System Analysis, Design, and Control: Unification and Decomposition

Stanley B. Gershwin
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139-4307
gershwin@mit.edu
http://web.mit.edu/manuf-sys/www

1 Introduction

The purpose of this paper is to propose a unification of much research in manufacturing systems. It brings together manufacturing system performance evaluation and scheduling, and has, as a goal, the development of rapid numerical methods for system analysis and synthesis. This unification promises to eventually provide practical methods for operating policy and system design. An extensive research program is described for bringing it about.[1]

2 Real-time Manufacturing Systems Control

2.1 Problem Formulation

A problem formulation for a tandem system is presented in detail in Gershwin (1997). A reentrant version is described here.

Consider a k-machine production system that produces ℓ part types. Part Type q follows a sequence of $K(q)$ steps. It visits Machines $M_{S(1,q)}$, $M_{S(2,q)}$, ..., $M_{S(K(q),q)}$ in sequence, where $S(s,q) \in \{1,...,k\}$. After receiving its sth operation at Machine $M_{S(s,q)}$, the part goes to Buffer B_{sq}, and then to Machine $M_{S(s+1,q)}$ for its $s+1$st operation. This notation allows us to model reentrant flow, which is an essential feature in semiconductor manufacturing, and appears in other kinds of manufacturing as well. It distinguishes the same part type (q) at the same machine ($M_i, i = S(s,q)$) at different stages (s).

Machines are flexible and operate on one part at a time. No cost is incurred or time spent in changing the machines from doing operations on one part to operations on another. (Relaxing these restrictions is an important research goal. See Section 4.)

[1] To satisfy space requirements, this version of this paper will forego the usual literature survey. It will be included in a later version.

The machines are unreliable. The fraction of time that M_i is operational is e_i and μ_{iq} is the maximum rate at which Machine M_i can do operations on type q parts. Assume that the demand for Type q parts is constant at rate $d_q > 0$, and that

$$\sum_{\{s,q|S(s,q)=i\}} \frac{d_q}{\mu_{sq}} < e_i \text{ for all } i \tag{1}$$

in which the sum is taken over all parts at all production stages that visit Machine M_i.

Let Machine M_i's repair state be given by α_i. When M_i is operational, $\alpha_i = 1$. When it is under repair, $\alpha_i = 0$. Let $u_{sq}(t)$ be the production rate of type q parts at stage s (which is performed at machine M_i, $i = S(s,q)$) at time t. Then

$$\begin{array}{ll} \text{if } \alpha_i(t) = 0, & u_{iq}(t) = 0; \text{ for } q = 1, ..., \ell; s = 1, ..., K(q); S(s,q) = i \\ \text{if } \alpha_i(t) = 1, & \displaystyle\sum_{\{s,q|S(s,q)=i\}} \frac{u_{sq}(t)}{\mu_{sq}} \leq 1; u_{sq} \geq 0. \end{array} \tag{2}$$

The production rates are the control variables in this formulation. We define $u_{sq}(t)/\mu_{sq}$ to be the *fraction of the capacity of Machine M_i devoted to part type q at stage s.*

The buffers are *homogeneous* — each buffer holds parts that are indistinguishable, except possibly according to arrival time, due date, etc. That means that only a single part type at a *single production stage* appears in each buffer. (See Section 4 for a discussion of some extensions.)

Let $b_{sq}(t)$ be the level of Buffer B_{sq} at time t, $s = 1, ..., K(q) - 1$. Then

$$\frac{db_{sq}}{dt} = u_{sq} - u_{s+1,q} \tag{3}$$

Buffer levels must satisfy

$$0 \leq b_{sq} \leq N_{sq} \tag{4}$$

where N_{sq} is the *size* of Buffer B_{sq}.

Let $x_{K(q),q}$ satisfy

$$\frac{dx_{K(q),q}}{dt} = u_{K(q),q} - d_q, q = 1, ..., \ell \tag{5}$$

This is the *system surplus.* It is the set of cumulative differences between production and demand. It is not bounded above or below. If it is negative, customers are not satisfied. If positive and large, there is a large finished goods inventory. Keeping $x_{K(q),q}(t)$ near zero is a major objective.

