December 1987 LIDS-P-1729

OPTIMAL CONTROL RULES FOR SCHEDULING JOB SHOPS

Sheldon X.C. Lou
Massachusetts Institute of Technology

December 25, 1987

Abstract

In this paper, we develop the control rules for job shop scheduling based on the *Flow Rate Control* model. We derive optimal control results for job shops with work station in series (transfer line). We use these results to derive rules which are suboptimal, robust against random events, and easy to implement and expand.

1 INTRODUCTION

The success of a job shop scheduling (sometimes called Short Interval Scheduling in contrast with the long term scheduling for a whole factory) system is primarily determined by its control rules. Unfortunately, due to the extremely complex, often randomly perturbed environment, the rules can not be obtained even from the most experienced managers. Since the search space is extremely large, the rules derived from different search algorithms usually are time consuming. Therefore, they cannot deal with the highly varying job shop environment in real time. There are different dispatching rules, such as First-In-First-Out, Last-In-First-Out, Sortest-Processing-Time. Although they are dynamic, they usually are ad hoc and lack systematic analysis. It is also difficult to determine which rules should be used under given conditions. Further, they often rely on local information such as the number of parts in the buffer of one machine but not the global information of the whole production line.

In this paper, a systematic analysis of optimal job shop scheduling rules is presented. The methodology we use is the *Flow Rate Control* approach, which is based on stochastic control theory and dynamic programming algorithms.

The job shop environment is characterized by many random events such as machine failures, demand, and yield. If the job shop is not fully automated, which in general is the case, the interference of the human operators (*e.g.* operators may make mistakes) should also be considered. Therefore, a successful scheduling algorithm should be robust in the presense of random interferences.

The algorithm should also be relatively simple, simple to understand and simple to implement. Moreover, it should be simple to expand when new machines and part types are added.

The scheduling rules proposed in this paper is robust and simple. Instead of providing a static schedule, it provides feedback control which is determined on line by the current state of the job shop. It adjusts the production according to changes which occur in the job shop. Further, the software can be easily expanded by adding new rules.

We first explain why, in deriving the rules, the flow rate control model is chosen to model a job shop. Then, the methodology for finding the optimal (or suboptimal) rules is presented, and compared with other possible choices. Based on this analysis, the optimal rules are derived.

2 ISSUES RELATED TO THE MODEL

2.1 THE FLOW RATE CONTROL MODEL

The primarily concern of a job shop scheduling system is the high dimension of the search space. It is well known that the scheduling problem is in general NP-hard. Without successful decomposition to reduce the dimension, real time production control is impossible. The scheduling approach based on flow rate control model contains two levels [13, 9]. At the high level, the manufacturing process is considered as a *continuous flow of materials* with random interruptions such as machine failures, processing time fluctuations, insufficient raw material supplies, random yield, and random demand. The production rate of each work station is determined by optimal control rules. At the lower level the detailed tracking of individual parts is considered. Taking this approach enables us to greatly reduce the dimensionality. It also permits us to apply stochastic control and optimization theories to the job shop scheduling problem, to obtain results superior to other methods such as simple dispatching rules. But, this approach is not applicable for all kinds of job shops. The general job shop scheduling problem remains as a challenge for further research. The continuous flow model works when there is production of sufficient volume so that a production rate makes sense. Many job shops, however, belong in this category.

Using this methodology, the desirable controls, roughly speaking, will *reduce*

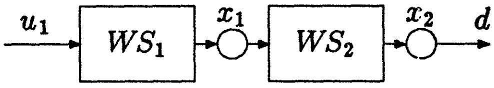


Figure 1: A two-*WS* system

the WIP (Work-In-Process) as much as possible while closely following the target production and observing the machine capacity constraints in a randomly perturbed job shop environment.

In this paper, we concentrate our attention on the high level control, *i.e.* the production control of work stations. In order to gain some idea about the model, let us start from a simple job shop containing two work station, shown in Fig. 1 (see [22] for more detail). State equations for this system are

$$\begin{aligned} x_1(k+1) &= x_1(k) + u_1(k) - u_2(k) && (1)\\ x_2(k+1) &= x_2(k) + u_2(k) - d(k) && (2)\\ 0 &\leq x_1(k) && (3)\\ 0 &\leq u_1(k) \leq \alpha_1(k) && (4)\\ 0 &\leq u_2(k) \leq \alpha_2(k) && (5) \end{aligned}$$

where $u_i(k)$ is the number of parts loaded in unit time interval at WS_i (the loading rate) at time k and $d(k)$ is the planned (target) production rate at time k. Note that $x_1(k)$—the inventory after the first work station—is restricted to be non-negative. The variable x_2 is defined as the *surplus*—the difference between the actual production and the target production. It can be positive, meaning there is an inventory at the last stage, or negative meaning a backlog due to insufficient production exists.

The objective is to minimize the discounted, infinite-horizon cost

$$\min_{u \in \Omega(t)} E \sum_{k=0}^{\infty} \beta^k g(x_1(k), x_2(k)) \qquad (6)$$

where $\Omega(\alpha)$ is a polyhedron defined by (4) and (5), $g(\cdot)$ is a convex function of x_1 and x_2, and β is a discount rate between 0 and 1. We use a $g(\cdot)$ which has the form as shown in Fig. 2, which can be characterized by the slopes c_1, c_2^+ and c_2^-.

