The factory system: Some notes

Dan Friedman*

May 29, 1996

Abstract

We present a framework for constructing simulations of a production process.

1 Introduction

These notes are intended to provide a framework for constructing simulations based on Leijonhufvud (1996) and related writings[1]. The basic idea is that a more explicit, extensive-form representation of the production process will help elucidate the nature and consequences of increasing returns to scale.

The elementary unit of analysis is a *task*, denoted τ. Each task τ can be thought of as a an activity to which are assigned one or more workers and perhaps some machines. A task is indivisible in the sense that no sub-activity (of a particular process) receives the exlusive attention of any worker.

Each task has assigned to it an output label $z = \mathcal{O}(\tau)$ and the vector of material, capital and labor inputs $\mathcal{I}(\tau)$ required to produce a unit of z.

A production process (or technique) is an ordered collection of tasks $T = \{\tau_i\}$ with precedence relation $\tau_i \to \tau_j$, meaning that the output of $\mathcal{O}(\tau_i)$ of task i is required in positive quantity in task j, i.e. the $\mathcal{O}(\tau_i)$ component of $\mathcal{I}(\tau_j)$ is positive. One uses the binary relation $\to$ on T; in the usual way to define indirect predecessors and successors, initial and terminal tasks and connectedness.

Formally, T is a production process for z_i if T is finite and connected under $\to$ and if $z = \mathcal{O}(\tau)$ for some terminal node $\tau \in T$. Figure 1 sketches an example. Note that T may have cycles and several initial nodes, so it is not necessarily a tree.

*Department of Economics, University of California, Santa Cruz

[1] Leijonhufvud, Axel (1996), "Three Items for the Macroeconomic Agenda," University of Trento, manuscript prepared for the Université Nice-Sophia Antipolis Conference on "Macroeconomics: Past and Future", April 25-26, 1996. The relevant item is #2.

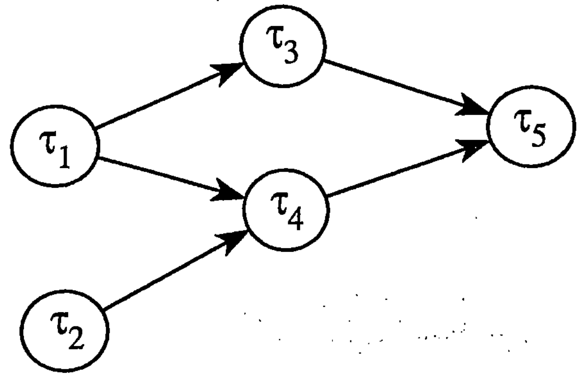

Figure 1: A production process. Here τ_1 and τ_2 are initial tasks and τ_5 is the final task of a production process for product $z = \mathcal{O}(\tau_5)$.

2 Basic Cost Functions

For a given process T with n tasks the material productivity and cost can be computed as follows. The input vecor $\mathcal{I}(\tau_i)$ can be written as (a_i, b_i) where $a_i = (a_{i1}, \ldots, a_{in})$ is the row vector of intermediate goods (output from tasks $\tau_j \in T$) required per unit of task i output and $b_i = (k_i, l_i, m_i)$ is the row vector of capital, labor and raw materials (not output of any $\tau_j \in T$). For the moment we assume l, k and m are scalar, i.e. only one kind of labor, capital and raw material.

Let $A = ((a_{ij}))$ be the matrix with rows $a_1, \ldots, a_n$. For an example like figure 1, the matrix has the form:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ + & 0 & 0 & 0 & 0 \\ + & + & 0 & 0 & 0 \\ 0 & 0 & + & + & 0 \end{pmatrix} \tag{1}$$

where 0's denote zero entries and + denote positive entries. Negative entries are ruled out by construction.

To produce output quantities of the tasks $y = (y_1, \ldots, y_n)'$ the direct "intermediate" input requirement is Ay, but the input requirement for Ay is A^2y, etc. Hence the total intermediate requirement is

$$x = Ay + A^2y + \ldots = \left(I + A + A^2 + \ldots\right) Ay = (I - A)^{-1} Ay \tag{2}$$

assuming that the series converges (i.e that $I - A$ is invertible). Hence to avoid infinitely costly production ~~we must~~ [it suffices to] have $1 > \|A\|$=modulus of largest eigenvalue of A. An interpretation is that as it is processed, any self-reproducing vector of throughput proportions $y/|y|$ (a unit eigenvector) increases in magnitude. Note that since $Ay = a'_n$ is the column vector (the $'$ denotes transpose) of direct intermediate input requirements, the total input requirement per unit of output z is $(I-A)^{-1}a'_n$