We generalize the concept of the system surplus by defining the *local surplus*

$$x_{sq} = \sum_{j=s}^{K(q)-1} b_{jq} + x_{K(q),q}, \tag{6}$$

which satisfies

$$x_{sq} - x_{s+1,q} = b_{sq}, \tag{7}$$

$$\frac{dx_{sq}}{dt} = u_{sq} - d_q \tag{8}$$

The system states are (α_1, ..., α_k) and (x_{11}, ..., $x_{K(\ell)-1,\ell}$, $x_{K(\ell),\ell}$). A natural formulation of a dynamic programming problem is then

$$J = \min E \int_0^T g(b_{11}(s), b_{12}(s), ..., x_{K(\ell),\ell}(s))ds \tag{9}$$

subject to (2), (4), (5), (7), (8),
and the dynamics of α

in which g is a function that penalizes surplus, backlog, and inventory; and T is a long time.

2.2 Summary of Previous Dynamic Programming Results

Kimemia and Gershwin (1983) established the general form of the policy for a single stage, multiple part type manufacturing system, using earlier general work of Rishel (1975). The dynamic programming problem divides the surplus space into regions; the optimal control vector in each region is constant, at a corner of the constraint set. The regions meet at a point, and the control vectors all move the surplus toward that point — called the *hedging point*, Z_{sq}. (All the surpluses are positive or zero at the hedging point.) The surplus moves away from that point only when a machine failure occurs. If the feasibility condition is satisfied, it always returns to it. Bielecki and Kumar (1988) obtained a complete analytic solution for a one-machine, one-part-type special case.

Srivatsan (1993) obtained a partial solution for systems with two part types and one machine. He found that when the hedging point is zero, the optimal boundaries are such that one part type has priority over the other. When the surpluses of both parts are negative, the policy always selects the high priority part until its surplus reaches zero. After that, it maintains that surplus at zero, and uses all remaining capacity to bring the other part's surplus to zero. However, he and Dallery also showed that this simple priority scheme is not always optimal when the hedging point is not zero (Srivatsan and Dallery, 1998).

Van Ryzin, Lou, and Gershwin (1993) found a numerical solution for a system with one part type, two machines in tandem, and one buffer. A rough approximation of their observations is that each machine is run according to a local hedging point policy. However, if the first machine is blocked by a full buffer or the second is starved by an empty buffer, that machine is operated at a reduced rate, which is determined by the other machine.

Bonvik (1996) performed simulation experiments on related policies in tandem systems, and found that a policy, similar to the one presented here, worked well.

2.3 Statement of Policy — Surplus Version

Gershwin (1997) conjectured an approximate solution for the general version of (9), including reentrant flows. This policy implements the hedging point policy at each machine.

A summary of the policy is:

1. In advance
   - Determine priority rankings, for all part types at all machines.
   - Determine all buffer sizes.
   - Determine hedging points, for all part types at all machines.
2. In real time, at each machine,
   - If the surplus of the highest ranking part is not at or above its hedging point, devote maximum available capacity to it.
   - If the surplus of the highest ranking part is at its hedging point, devote as much capacity as is needed to keep it there. If this is not possible, devote all available capacity to that part type.
   - Devote as much remaining capacity as possible to bringing the second ranking part to its hedging point, or maintaining it there.
   - Repeat for all part types, in priority order, until there are either no more parts or no more capacity available.

Devoting maximum available capacity to part type q at stage s means choosing $u_{sq} = \mu_{sq}$ if (s, q) is the highest ranking part at machine $M_{S(s,q)}$. If (s, q) is not the highest ranking part, then u_{sq} is receiving all available capacity if

$$u_{sq} = \mu_{sq} \left(1 - \sum_j \frac{u_{vj}}{\mu_{vj}} \right)$$

in which the sum is taken over all higher ranking parts such that $S(v, j) = S(s, q)$.

If buffer B_{sq} is full, the rate at which machine $M_{S(s,q)}$ may work on part type (s, q) is limited by the speed at which Machine $M_{S(s+1,q)}$ is currently working on part type $(s + 1, q)$.

When buffer sizes N_{sq} are small, the b_{sq} arguments of g in (9) will be small, and that will tend to reduce J. However, buffers that are too small may prevent the system from having the flexibility to keep $x_{K(q),q}$ near zero, so J would be large. Small hedging points will similarly keep in-process inventories (b_{sq}) small, but will also make it more likely that $x_{K(q),q}$ will become very negative in the event of a machine failure.

2.4 Statement of Policy — Due Date Version

The surplus version of the policy is not in a form which is natural for factories to implement. It is possible to convert it to a form which makes use of due dates and conservative estimates of lead times (which we call *hedging times*). The hedging time for part q at its sth stage is Z_{sq}/d_q. This conversion is exact when the demand is constant; and this version is sensible in any case.

