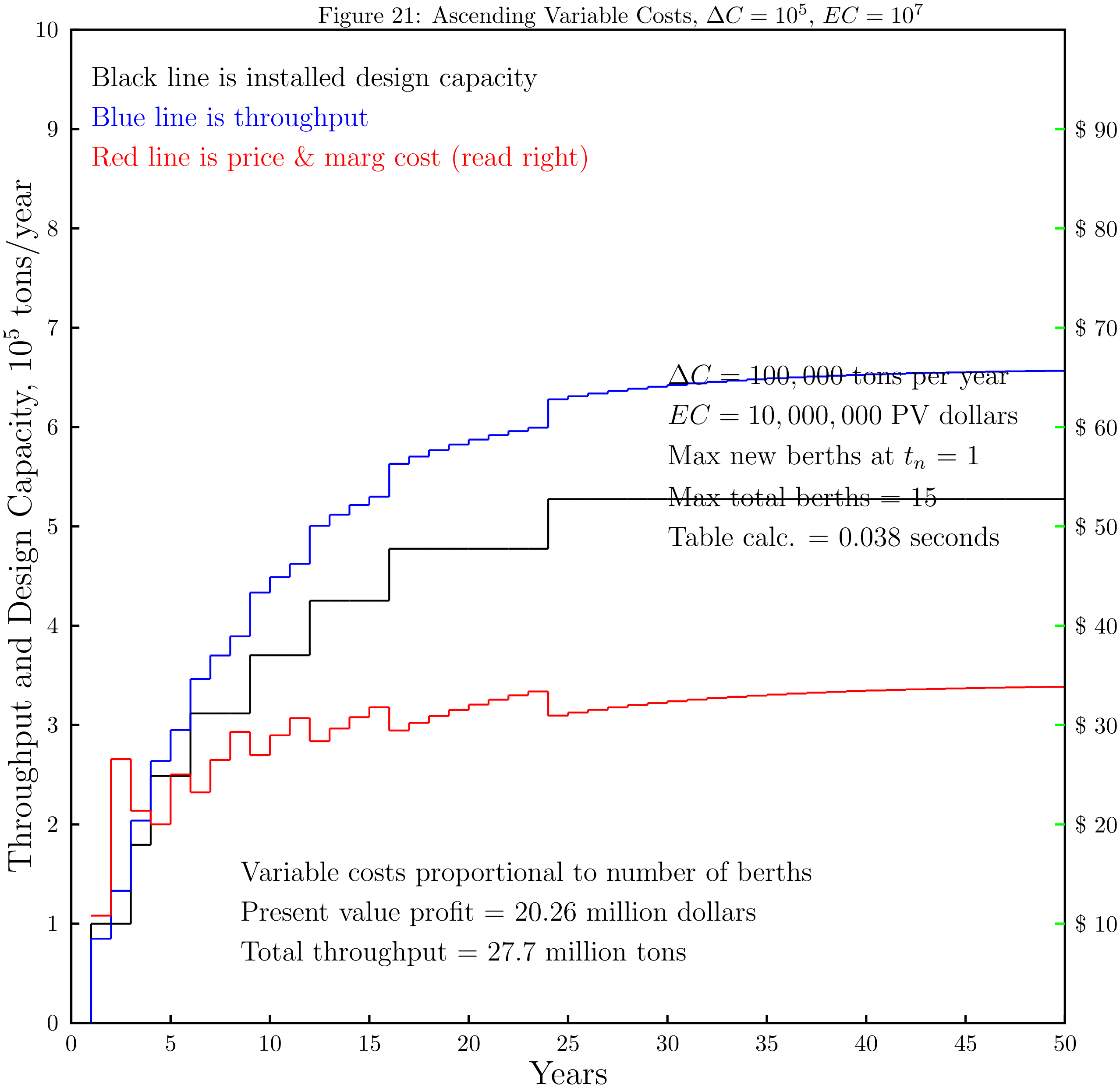


Figure 21: Ascending Variable Costs, $\Delta C = 10^5$, $EC = 10^7$



7 Generalizing the Simple DandT Port Model

7.1 The Most General Model

The DandT algorithm is actually more general than Devanney and Tan indicated; but its most fundamental limit, the requirement than all possible investments have the same variable costs, is very restrictive. The surplus maximizing algorithm easily generalizes to all sorts of cases which the DandT algorithm can't handle. In generalizing our model, we can have two quite different, if related, goals:.

A Showing that there is no conflict between marginal cost pricing and long-run capital efficiency. If this is all we are trying to do, as long as we can write down the algorithm for determining the optimal expansion policy, we are done.[10] We don't actually have to compute it. This is in the spirit of the mid-20th century economists who studied competitive market equilibria with sets of equations they had no hope of actually solving, but for which they knew or could show a solution exists.

B Actually determining the optimal policy for a real world port. In this case, the algorithm must actually be computable in a reasonable time with a reasonable amount of computer resources.