The ultimate requirements $R_T = (K_T, L_T, M_T)'$ of non-intermediate inputs per unit of final output can be calculated from the matrix $B = ((b_i))$ as

$$R_T = (I-A)^{-1}a'_n B \tag{3}$$

Given factor prices $r = (r_K, r_L, r_M)$ for these non-intermediate ("raw") inputs, the marginal production cost = AVC is $c_T = rR_T > 0$. There are also fixed costs F_τ associated with each task τ and a fixed cost F_P of coordination associated with the process T as a whole so

$$F_T = F_P + \sum_{\tau \in T} F_\tau \tag{4}$$

is total fixed cost. Thus, if firms are price takers in the factor market (as we shall always assume) and if they can freely adjust all quantities (as we shall assume for the moment) then the cost of producing output quantity $q > 0$ is

$$C_T(q) = qc_T + F_T \tag{5}$$

At any point in time there is a non empty technology set $\mathcal{T}$ of processes available for producing product z. Then the cost function is the lower envelope of equation 5 over $T \in \mathcal{T}$ i.e.

$$C_{\mathcal{T}}(q) = \inf_{T \in \mathcal{T}} \{qc_T + F_T\} \tag{6}$$

See figure 2 for an illustration. Note that we have implicitly assumed away capacity constraints. Of course, only the processes with high fixed cost and low marginal cost will be chosen at high output levels and these processes typically will have the greatest capacity.

3 Integer constraints

The cost function in equation 5 above assumes that workers and machines are completely divisible and homogeneous, and can be freely allocated across tasks. In actuality even if *ex-ante* homogeneous, both capital and labor *ex-post* become quite task-specific. Even *ex-ante* it is often necessary to avoid dividing workers between tasks. In that case we have an assignment problem for any given process $P = \langle T, \rightarrow \rangle$ with tasks $T = (\tau_1, \ldots, \tau_n)$ and output rate $q > 0$.

$$\min_{L \in \mathcal{N}^n} L_T = \sum_{i=1}^{n} L_i \quad \text{s.t. } L_i \geq ql_i \tag{7}$$

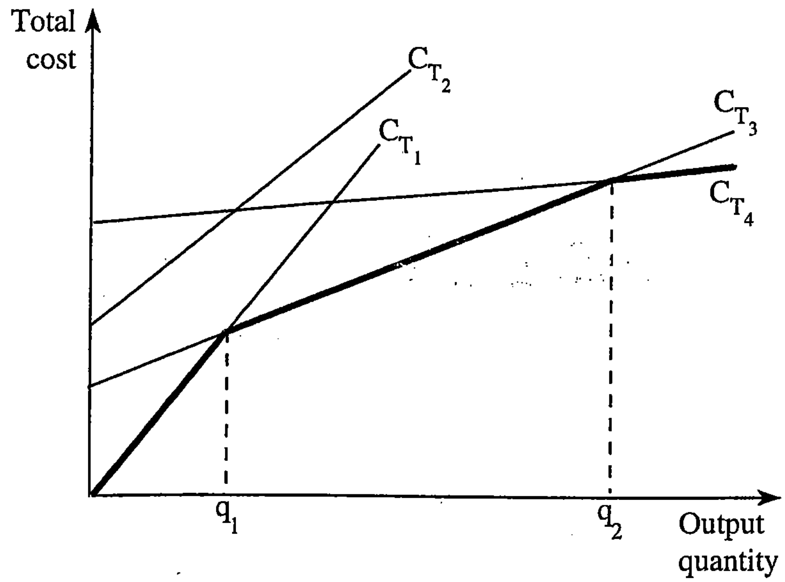

Figure 2: For the technology $\mathcal{T} = \{T_1, T_2, T_3, T_4\}$ the cost function $C_{\mathcal{T}}$ coincides with the cost function C_{T_1} (zero fixed cost) process T_1 when output quantity q is less than q_1 but with C_{T_4} when $q > q_2$. For intermediate output levels, the cost comes from the process T_3. The process T_2 is dominated and never used.

Here $\mathcal{N}$ represents the positive integers and $\mathcal{N}^n$ the positive integer lattice points in n dimensions. Note that output rate q is feasible only if task i has labor input at least ql_i. The solution to the integer programming problem in equation 7 is $L_T^*(q) = \sum_{i=1}^n L_i^*(q)$, where $L_i^*(q)$ typically is the smallest integer that exceeds ql_i. The difference $s_i(q) = L_i^*(q) - ql_i$, is the slack.