A part is *available* at a control point if it is present and has had the previous operation, and if the buffer of that part type immediately downstream of the control point is not full. A part is *ready* at a control point if it is available and

$$\text{the current time} + \text{the hedging time} > \text{the due date.}$$

The policy is:

1. In advance
   - Determine priority rankings, for all part types at all machines.
   - Determine all buffer sizes.
   - Determine hedging times for each machine, for each part type.
2. In real time, at each machine, whenever it is available,
   - Find the highest ranking part type that is ready. Load it and work on it immediately.
   - If no available parts are ready, wait. Look at the system again when either another part arrives from upstream or one of the available parts becomes ready, whichever comes first. Then go to the previous step.

2.5 Statement of Policy — Token Version

A third version of the policy makes performance analysis by decomposition possible (after certain research steps are performed successfully). It is based on Bonvik's (1996) observation that a machine operated according to the hedging point policy is equivalent to an assembly system. To do this, we introduce tokens, which travel in the network like parts.

Figure 1 demonstrates this for a single machine making multiple part types. On the left is a machine that processes three part types, which is controlled by the surplus version of the hedging point policy. On the right is the same machine. The machine generates parts as before (solid lines) and it also generates *production tokens* (dashed lines). (We have thus transformed the machine into a flexible disassembly machine: when it does an operation, one part comes in and two parts go out.) The tokens go to three homogeneous finite buffers. Also on the right are three demand-generation machines (one for each part type) each of whose output is *demand tokens*. The tokens go to three homogeneous infinite buffers. Both sets of buffers send their tokens to three assembly machines, one for each part type.

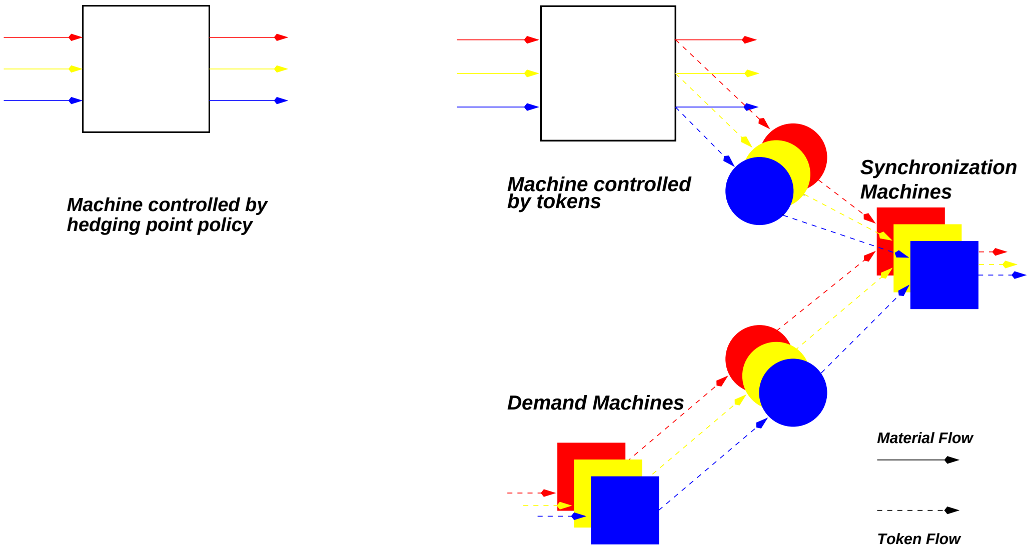

Figure 1: Hedging and token equivalence

The assembly machines, which are in this context called *synchronization machines*, are perfectly reliable and infinitely fast. As a consequence, we have the following crucial point: *each part type may have tokens in its upper buffer or in its lower buffer, but not both.* For each part type, the difference between the number of tokens in the upper buffer and the number of tokens in its lower buffer is the surplus. The sizes of the upper buffers are the hedging points.

Parts are ranked. The machine first allocates as much capacity as it can to processing type 1s. If the upper type 1 buffer is not full, that rate is μ_1. If the buffer is full, the rate that the machine produces type 1s is d_1, exactly the same as the rate at which the type 1 demand machine generates tokens. In either case, the machine may have additional capacity, which it allocates first to type 2. Again, the type 2 buffer may or not be full, and all remaining capacity is allocated to type 3. If all three buffers are full, the rates that the machine processes all three parts are d_1, d_2, and d_3.