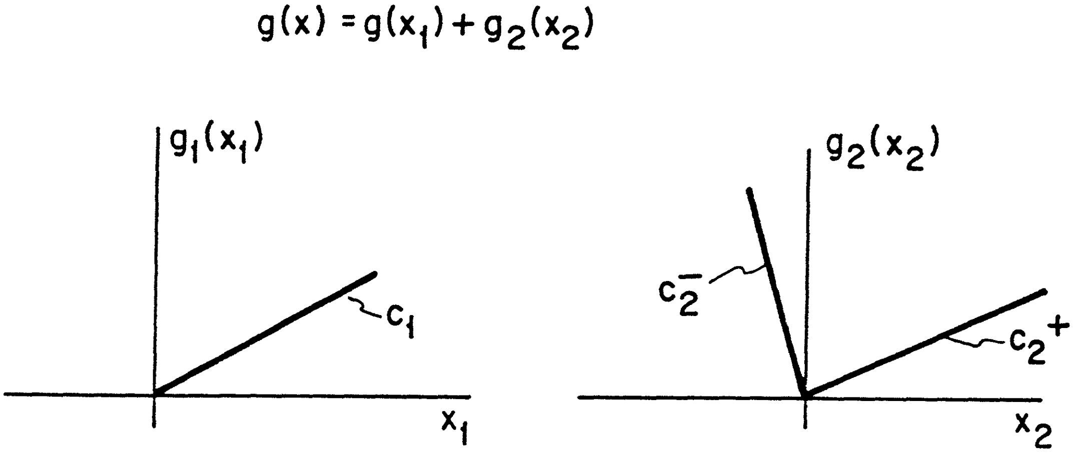


Figure 2: Piece-wise linear $g(\cdot)$ function

In this paper, we only consider the *Transfer Line*, where machines are connected in series.

2.2 REASONS FOR CHOOSING A FLOW RATE MODEL

After comparing existing methodologies to obtain scheduling rules including combinatorial optimization, queuing network theory, heuristic dispatching rules, we came to the conclusion that the flow rate control model was most promising for our purpose.

Job shop scheduling is one of the oldest and hardest problems in manufacturing and has attracted the attention of many researchers. But, due to the combinatorial nature of the problem, it remains unsolved. Except for very few problems, under specific conditions, the exact solution of the job shop scheduling problem formulated as a combinatorial optimization problem is known to be computationally intractable [6, 15]. There are at least three classes of methods for dealing with this problem. The first one uses a heuristic search, such as branch and bound [17], or constraint-directed search [8], to prune the search tree. However, heuristic algorithms are not efficient enough to reduce the computational burden to a realistic level. Furthermore, they are based on deterministic assumptions, namely, that machine states, yield, and processing times are all deterministic. Any major perturbations changing the present conditions require a recomputation, which is often impractical due to the computational complexity.

The second class of methods is based solely on heuristics [5]. The results are generally tested by simulation under some specific conditions. Although some heuristic rules are dynamic, most only take *local information*, such as the inventory at each machine into account. They are also ad hoc.

The third class uses queuing theory. Queuing network theory is generally used to *model* a system, but not to *control* the system.

Since a *complete* and *exact* solution (an optimal solution which takes every detail into account) is difficult, a natural compromise is to try to ignore some information of *secondary importance* so that the search space can be reduced and the issues of primary concern can be taken care of. Since the flow rate control model groups together part types at the high level, one only worries about the **production rate** of each part type not the location of individual part. The part dispatching is carried out at the lower level. This hierarchical structure greatly reduces the computation burden by distributing computation to each level. Using this model, one can use stochastic control theory to achieve a feedback control law that responds to random interruptions.

2.3 WHAT IS NEW IN OUR MODEL

The flow rate control model has been used in [13, 1 and 9], where a work station with negligible delay and internal inventories was considered. In this work, the **state** is the surplus of the work station. A feedback law then determines the production rate of each part, taking the current machine states into account. This paper extends the flow rate model to job shops with multiple work stations with significant internal inventories[1].

The major difference between our work and earlier application of flow rate model is that we allow *internal buffers.* Without internal buffers, there must be a unique production rate *throughout* the whole system. There are many systems, however, where a single production rate is not desirable. For example, consider several machines connected in series with buffers between successive machines. If one machine in this chain of machines is down, it may not be necessary in general to stop other machines (if they are not starved, i.e. the previous machine cannot provide parts, or blocked, i.e. the immediate down stream machine is not working). Indeed, internal buffers are used primarily to prevent the whole line being stopped when only a few machines are down.

A system with buffers was considered in [11] where a discrete time system model with a linear control rule was established. In our model, instead of analyzing some specific control rule, we try to determine the *optimal* one. Also, the capacity constraints (the maximum machine loading rates) and the random machine states are taken into account.

Similar models can also be seen in queuing network literature where the research purpose is to *estimate* the parameters of a given system under a *given*

[1] The system with significant delays were addressed in a separate paper [19].

control rule. However, our model is used to *derive* the optimal control rules, not to simply model a system. A system similar to that proposed in this paper has been analyzed in [14]. The major difference is that in our model, the *Surplus*–the difference between the real and the target productions (it can be negative if the real production is behind schedule) at the last work station, is observed while in [14] only the number of parts in the last buffer is considered. As we will see shortly, this difference is essential.

Furthermore, the optimal control derived in this paper is presented as *simple rules,* which are easy to understand and implement.

In the next section, we describe our solution approach.