In this section, we try to go as general as possible. Examining the core of the surplus maximizing algorithm, without any concern about actual computability, we need only the following:

1. At each decision point, t_n, we can characterize the the current state of the port $\vec{i}$ where $\vec{i}$ is a vector describing the number of berths of each of K types or L ages in operation, the number of cranes on each berth, the number berths/cranes on order, etc, etc.. Let $\mathbf{I}$ be the state space; the set of all possible states the port could get itself into. $\mathbf{I}$ must be finite.
2. The port handles M commodities (offers M services). Let $\vec{x}(\vec{i}, t_n) = x_1, x_2, \ldots, x_M$ be the throughout in the nth period given $\vec{i}$, and let $\vec{p}(\vec{i}, t_n)$ be the corresponding vector of prices. Let $vc(\vec{x}, \vec{i}, t_n)$ be the variable costs of handling $\vec{x}$ given $\vec{i}$ in time period $(t_n, t_n + 1)$. The partial derivatives of vc with respect x_m, $mc_m(\vec{x}, \vec{i}, t_n)$, are known (or can be estimated) and are increasing.
3. The demand, $D_m(\vec{p}, t_n)$ for service m in period n is known, non-increasing in p_m, and the integral of this surface with respect to p_m is finite, which by 4.3 is always possible.
4. At each decision point, t_n, the short run situation (port state and demand) between t_n and the next decision point t_{n+1} is fixed or can regarded to be so.

   With these assumptions, we can figure out the short-run surplus by first solving the set of equations

$$D_m^{-1}(\vec{x}^*, t_n) = mc_m(\vec{x}^*, \vec{i}, t_n) \qquad m = 1, 2, \ldots, M \tag{15}$$

   for the throughputs, $\vec{x}^*$ that will result from marginal cost pricing and the corresponding prices $\vec{p}^*$. The resulting short-run surplus is

$$S^*(\vec{i}, t_n) = \sum_{m=1}^{M} \int_0^{x_m^*} D_m(\vec{p}^*, t_n) dx_m - vc(\vec{x}^*, \vec{i}, t_n) \tag{16}$$

   .

   $vc(\vec{x}, \vec{i}, t_n)$ can depend on $\vec{x}$ in any way it wants as long as the marginal costs are positive and non-decreasing. Whether some of these expenses are "joint" is totally irrelevent. The pricing will be determined by the partials. ***For a surplus maximizing port, there is no more need to allocate "joint" costs across services than there is to allocate fixed costs across time periods.***
5. At each decision point, we have a finite number of possible investment decisions. For each such investment decision, $\vec{z}$, where $\vec{z}$ can be a vector, there is a transformation from the state $\vec{x}$ at t_n to the state at $t_n + 1$, $T(\vec{i}, \vec{z}, t_n)$ such that $T(\vec{i}, \vec{z}, t_n)$ is an element of $\mathbf{I}$. Associated with each such possible investment decision, $\vec{z}$ is a set of fixed expenses whose present value is $EC(\vec{z}, \vec{i}, t_n)$.
6. We can specify a reasonable boundary condition at some time in the possibly distant future, t_N, which allows us to compute the present valued surplus of the port from t_N on.[11]

Given the first five requirements, we can develop a recursion relationship determining the port's optimal expansion policy.

$$V_n(\vec{i}) = \max_{\vec{z}} \left\{ S^*(\vec{i}, t_n) - EC(\vec{z}, \vec{i}, t_n) + \rho V_{n+1}(T(\vec{i}, \vec{z}, t_n)) \right. \tag{17}$$

where the max is over all possible $\vec{z}$ at t_n given $\vec{i}$.

[10] More precisely, construct a computer program that would implement the algorithm given enough computational resources and time.

[11] By moving t_N far enough into the future, any reasonable discount rate will result in a policy that is only weakly dependent on the boundary condition. Thus, if we set t_N to say, 200 years, a possible boundary condition is simply $V_N(\vec{i}) = 0$.

Given the sixth requirement, we can *in theory* compute the optimal value function table by backwards recursion and determine the optimal expansion policy (and its corresponding prices) by moving forward through this table.

This is a very general algorithm. With the exception of the known demand surface, it could model a rather realistic port. Of course, at this level of generality, there is little hope of actually computing the optimal value function. The size of the state space, $\mathbf{I}$, will increase combinatorially with each additional component of the state vector $\vec{i}$. This will very quickly get out of hand if the dimensionality of $\vec{i}$ is much more than 6 or 7.

But if all we are doing is making the point that marginal cost pricing does not imply subsidies in the face of large indivisible fixed investments, we don't care. The fact that the algorithm is theoretically computable and that the problem parameters allow the port an expansion and pricing policy which will balance periods of over-supply and under-supply, and non-decreasing fixed costs are all we need. proof? not proven?

7.2 The Multi-Port Problem

An important problem is buried in Equations 15, 16, and 17: the multi-port problem. Consider a public entity that is responsible for two container ports: A and B, serving over-lapping hinterlands. Obviously, the demand that Port A sees will depend on the price Port B charges and vice versa.[12] But from the point of view of the public entity, Ports A and B are really just portions of a single super-port and its job is — or at least should be — to price and expand both ports in a manner that maximizes societal income. If we can actually compute Equation 17 for an M of two or more, we can address the multi-port problem.