To complete the picture for a general case, we add the token flow control mechanism to every machine in the system. This does not affect the material flow paths, but it does affect the timing of when material flows. Thus, if we add more machines and buffers to Figure 1 (for example, so that each part type visits one machine, then goes to a buffer, and then goes to the other machine), we must have as many synchronization machines as the number of real machines times the number of part types, and we must have two new buffers for every synchronization machine.

By creating tokens in this manner, we have created systems that operate without the need for an external controller. These systems have the following characteristics. All statements apply to all parts types, machines, and buffers; we need no longer distinguish between real machines and synchronization or demand machines, and we need not distinguish between parts and tokens.

- The machines may be flexible and may be unreliable.

- Parts are ranked. The ranking may differ at different machines.
- The machines may operate on a single part at a time and produce a single part; or they may be more general assembly/disassembly machines. That is, a machine, when it is working on type (s, q) parts, may take in $\Upsilon_{sq} \geq 1$ different components (each from its own buffer) and put out $\Delta_{sq} \geq 1$ different sub-parts (each into its own appropriate buffer).
- Buffers are homogeneous and are each connected to two machines. Buffers may be finite or infinite.
- Each kind of material has a specified route.
- There are closed loops (primarily due to the single set of demand machines).
- There may be reentrant flow.

2.6 Variable Demand

All three versions of the policy produce exactly the same behavior when the demand is constant. When the demand is variable, however, the behaviors need not be the same, and we must add one restriction to the token version for it to be meaningful: There can be only one demand generator for each part type. The consequence of this restriction is discussed in Section 3.1. (Note that we can use decomposition to deal with stationary random demand, by making the demand machines unreliable.)

2.7 Practical Considerations: Control Points and other Variations

The effort to implement the control policy at each machine may prove to be substantial. For that reason, industry colleagues have suggested that the policy be implemented as described above only at a limited set of points in the system, called *control points*. A simpler policy, such as FIFO, may be implemented elsewhere. When the due date version of the policy is run with limited control points, the policy becomes very similar to Goldratt's drum–buffer–rope policy (Goldratt, 1990).

This leads to other possible variations: Instead of limiting inventory by considering the contents of individual buffers, the policy can limit it according to the total amount of material of a given type in all buffers between adjacent control points.

This affects the preparation phase of the policy: less work is required in establishing hedging points or times and possibly in setting buffer sizes. However, there is now one additional step: selecting the control points.

3 Analysis, Decomposition, and Design

We have not yet discussed the selection of the parameters: the buffer sizes and the hedging points (or hedging times). The strategy for parameter selection is to develop a method for estimating the cost function (9) in steady-state as a function of the parameters, and then optimize over the parameters.

3.1 Decomposition

To predict the performance of a factory, we must have a working decomposition for the class of networks described in Section 2.5.

Decomposition methods have been designed to predict the performance of flow systems that have no closed form analytical solution. A set of two-machine lines is created, in which each line corresponds to one of the buffers in the original system. Such small systems are easy to evaluate. The strategy is to create systems such that an observer in the buffer of the small line could be convinced that he is watching material flow in the corresponding buffer of the original line. Thus developing a decomposition method consists of (1) developing the two-machine line model and solution, (2) developing a set of equations that relate the parameters of the two-machine lines to the behavior of other two-machine lines, and (3) developing an algorithm to solve the equations.

Decomposition methods have been successful for systems that are special cases of the class defined in Section 2.5. The original systems that were studied were tandem lines with single part types. This was extended to assembly/disassembly systems without loops. (Literature surveys can be found in Dallery and Gershwin, 1992; and Gershwin, 1994.) Methods exist for systems with single loops, but they are not as accurate as for acyclic systems. Loops are important because the single demand generator leads to loops: one for each part type at each control point. There has been recent work on two-part-type systems, but they also suffer from problems of accuracy and convergence when the system is not small (Nemec, 1999).

3.2 Optimization of Policy

Schor (1995) and Schor and Gershwin (1996) demonstrated efficient methods for optimizing buffer sizes in single part-type tandem systems. The methods were efficient because production rate exhibits continuity (or near continuity, in discrete-material systems), monotonicity, and concavity as a function of buffer sizes. (The latter properties do not hold when the objective includes average inventory, but there is only a single global optimum and the methods appear to work well.) We expect these properties to continue to hold, and we therefore expect that if the decomposition methods can be extended to the class of systems we need, optimization of buffer sizes — and therefore of hedging points — should not be difficult.