3 SOLUTION APPROACH

Although the flow rate control model greatly reduces the dimension of a problem, the direct solution of any problem of practical importance is still formidable. The computation for this dynamic program is still NP-hard. In [2, 4] the closed form solution for a one–machine one–part system is given. Although the results provide great insight into the problem, extensions to more complex problems appear difficult.

The results in [13, 1] show that control regions are divided by surfaces. Computing the regions requires knowing the **optimal cost to go** functional, $J^*(x)$. But knowing $J^*(x)$ is equivalent to having solved the problem. A quadratic approximation of $J^*(x)$ [1] reduces this burden somewhat, getting a good quadratic J^* is still a very difficult task. Also, the J^* may differ from quadratic drastically (as we show below).

Therefore, instead of searching for a formulation to solve a complex problem in one step, we first find *exact and optimal* solutions (infinite horizon, steady state) for a series of small problems using numerical solution techniques (see [3], [7] and [12]). We then derive several control rules out of these results, which can be applied to a general job shop. This **Rule–Driven** approach satisfies the criteria proposed in Section 1: Rules are clearly defined, easy to understand, simple to implement, and easy to expand in the future. The control of each *WS* governed by these rules is based on the observation of the states of the entire system and are robust.

In the next section, control rules for seriesly connected work stations (the *Transfer Line*) are presented. We start from Two–Work–Station case and then continue to analyze the Three–Work–Station case.

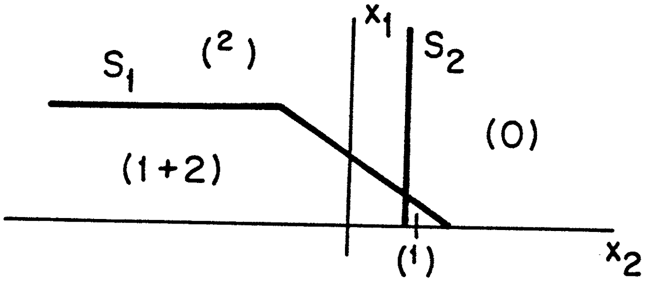


Figure 3: Optimal control regions for a two *WS* system

4 CONTROL RULES FOR A TRANSFER LINE

4.1 TWO WORK STATIONS

We start from a simple case, two work stations producing one part (see Fig. 1 and Eq. (1)–(5)). The *WS*s are not reliable. They can either be up or down. The transition probabilities from up to down (due to failures) are p_{f1} for WS_1 and p_{f2} for WS_2. The transition probabilities from down to up (due to repairs) for WS_1 and WS_2 are p_{r1} and p_{r2} respectively. The *WS*s have limited capacities, *i.e.* when they are up, the maximum number of parts loaded each time interval is finite and denoted by U_{m1} and U_{m2} respectively[2]. In this equation, $x_1(k)$ is defined as the number of parts, *i.e.* inventory, in the first buffer and $x_2(k)$ is the difference between the target production and actual production. Therefore, $x_1(k)$ cannot be negative, while $x_2(k)$ can either be positive, meaning an inventory at the last stage, or negative, meaning a backlog. An optimal control should minimize both x_1 and x_2 so that the inventories can be kept at a low level while following the target production as close as possible. More precisely, the objective can be described as minimizing

$$\min_{u \in \Omega(t)} E \sum_{k=0}^{\infty} \beta^k g(x_1(k), x_2(k)) \tag{7}$$

subject to (1)–(5). Using a Dynamic programming (value iteration) (see [3], [7]) to solve this problem, we can calculate the optimal control law. The control regions when both machines are up is shown in Fig. 3.

The two–dimensional half–space (x_1 can only be zero or positive) is divided by two curves–S_1 and S_2. The second curve S_2, is a straight line parallel to x_1 axis.

[2] When the *WS*'s are down, the capacities will be zero.

It determines the control for the second *WS*. When $x = [x_1\, x_2]'$ lies to the right of S_2, (meaning the surplus is too big), the second *WS* is stopped. Otherwise it keeps operating at full speed (the maximum loading rate is determined by the work station capacity and the number of parts available at the previous buffer, *i.e.* x_1). This implies that *the control is independent of the first WS and operates like the single hedging point control* described in [2, 4].

The control for the first *WS* is quite interesting. When $x = [x_1\, x_2]'$ lies in the region below S_1, the first *WS* operates at full speed. Otherwise it stops. The control can be explained as follows: When x_2 is very negative, *i.e.* there is a big backlog at the output), the system is far behind its schedule. The first *WS*, therefore, tries to store more parts. There is, however, a limit to how much stack is stored. When the storage is beyond this limit, the production is stopped. When x_2 is close to zero or even positive, meaning that the system is close to or ahead of its schedule, the optimal control tries to reduce the storage at the first *WS*. Closer study [23, 22] has shown that this region of S_1 (B–C) can be approximated as $x_1 + x_2 = h_{s1}$. Here $x_1 + x_2$ is nothing but the *SURPLUS* at *WS* 1, the difference between the target and the actual production after the first *WS*. Therefore, the optimal control for the first *WS* can be approximated by two regions according to the value of x_2. We call this strategy a *TWO BOUNDARY CONTROL*, because the first part is a *SIMPLE INVENTORY CONTROL* policy and the second part is a *SIMPLE SURPLUS CONTROL* policy.

In order to extend this approximation to the optimal control for a multiple *WS* system, we next examine a three–*WS* system.

4.2 THREE–*WS* SYSTEM

Consider three *WS*s connected in series, producing a single part as shown in Fig. 4. Again the *WS*s are unreliable. As in the previous section, they can be either up or down. When they are up, they have certain capacity limits. The system equations are very similar to those of the two–*WS* case.