[12] Sometimes called "kinked" demand curves referring to the loss of business if A charges more than B and vice versa.

8 Specific Varients on Simple DandT Port

In this section, we consider a few specific varients on the DandT port and comment on their computational feasibility. These varients have been chosen mainly to suggest possible ways the approach can be taken.

8.1 Construction period longer than decision period

Real world berth construction could take as long as 5 years, much longer than the period for which the the port could reasonably act as if demand were constant. Fortunately, this is not a tough problem as least for the single commodity port. There are two approaches.

1. Allow an expansion decision every L "demand mini-periods". Demand is allowed to shift at the end of each mini-period.[13] For example, we might set the length of the "demand mini-period" to say three months and allow an expansion decision only every 12 months. This complicates the calculation of the short-run surplus in decision period (t_n, t_{n+1}) slightly, for now we must figure out the price and throughput for four different mini-periods for each combination of t_n and i. However, the size of the state space stays the same and the overall effect on computation time will be less than linear in the number of mini-periods in each decision period. Linear increases in computational effort are rarely a problem for today's computers.
2. Or we can allow a construction lag that is J decision periods long. If at each decision period we can order at most K berths. The resulting recursion relation is

$$V_n(i, i_1, i_2, \ldots, i_{J-1}) = \max_k \{S^*(i, t_n) - EC(k, i, t_n) + \rho V_{n+1}(i + i_1, i_2, \ldots, i_{J-2}, k)$$

   where max is over $k = 0, 1, \ldots, K$ and i_j is the number of berths which will be on-line j decision periods from now, that is the number of berths ordered $J - j$ periods ago. This increases the size of the state space by $(K+1)^{J-1}$ but as long as K and J are small, this is not too bad. If the decision period is short, we can usually make $K = 1$ — can order at most one new berth each "short" period — in which case any $J < 10$ is no problem.

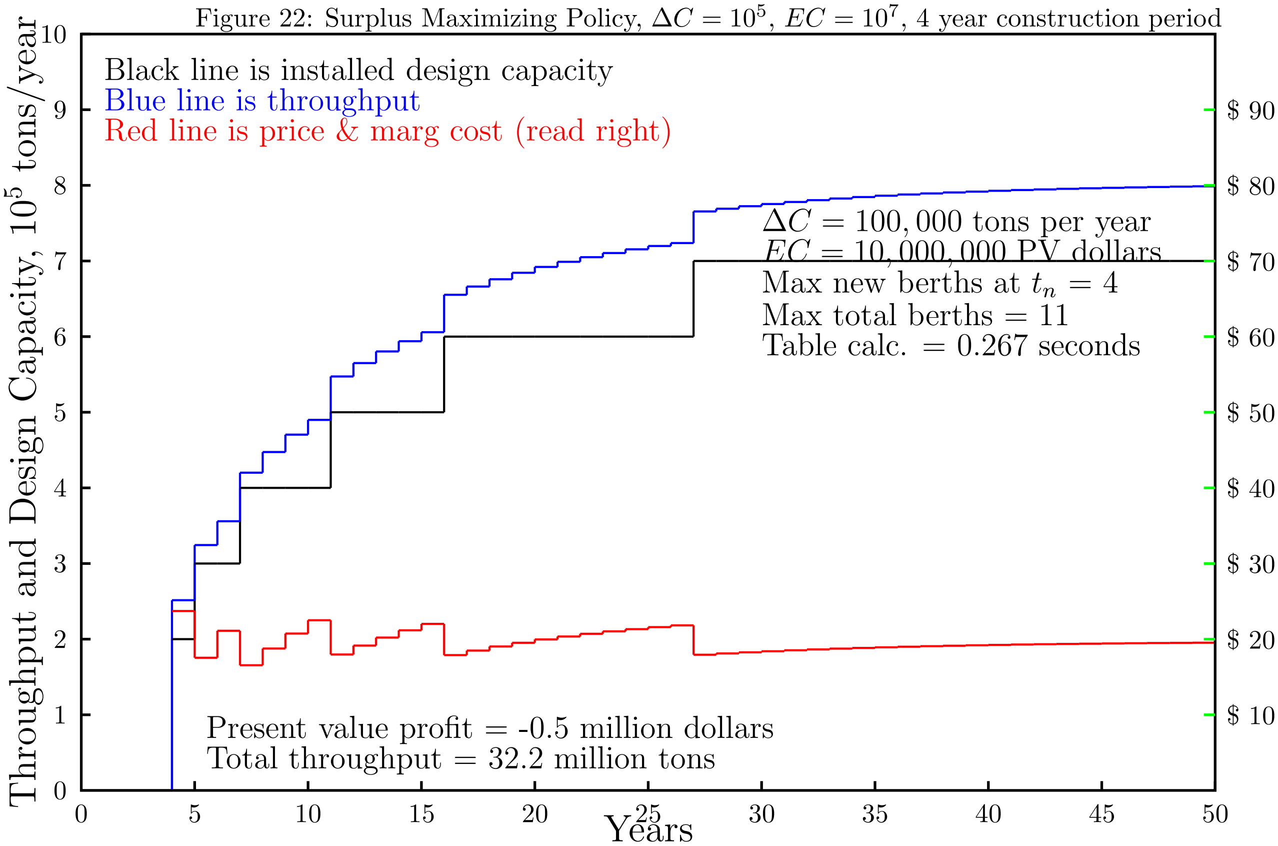
Figure 22 shows an example of approach 2 with $J = 4$ and $K = 4$. Everything is the same as in Figure 8 except a berth ordered at t_n comes on line at t_{n+4}, four years later, and we allow up to 4 berths to be ordered at once. Given the rather high discount rate, this three year increase in construction time is very roughly equivalent to a ρ^{-3} or 37% increase in fixed costs. Thus, societal surplus is maximized by operating a bit further up the demand surface. As usual, for level fixed costs, the surplus maximizing port nearly breaks even.