3.3 Optimization of System Design

Factory design is, in principle, a problem which is a superset of the parameter optimization problem described above. In addition to the continuous variables that must be selected to choose the hedging points and buffer sizes, there are also a set of binary or integer variables that indicate which machines, and how many copies of each, should be included. Capital and operating cost now becomes an important part of the objective function.

4 Research Program

The following is a list of research areas in which progress will have an important impact on the analysis, scheduling, and design of manufacturing systems. These will be discussed in an expanded version of this paper. Some progress has already taken place in some of these areas.

- ***Decomposition***, to analyze the systems of Section 2.5
  - ⋆ Multiple part types
  - ⋆ Non-homogeneous buffers
  - ⋆ Multiple closed loops
  - ⋆ Yield, scrapping, and rework
  - ⋆ More general two-machine lines
  - ⋆ Evaluation of J
- ***Dynamic Programming***, to extend the problem formulation and solution
  - ⋆ Workstations containing different kinds of machines
  - ⋆ Determining the local DP problem from the system problem
- ***Other***, to make the results more useful
  - ⋆ Batches and Setups
  - ⋆ Selecting control points

Acknowledgments

I am grateful for the support that has been provided by the Lean Aircraft Initiative, the National Science Foundation, and the Semiconductor Research Corporation.

References

[1] T. Bielecki and P. R. Kumar (1988), "Optimality of Zero-Inventory Policies for Unreliable Manufacturing Systems," *Operations Research,* Volume 36, No. 4, July-August, 1988, pp. 532-541.

[2] A. M. Bonvik (1996), Performance Analysis of Manufacturing Systems Under Hybrid Control Policies, Ph.D. Thesis, Massachusetts Institute of Technology, Operations Research Center, June, 1996.

[3] Y. Dallery and S. B. Gershwin, "Manufacturing Flow Line Systems: A Review of Models and Analytical Results," *Queueing Systems Theory and Applications*, Volume 12, December, 1992, pp. 3-94.

[4] S. B. Gershwin, *Manufacturing Systems Engineering,* Prentice-Hall, 1994.

[5] S. B. Gershwin (1997), "Design and Operation of Manufacturing Systems — Control- and System-Theoretical Models and Issues," Proceedings of the 1997 American Control Conference, Albuquerque, New Mexico, June, 1997, pages 1909-1913.

[6] E. M. Goldratt (1990), *Sifting Information Out of the Data Ocean: The Haystack Syndrome*, North River Press, 1990.

[7] J. G. Kimemia and S. B. Gershwin (1983), "An Algorithm for the Computer Control of Production in a Flexible Manufacturing Systems," *IIE Transactions*, Vol. 15, No. 4, pp. 353-362, December, 1983. Reprinted in *Modeling and Control of Automated Manufacturing Systems* by Alan A. Desrochers, IEEE Computer Society Press Tutorial, 1990.

[8] Joseph E. Nemec (1999), "Diffusion and Decomposition Approximations of Stochastic Models of Multiclass Processing Networks," MIT Operations Research Center Ph. D. thesis, February, 1999.

[9] R. Rishel (1975), "Dynamic Programming and Minimum Principles for Systems with Jump Markov Disturbances," *SIAM Journal on Control,* Vol.13, No.2, February, 1975.

[10] J. Schor (1995), "Efficient Algorithms for Buffer Allocation," MIT EECS MS Thesis, May, 1995.

[11] James E. Schor and Stanley B.Gershwin (1996), "Efficient Algorithms for Buffer Allocation," 1996 Manufacturing and Service Operations Management Conference Proceedings, Dartmouth College, The Amos Tuck School, Hanover, New Hampshire 03755.

[12] N. Srivatsan (1993), "Synthesis of Optimal Policies for Stochastic Manufacturing Systems," Ph.D. Thesis, Massachusetts Institute of Technology, Operations Research Center, September, 1993.

[13] N. Srivatsan and Y. Dallery (1998), "Partial characterization of optimal hedging point policies in unreliable two-part-type manufacturing systems," *Operations Research*, Vol. 46, pp. 36-45, 1998.

[14] G. J. Van Ryzin, X.-C. Lou, and S. B. Gershwin (1993), "Production Control for a Tandem Two-Machine System," *IIE Trans.*, Volume 25, No. 5, pp. 5-20, Sept., 1993.