The general shape of the optimal control regions when three *WS*'s are all up, can be seen from Fig. 5.

In this paper, we will study four different cases of 3–*WS* systems, see Table 4.2. They have the same structure as Fig. 4 but different parameters (such as probabilities and cost coefficients).

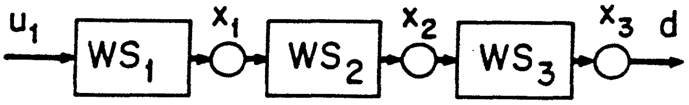


Figure 4: A three *WS* system

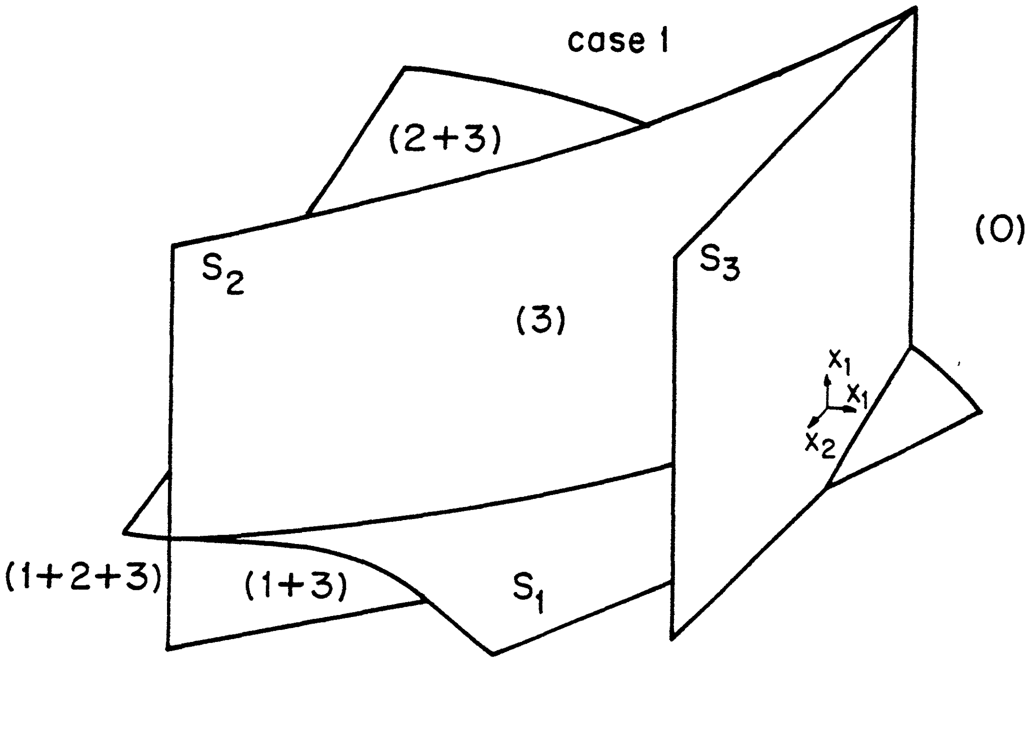


Figure 5: Optimal Control Regions for a 3-*WS* System when all *WS*s are up

	c_1	c_2	c_3^+	c_3^-	p_{f1}	p_{f2}	p_{f3}	p_{r1}	p_{r2}	p_{r3}
case 1	0.5	0.7	2.0	10.0	0.1	0.1	0.1	0.2	0.5	0.2
case 2	0.5	1.0	5.0	10.0	0.1	0.1	0.1	0.2	0.5	0.2
case 3	0.5	0.3	2.0	10.0	0.1	0.1	0.1	0.2	0.5	0.2
case 4	0.5	0.7	2.0	10.0	0.18	0.1	0.1	0.2	0.5	0.2

Instead of two curves S_1 and S_2 as in the two *WS* case, *the optimal control is determined by three surfaces S_1, S_2 and S_3, each corresponding to the control of one WS*. In other words, the i^{th} surface S_i, i=1,2,3, (called *control surfaces*) divides the entire space into two parts. In one part WS_i operates at full speed (again determined by the work station capacities and the contents at the previous buffers). In the other it stops. For example, S_1 determines the control of WS_1. When $x = [x_1 x_2 x_3]'$ is above S_1, WS_1 stops. Otherwise, it operates. The operating regions of each *WS* are denoted by numbers in Fig.5. For example, (1+2+3) means all *WS*'s should be operating.

The optimal control for different *WS* states (different combinations of working and non working *WS*s) of the same system is shown in the next two figures. Fig. 6 shows the optimal controls when only one *WS* is down while Fig. 7 shows the controls when only one *WS* is up.