The time to calculate this optimal value table on a very ordinary desktop with totally unoptimized code is about 0.3 seconds. Clearly, we have a long way to go before we have to worry about computational feasibility.

Obviously, we could combine L demand mini-periods with a multi-decision period construction lag and in most reasonable real world cases still be feasibly computable.

[13] Demand can shift to the left from one mini-period to another as long as the long-run demand pattern is one of growth. Thus, the approach can handle seasonal/diurnal fluctuations in demand.

Figure 22: Surplus Maximizing Policy, $\Delta C = 10^5$, $EC = 10^7$, 4 year construction period



8.2 Different Services using the same fixed investment

For a socially efficient port, two services provided by the same berth are different ***only if their marginal cost curves are different***. For an efficient port, there are not nearly as many different "services" as a standard port tariff would have us believe. But it can happen. Handling containers with hazardous contents is a possibility. If handling such containers requires more resources or takes longer than handling a non-hazardous container, then this truly is a different service, with its own marginal cost curve. So let's consider a two service berth which handles both hazardous and non-hazardous (safe) containers. Let x_s and x_h be the throughput of safe and hazardous containers in some period. We need a variable cost surface $vc(x_s, x_h, i, t_n)$; and we need two different demand surfaces $D_s(p_s, t_n)$ and $D_h(p_h, t_n)$.

For our sample problem, we will assume identical berths operationally — which will allow us to divide the throughput evenly — and we will assume each berth has the following simple cubic variable cost surface

$$vc(x_s, x_h, i) = (K/3)(x_s/i + 2x_h/i)^3 \tag{18}$$

for which the two partials are

$$\frac{\delta}{\delta x_s} = K(x_s/i + 2x_h/i)^2 \tag{19}$$

$$\frac{\delta}{\delta x_h} = 2K(x_s/i + 2x_h/i)^2 \tag{20}$$

where once again we set $K = 1.5EC(1-\rho)/\Delta C^3$ so we can talk in terms of design capacity. It is easy to see that, for this simple surface, handling one hazardous unit is equivalent to handling two safe units. Perhaps it takes twice as long. We can also see that, whatever the throughputs are, the marginal cost of a hazardous unit is double that of a safe; and hence the efficient price of handling a hazardous unit will be twice that of handling a safe unit. But the key point for now is that the variable cost of handling a safe unit depends on how many hazardous units the port is handling and vice versa. We have joint variable costs.

For our sample safe container demand curves, we used DandT's linear surface, Figure 1. For our hazardous container demand curves, we simply increased both α and β by 50%.

To find the efficient throughputs, we need to solve the following set of equations for each combination of i and t_n.

$$D_s^{-1}(x_s, t_n) = \frac{\delta}{\delta x_s} vc(x_h, x_s, i, t_n) \tag{21}$$

$$D_h^{-1}(x_h, t_n) = \frac{\delta}{\delta x_h} vc(x_h, x_s, i, t_n) \tag{22}$$

For our extremely simple, example variable cost surface, we could use the equivalence of one hazardous unit to two safe units, to help us do this; but, in the interests of future generalization, we didn't. Rather the set of equations were solved for each combination of i and t_n by a simple implementation of the Newton-Raphson method.

The results for $EC = 10^7$ and $\Delta C = 100,000$ tons/year are shown in Figure 23. This problem involves much more demand than the same single service problem, Figure 8, so the port ends up providing 23 berths. If you double the hazardous throughput and then add in the safe throughput, you will see that the port keeps *equivalent* throughput very close to design throughput throughout. As expected, the efficient price for the hazardous cargo is always double than for the safe cargo. Finally, we note that overall the port (nearly) breaks-even. In short, nothing much is changed. ***In particular, the fact that the variable costs are joint poses absolutely no problem for the efficient port.***

This 2-service problem takes about 0.7 seconds to solve despite un-optimized code and very modest computational resources. This is pretty impressive given that Newton-Raphson requires us to solve a set of linear equations multiple times for each combination of state and stage. Obviously, considerably larger multi-service problems are well within the realm of computational feasibility. We expect computer times to go up roughly as M^3.