If L is optimized *ex-ante* for technology $\mathcal{T}$ and output level q and output is expanded to $q' > q$, then the new solution to equation 7 simply adds workers to some of the tasks. Likewise for contracting output for a given process T the cost minimizing firm simply lays off workers when the slack $s_i(q) \geq 1$. Figure 3 gives an example. The *ex-post* difficulty arises when changing the process, eg. crossing q_1 or q_2 in figure 2. Then the firm would want to reassign some workers to new tasks, which is costly, perhaps prohibitively so.

Integer constraints on machines could be imposed in an entirely similar fashion. But it may make more sense to think of rendesigning machines so that 1 or 2 are ever needed. E.g. if output rate $q = 100q_1$ and output rate q_1 uses 1 specialized machine, then don't use 100 of the same machines at q, but rather replace them by 1 or 2 more specialized, high volume machines. Similar considerations apply to labor. But formally we are talking about new processes, since the task is changed. We now consider the generation of new tasks and processes more generally.

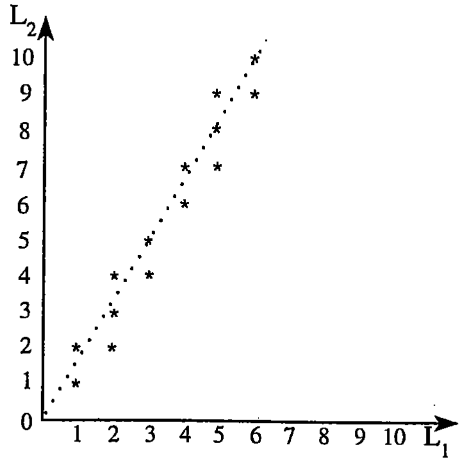

Figure 3: The * denote the scale expansion path (SEP) for a two task process with l_1=0.3 and l_2 = 0.5. The dashed line denotes the SEP in the absence of integer constraints. The rule for the integer constrained SEP is to increment L_i by 1 as q increases whenever the *slack* $s_i(q) = L_i - ql_i$ falls to 0.

4 New tasks and processes

So far the analysis has assumed a prespecified technology $\mathcal{T} = \{T_1, \ldots, T_m\}$. Perhaps the most important dynamic feature of production processes is that there is innovation. Tasks can be subdivided, lowering net input requirements while increasing the range of intermediate goods (cf. Romer and followers) and the fixed (or set-up) costs. The basic sort of innovation (a point mutation, if you like) is to take a task τ with an intermediate input requirement vector, a_τ (say, components $i_1, \ldots, i_k$ positive) and raw input requirement b_τ, and replace it by a sub-process T_τ that produces the same output z. If marginal production costs for T_τ (see equation 3 above) exceeds that for τ, the innovation is rejected. If it is less the innovation is potentially cost saving (recall it will have higher fixed costs) and so enters the technology. Of course it may still be dominated.

A possible algorithm for innovations is now sketched. No doubt experience will strengthen and streamline it considerably.

1. Pick a task τ used in the current process. A random device that chooses τ by its share of marginal cost seems appropriate.

2. Pick a random integer ≥ 1. This is the number of tasks in the subprocess that will replace τ[2]. Let us assume that this integer is 2, and the new subprocess is $\sigma \rightarrow \nu$.

[2]The case that it is 1 is important, as argued at the end of last section

3. The output of ν is the same as of τ, and the inputs include the output of σ and (with small probability) of other tasks in the old process. The output of σ is used only in ν (until possibly later innovations call on it). It's inputs are the same as for τ, plus (or minus) other tasks with small probability.

4. The material requirements for σ and ν are determined as follows. Let $e, e' \sim$ i.i.d uniform $[0, 1]$. Set

   (a) $l = \left(e + \frac{1}{2}\right) l_\tau$, $l_\sigma = e'l$ and $l_\nu = (1 - e')\, l$

   (b) Similarly generate k_σ, k_ν and m_σ, m_ν.

   (c) For intermediate good requirements, use the a_τs as in item 3 above. We will consider random perturbations at a later stage.

5. Check that the new sub-process has lower marginal cost than τ, using equation 3, at current factor prices. If not, put it on the shelf. If so, check whether it is lowest cost at some (typically higher) output rate, using equations 5-6. If not, put it on the shelf.

5 Excercises

Extent of market Expand demand and track performance variables such as MC, AC, number of tasks employed, cost of slack.

Shock treatments Change factor prices or temporarily reduce demand, and track the same performance variables.

Other Compare asymptotic cost, etc. older simulation to Dixit-Stiglitz, "learning curve" etc.