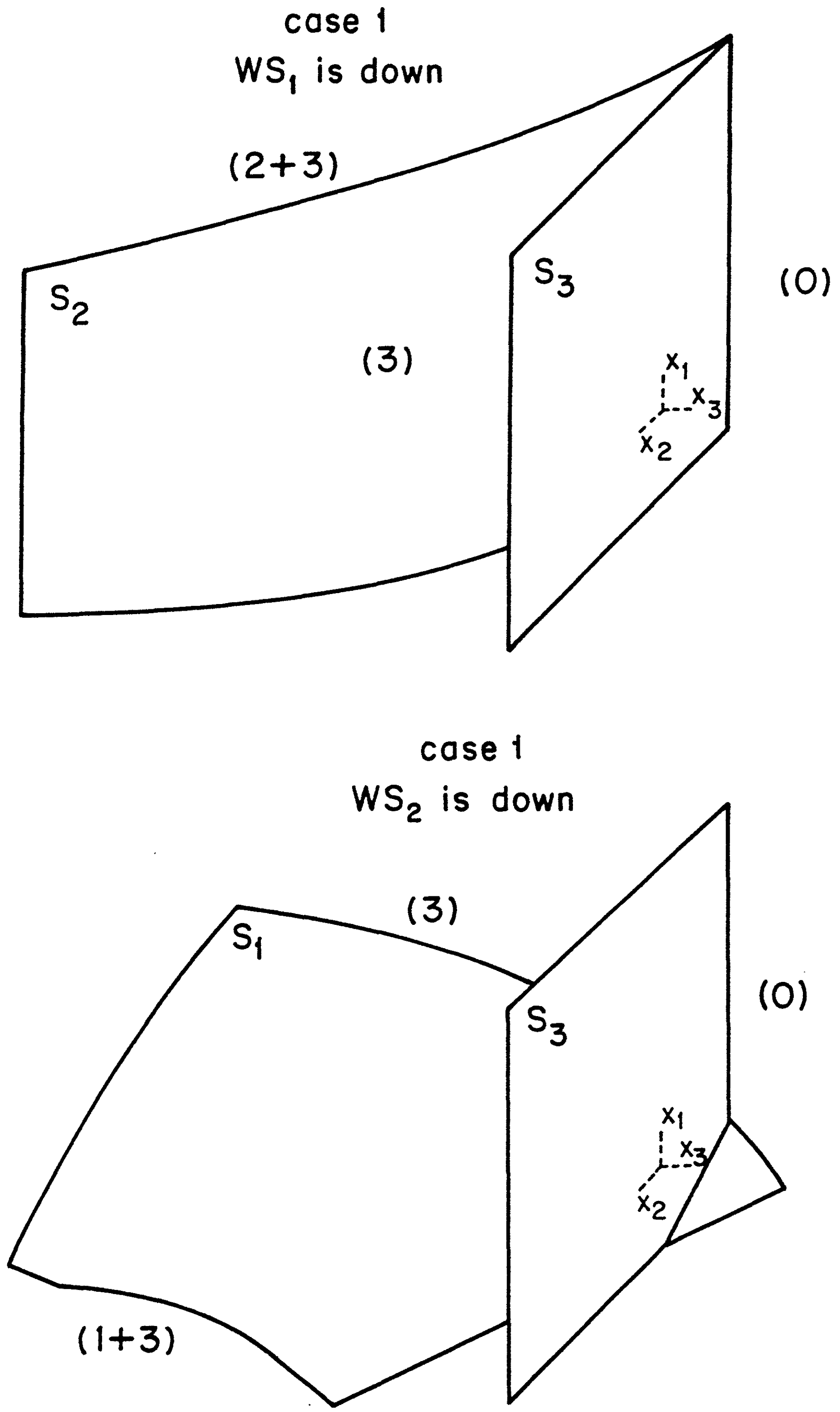
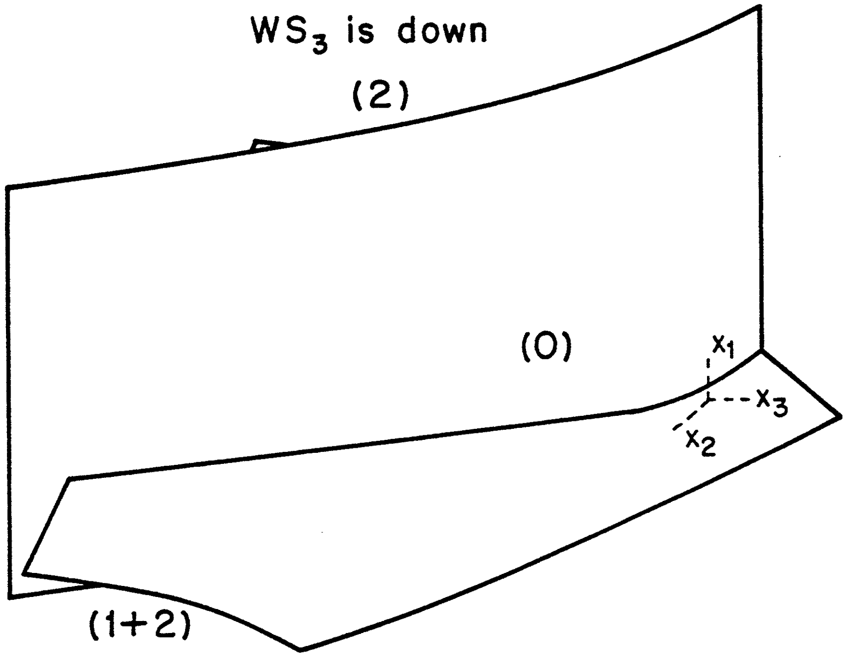


Figure 6: Optimal Control Regions for 3-*WS* System when one *WS* is down

case 1

WS_3 is down



Let us discuss several important features of Fig. 5 to 7 which form a basis for some general control rules.

1. S_3 is a plane perpendicular to the x_3 axis. It only has one degree of freedom: Namely, changing parameters like failure and repair probabilities or capacities only shifts this plane to the left or right along the x_3 axis. S_2, whose projection on the $x_2 - x_3$ plane is a curve, see Fig. 8, has two degrees of freedom. Finally, the S_1 has three degrees of freedom.