The efficiency of Newton-Raphson and similar methods depends critically on how close the initial guess is to the solution. In most cases, the solution at the last i for any t_n will be an excellent choice for the guess at the next i. If i is increasing, we can be sure the new x_s^* and x_h^*, will both be moderately larger than the old. If we save the solution for $i = 1$ at $n + 1$, then the solution for $i = 1$ at n will involve "slightly" smaller throughputs where the slightly will certainly be true for n greater than about 10. No advantage of this was taken in Figure 23. In fact, the same initial guess was used throughout and that guess was far from the solution in most cases.

Computational efficiency will also depend critically on how well-behaved the equations are. In particular, we need a solution in which all the x_m's are non-negative. In Figure 23, we picked demand surfaces for which this was the case.[14] A real world analyst won't have this freedom.

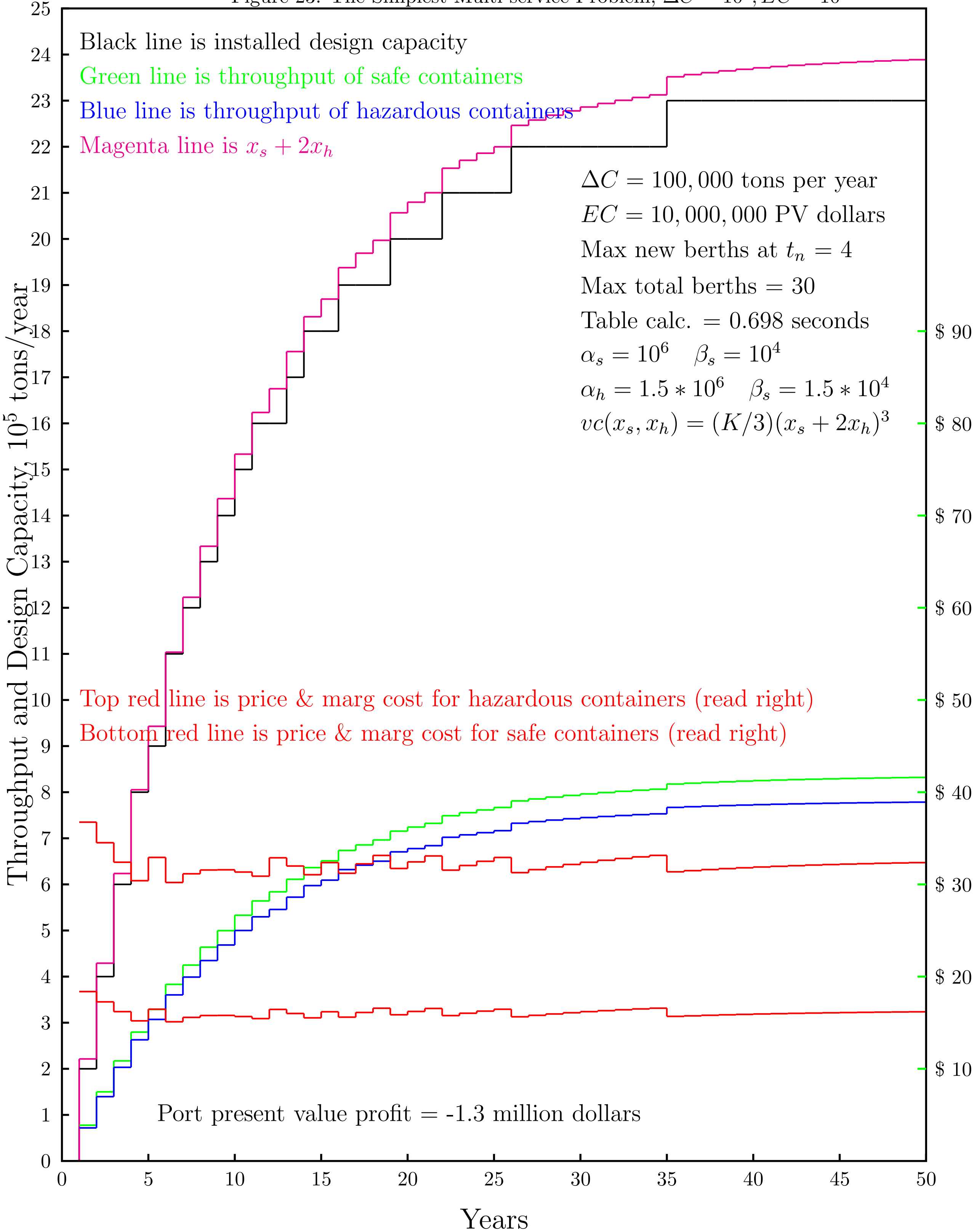
The short-run equilibrium for any combination of i and t_n is itself a constrained, optimization problem. For any given (i, t_n), the problem is to maximize the surplus in period n given all the constraints implied by i. In some cases, this short run problem can be solved very efficiently via linear programming. An elastic network (Elnet) is one such case, See Devanney 2007. In an Elnet, the demand and marginal cost curve for each service can depend only on the amount of that service provided. However, an Elnet can accept over-all constraints which can represent the state of the system. Elnets also have the capability of handling intra-port transportation costs within the short-run model.

The picture that is emerging is pretty clear. Dynamic programming allows us to couple a whole series of short-run equilibrium problems. The short run problems can be as solved any way that makes sense. The solutions to these short-run problems are then fed back to the dynamic program to determine the over-all equilibrium through time.