   We observed that when considering the control for WS_i, **only the down stream *WS*s (including WS_i) should be taken into account.** It should be pointed out that this does not mean that the up stream *WS*s have no effect at all on the controls of the down stream *WS*s. The ***parameters*** of the up stream *WS*s (such as probabilities and cost coefficients) have influence on the positions of the control surfaces (the S_i) of the down stream *WS*s. However, the ***on line*** decisions of the down stream *WS*s are not affected by the states (buffer levels, *WS* states) of the upstream buffers.

2. The general shapes of the control surfaces can be described as the follows: S_3 again defines a *simple surplus control* (a control determined by comparing the surplus value with a single threshold or hedging point). S_1 defines a *Two Boundary control*, as in Fig. 9. In one region, when x_3 is negative, WS_1 follows a *simple inventory control* (a control determined by comparing the inventory of a work station with some threshold). When x_3 is close to zero or even positive, it follows a Simple Surplus Control. In the surplus region, the operation of WS_1 is determined by the sum of x_1, x_2 and x_3, which is the surplus or the difference between the actual and planned productions of WS_1. The control tries to keep a fixed surplus level.

   For WS_2, we observed a Simple Surplus Control, a control determined by $x_2 + x_3$. Experiment with a larger x region (x_3 can vary from -30 to $+30$) showed a saturation of S_2 as x_3 went negative (Fig.8). Therefore, in general, a **Two-Boundary control is again close to optimal.** It should be pointed out that for the last *WS*, the Simple Inventory and Simple Surplus Controls become the same, because optimal hedging points are always positive (see [4]).

3. Another phenomenon we observed is how the optimal control changes when some *WS*s are down. In Fig. 5, 6 and 7, notice that when WS_i is down, S_i disappears, but the general shapes for the remaining control surfaces remain essentially the same. That is to say **the optimal control is primarily determined by inventories and the surpluses.** The *WS*'s states (down

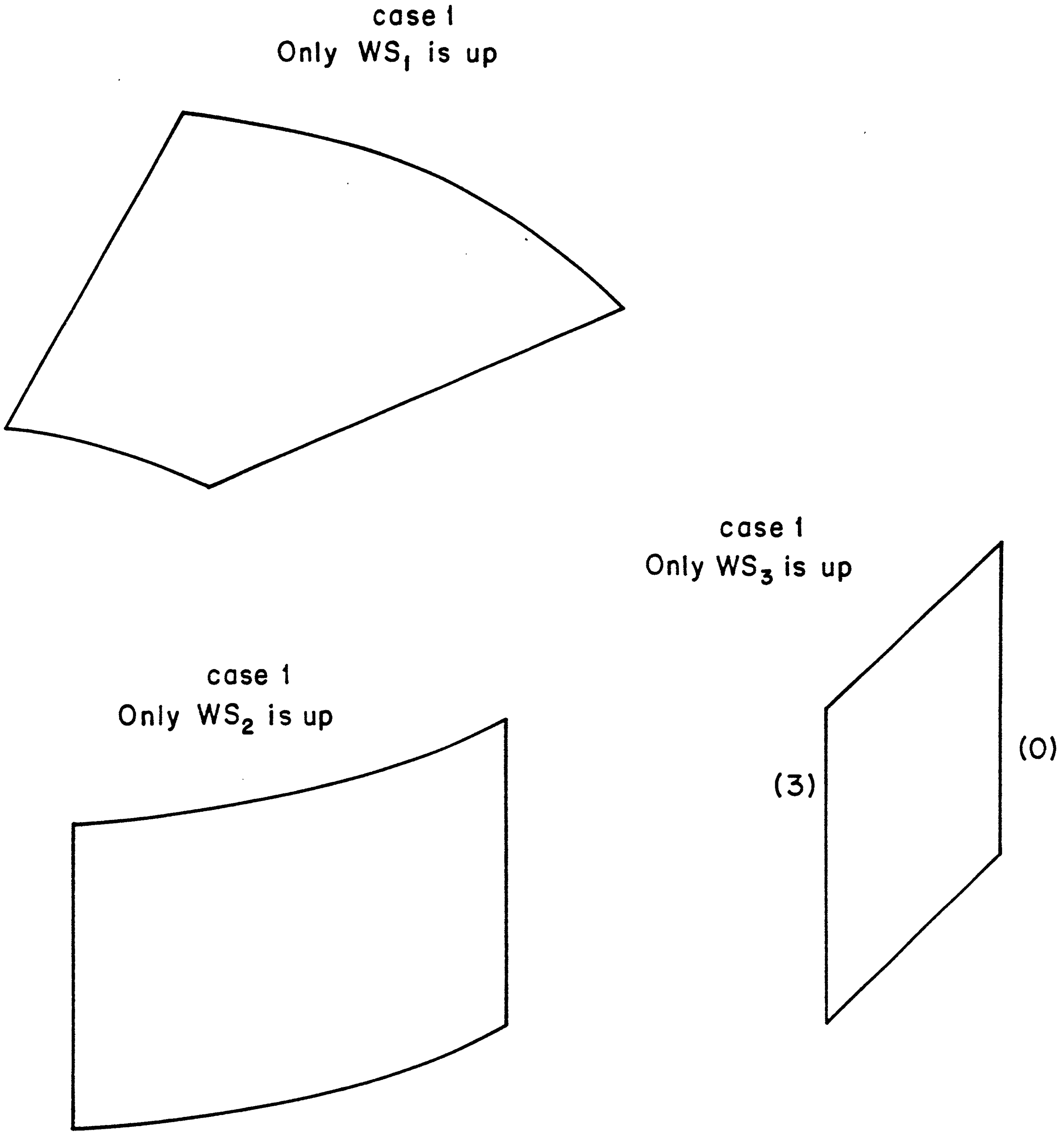


Figure 7: Optimal Control Regions for 3-*WS* System when only one *WS* is up

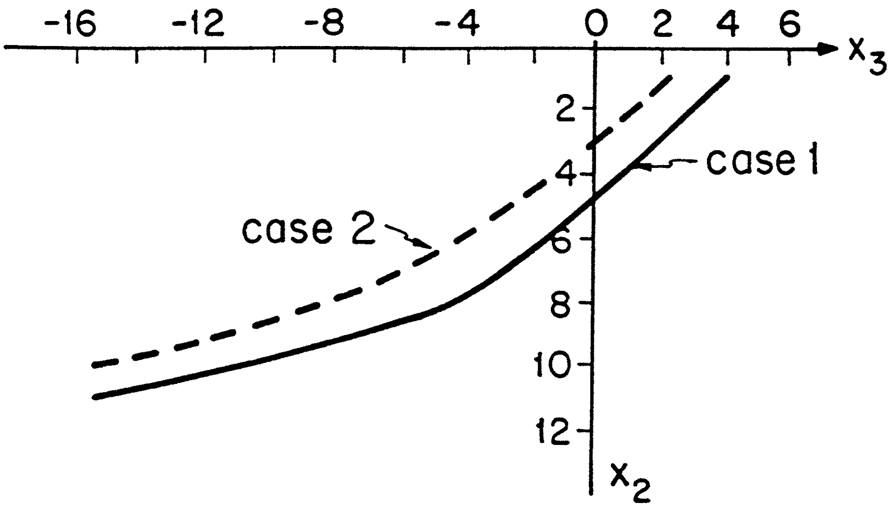


Figure 8: Optimal Control Regions for 3-*WS* System when x_1 is fixed

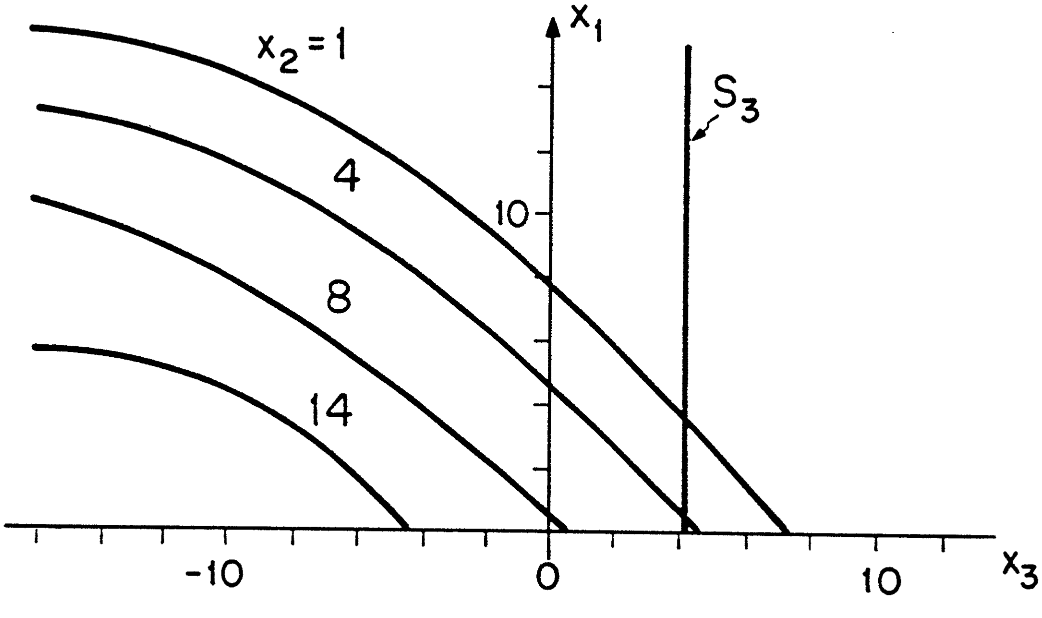


Figure 9: Optimal Control Regions for 3-*WS* System when x_2 is fixed

or up) are of secondary importance. A close comparison of the three figures shows that when WS_{i+1} is down, WS_i (and all upstream WSs) actually try *to reduce their inventories* a little (not to increase it, as one might predict). This is because inventories are used to supply the down stream WSs, preventing them from being starved. When downstream WS's are under repair, they consume no parts. Therefore the inventories of the upstream WS's can actually be less.

4. The trajectory and equilibrium point.

   Each control surface is *attractive*, meaning, the trajectory of x (the position of x as the function of time k) hits any surface, it will stay in that surface (or go zig-zag along the surface) until the WS states change. If there is sufficient capacity, there is an equilibrium point when all the WSs are up. Whatever the initial x is, if the WSs stay up long enough, the trajectory always ends at this equilibrium point and stays there until the WS states change. That is, this is the point at which the system will stay if WSs are up long enough. This behavior implies that the control is stable. An example trajectory is shown in Fig. 11.