From this view point, the port can be regarded as a micro-economy. If we had enough computational resources, we could expand the port to represent an entire economy, and proceed in the same fashion.

[14] If the solution involves a negative throughput, this is probably a signal that the efficient short-run equilibrium involves providing zero units of this service. Currently, our algorithm simply replaces any negative throughputs with zero. But the author does not know under what conditions, if any, this ad-hoc procedure retains optimality.

Figure 23: The Simplest Multi-service Problem, $\Delta C = 10^5, EC = 10^7$



8.3 Two terminals competing for the same space

Modern ports tend to be a collection of specialized terminals. Consider a port that comprises a container terminal, A, and a dry bulk terminal, B. The terminals are essentially two separate ports operationally, but they compete for the same space. If a container berth is built at a particular location, then the next bulk carrier berth will have to go elsewhere at an increase in fixed costs. In this situation, the state of the port at t_n is described by the pair (i_A, i_B) where i_A/i_B is the number of container/bulk berths in operation at t_n. The recursion becomes

$$V_n(i_A, i_B) = \max_{k_A, k_B} \{S_A^*(i_A, t_n) + S_B^*(i_B, t_n) - EC(k_A, k_B, i_A, i_B, t_n) + \rho V_{n+1}(i_A + k_A, i_B + k_B)$$

where k_A and k_B are the number of new containers/dry bulk berths ordered at t_n. In this situation, these two investment decisions are coupled only by the fixed cost function.

For a simple example of this sort problem, consider a situation in which the cost of the next A type berth is $EC_A(1.0 + f)^{i_A + i_B}$ and the cost of the next B type berth is $EC_B(1.0 + f)^{i_A + i_B}$. where f is the fixed cost escalator. To make this work, we need a decision period short-enough so that at each decision point, the port can order only 1 container berth, or only 1 dry bulk berth, or nothing.[15] Otherwise, the cost of a berth depends on the order in which the expansion decisions are evaluated. For our sample container terminal, A, we used the linear demand surface of Section 4. And for the dry bulk terminal, B, we used the constant elasticity surface of Section 5. Figure 24 shows the results for $EC_A = 10^7$, $DC_A = 100,000$, $EC_B = 10^6$, $DC_B = 50,000$ and an escalator of 20% per berth.

There's nothing really new about Figure 24. Since the fixed costs are increasing for both terminals

1. The design capacities don't have any real meaning.
2. Both marginal cost pricing terminals make money.

The significance of Figure 24 is that is it our first baby step into handling different kind of berths. The computational issue is that the number of states increases combinatorially with the number of different kinds of berths. In this case, computation time is up an order of magnitude over the one-dimensional port; but at 3 seconds on an ordinary desktop, we still have a way to go before we face real computational problems.

In this regard, it is important to point out that dynamic programming parallelizes easily and efficiently. If we have a number of processors at our disposal, we can assign each processor a portion of the state space. At any stage, n, the processors have no need to communicate with each other. They only need access to the optimal value function computed at the last, $(n + 1)$st, stage, which could be stored in common memory. After all the processors have finished their computations at stage n, the results become the new common memory; and everybody moves on to stage $n - 1$. There is nil overhead associated with dividing the problem among multiple CPU's. If a problem justifies the use of a cluster, we can scale up nearly linearly. If a problem is really worth solving, there's a pretty good chance we can actually do the computation.[16]

[15] Using the techniques of Section 8.1, we can make the decision period as short as necessary if we are willing to pay the computational price.

[16] It has been estimated that we could compute problems in which the size of the state space is 10^{12} with a 100 CPU cluster over night. See Devanney 2009. In the port context, this is something like nine different kinds of berths.

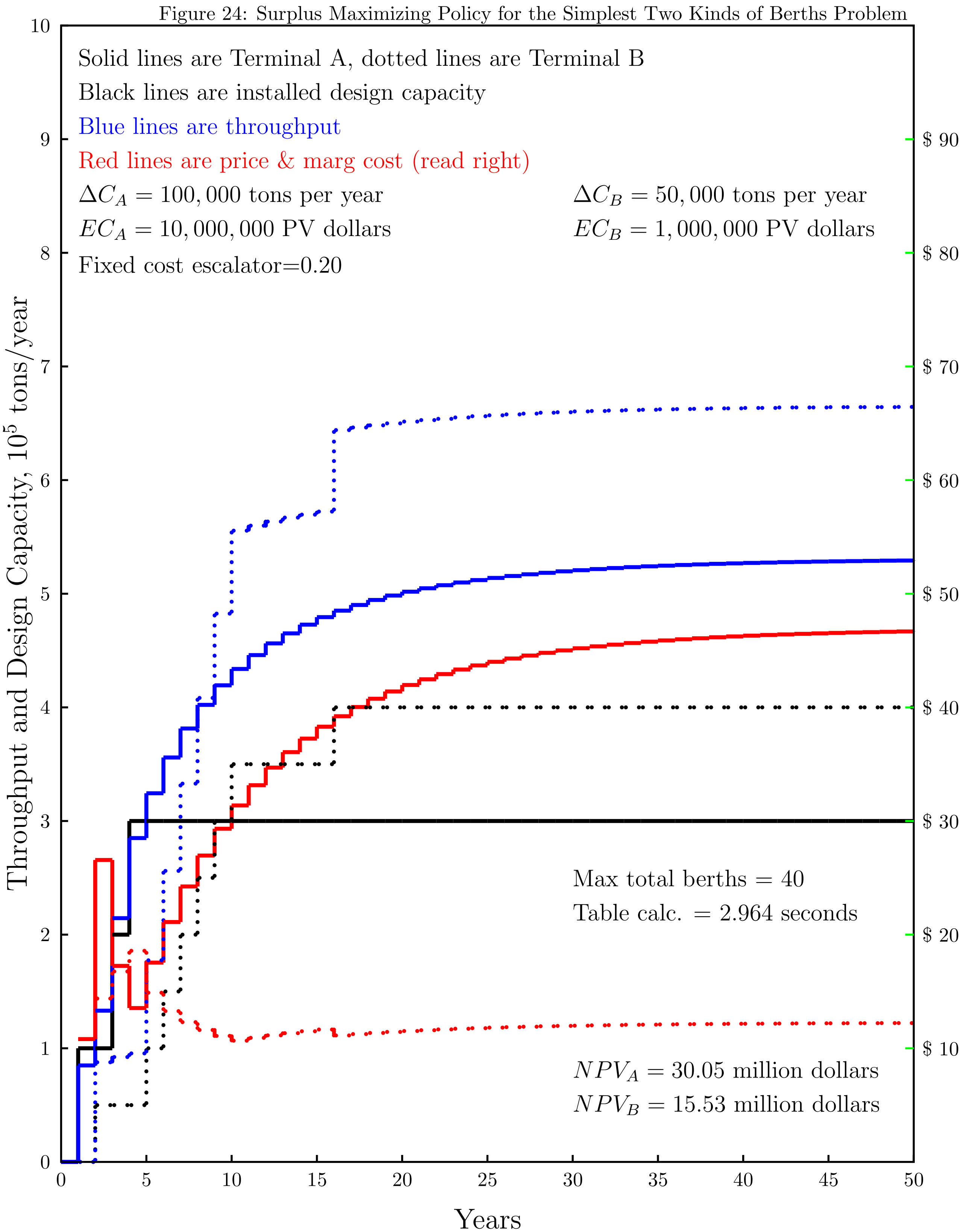


Figure 24: Surplus Maximizing Policy for the Simplest Two Kinds of Berths Problem
Solid lines are Terminal A, dotted lines are Terminal B
Black lines are installed design capacity
Blue lines are throughput
Red lines are price & marg cost (read right)
$\Delta C_A = 100,000$ tons per year
$\Delta C_B = 50,000$ tons per year
$EC_A = 10,000,000$ PV dollars
$EC_B = 1,000,000$ PV dollars
Fixed cost escalator=0.20
Max total berths = 40
Table calc. = 2.964 seconds
$NPV_A = 30.05$ million dollars
$NPV_B = 15.53$ million dollars
Throughput and Design Capacity, 10^5 tons/year
Years
0
1
2
3
4
5
6
7
8
9
10
0
5
10
15
20
25
30
35
40
45
50
$ 10
$ 20
$ 30
$ 40
$ 50
$ 60
$ 70
$ 80
$ 90

9 Handling Uncertainty

The goal of this section is to introduce uncertainty into the port pricing and expansion problem and demonstrate how dynamic programming allows us to handle this complication, provided the port is willing to maximize the expected value of the surplus. The main goal here is to show that uncertainty in no excuse for departing from marginal cost pricing.

The single biggest uncertainty that a port faces is on the demand side. Let's return to our simplest port whose linear demand surface is given by

$$x(p,t) = (1 - e^{-\gamma t})(\alpha - \beta p) \tag{23}$$

We focus on α keeping γ and β fixed. Suppose at decision point t_n, we observe a throughput x at price p. If γ and β are fixed, there is only one α that is consistent with that observation. That α determines our current demand curve.[17]

If we could somehow assign a probability density on α_{n+1} given α_n, $P(\alpha_{n+1}|\alpha_n)$, then we would be business, for the recursion becomes

$$V_n(i, \alpha_n) = \max_k S^*(i, \alpha_n, t_n) - EC(k, i, t_n) + \rho \sum_{\alpha_{n+1}} P(\alpha_{n+1}|\alpha_n) V_{n+1}(i + k, \alpha_{n+1}) \tag{24}$$

Dynamic programming is perhaps unique in its ability to incorporate uncertainty into a problem with almost no changes in the basic framework, largely because the approach requires us to figure out what we are going to do for any possible situation in which we could find ourselves. Under certainty, this means we expend a lot of computational effort on states which in hindsight are of no interest. But under uncertainty, figuring out what you are going to do for whatever happens is precisely what we need to do.

However, there are at least three things to notice about Equation 24.

1. We have had to increase the size of the state space. The demand curves are no longer a simple function of time, so we need to keep track of demand separately.
2. For computability we will need to work with discrete α's. For example, if our uncertainty is such that α could be anywhere between 0 tons and two million tons, we might chop the α space into say 20,000 ton increments, that is, treat any α between say 870,000 and 890,000 tons as if it were 880,000 tons. If we do this, then we will have 100 possible different α's at each combination of i and n, and we will have to take the expectation over 100 possible α's. Roughly speaking, the computational effort has gone up by 100^2 relative to the no uncertainty problem. This is not a show stopper for the single commodity port; but will severely limit the amount of generalizing we can do and still have a feasibly computable problem.
3. Much more fundamentally, we can no longer determine the port's entire future expansion policy at t_0. In fact at any time t_n, we can only determine our immediate investment decision, $k^*(i, \alpha)$ based on the current α. We will then have to wait until t_{n+1} before we can decide what to do at t_{n+1} based on whatever new α Nature decides to give us.

The next question is what do we use for $P(\alpha_{n+1}|\alpha_n)$, The simplest approach might be to assume α_{n+1} is distributed Normally with a mean of α_n and a variance σ^2.[18] This raises the question: whence came the σ^2? If we are not willing to specify a σ^2 from some distillation of past experience, then we will have to go Bayesian, specify a (presumably wishywashy) prior on σ^2 at t_0 and update that prior according to Bayes Rule as we observe the process unfold. This can be done using standard Bayesian techniques, but we will need to keep track of an additional state variable for each sufficient statistic.

Clearly, we are sharply limited in how much uncertainty we can introduce into the problem, and still have a feasibly computable algorithm. But the main point here is that uncertainty in no argument for departing from marginal cost pricing. In the short run, we still must allocate resources efficiently and that implies marginal cost pricing. What Equation 24 says is, under the same conditions as its certain counterpart, the marginal cost pricing, surplus maximizing port will ***on average*** break-even. Of course, the port could get surprised on the upside and find itself with less investment than *in hindsight* is optimal, in which case it will end up with a positive present value. Or the port could get surprised on the downside and find itself with more investment than *in hindsight* is optimal, in which case it will end up with a negative present value. But on average it will break-even.

[17] In this linear case, α is the demand at zero price, a somewhat nebulous concept in the real world, But we could have converted the observation, (x, p) at t_n, to the demand at any other price to specify the current demand curve.

[18] Of course in doing the calculations, we would have to lump the probabilities so that all the α's between, say, 870,000 and 890,000, end up at 880,000.

10 Future Work

This paper has barely scratched the surface of the kinds of port (and other transportation) pricing and expansion problems which can be solved by maximizing societal surplus using dynamic programming. An obvious next target would be the multi-port problem alluded to briefly in Section 7.2,

Another thrust would be to follow up on the suggestion in Section 8.2 and model the port's short-run situation for a given level of investment as an Elnet which can then be solved by linear programming. An even more obvious use of this approach is in applications where the short-run situation is a network, such as a road or rail network. Or for that matter a network of ports.

In either case, a key to pushing the computational feasibility of the approach will be the intelligent use of the short-run solutions from earlier combinations of stage and state to help us in solving the problem for the next combination of stage and state.

11 Conclusion

The paper shows it is possible to produce a Pareto-efficient port pricing and expansion policy by using dynamic programming to solve for the pricing and investment policy that maximizes present valued societal surplus. This policy simulates a a competitive equilibrium ***through time***. Preliminary computational results indicate that fairly interesting problems should be actually computable.

Moreover, these results show that, if a port follows an optimal expansion policy, it will not require a subsidy even though it charges marginal costs throughout provided only:

1. The port can get on the upside of the average cost curve ***for the smallest possible investment at full demand growth***.
2. Non-decending fixed costs. Each identical investment costs at least as much as the previous investment.
3. Non-decending variable costs. The variable cost curve for the next identical investment is no lower than that for the last. (The concept of ascending/descending costs has almost nothing to do with the normal use of increasing/decreasing in the context of costs. See Section 6.1.)

Provided only that (1) is true, the size of the individual investment is irrelevent.[19]

The first two requirements are almost never violated in the port context. The third requirement can usually be met by upgrading older berths as new technology becomes available. If a port loses money — or would lose money for an extended length of time — if it charges marginal costs, then society has over-invested in the port. Some of the port's resources should be released to other uses.

References

Allais, M.(1964), "Options in transport tariff policy", EEC Transport Series I, Brussels.

Devanney, J., Tan, L.,(1975), "The Relationship between Short-run Pricing and Investment Timing: The Port Pricing and Expansion Example", Transportation Research, Vol. 9, pp 329-337, http://www.c4tx.org/ctx/pub/port_price.pdf

Devanney, J., (2007), "Solving Transport Networks in which Supply and Demand are Elastic", http://www.c4tx.org/ctx/pub/elnet.pdf

Devanney, J., (2009), "Boiler Tube Pulling: An Example of Adaptive Maintenance", http://www.c4tx.org/ctx/pub/tube_pulling.pdf

[19] To be complete, we also need the standard requirements for short-run equilibrium: non-increasing short-run demand curves and non-decreasing short-run marginal cost curves.