5. The effects of the cost coefficients.

   In general, the less costly the storage is (*i.e.* a smaller coefficient c_i), the larger the storage limit will be. In other words, the smaller the c_i is, the higher the hedging point for both inventory and surplus (see Fig. 8). Combining this fact with the equilibrium point discussed above, we see that **an optimal control determines the equilibrium distribution of inven-**

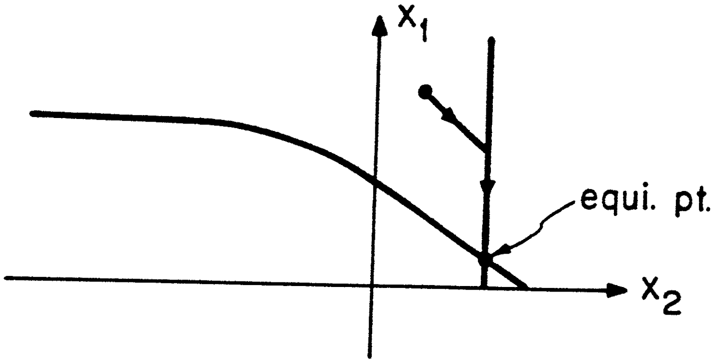


Figure 11: Control trajectory

tories within the system, a phenomenon usually called *Line Balancing.* By properly adjusting the coefficients, an ideal distribution can be achieved.

Usually, the storage costs of the down stream *WS*s are higher than that of the up stream *WS*'s, because the parts processed by the down stream *WS*s have a higher added value. But, what if the storage cost of the $i+1^{th}$ *WS* is less than or equal to that of the i^{th} *WS* ? We observed that in this case there will be no hedging points for WS_{i+1}. Its control surface simply disappears. The optimal control policy for WS_{i+1} is: **Operate WS_{i+1} whenever you can !**

Fig. 13 shows a two *WS* system with a c_2 less than c_1. Fig. 14 shows a three–*WS* system where $c_2 = 0.3$ is less than $c_1 = 0.5$. In both figures, S_2 disappears. The optimal control requires WS_2 to operate in the entire space

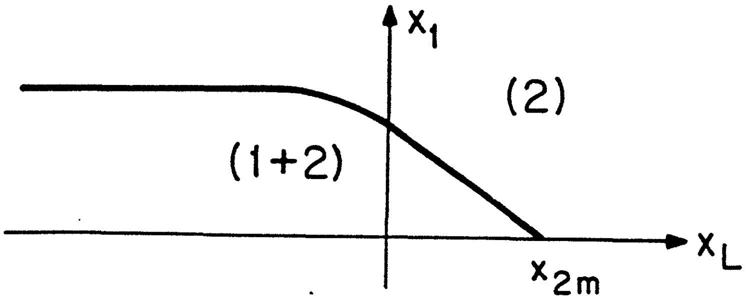


Figure 13: Control region for Two *WS* system, $c_2 \leq c_1$

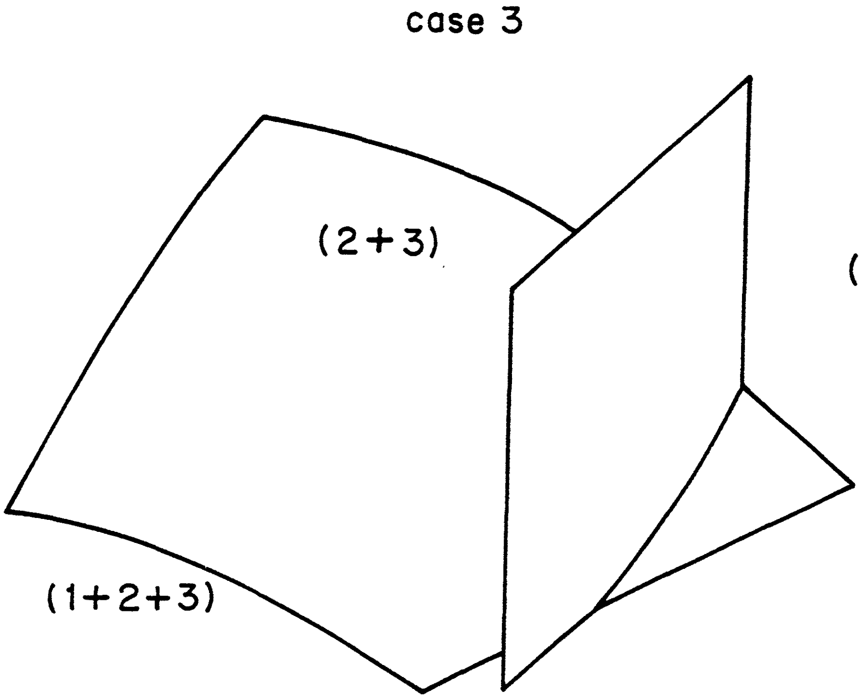


Figure 14: Control regions for Three–*WS* system, $c_2 < c_1$

(we will comment on this further below).

Comparing Fig. 13 with Fig. 11, we notice that

- S_3 slightly shifts to the left, *i.e.* WS_3 tries to reduce its inventory due to the fact that the previous *WS* may store more parts.
- S_2 disappears, as pointed out above.
- S_1 slightly shifts downwards, again because WS_2 stores more parts.

These results are somewhat intuitive. If the storage cost of WS_{i+1} is less than or equal to that of WS_i, it costs less than to leave them at WS_i. Therefore, parts in WS_i are always advanced if possible. The reader may wonder why, since that there is no restriction on WS_{i+1}, the number of parts stored in

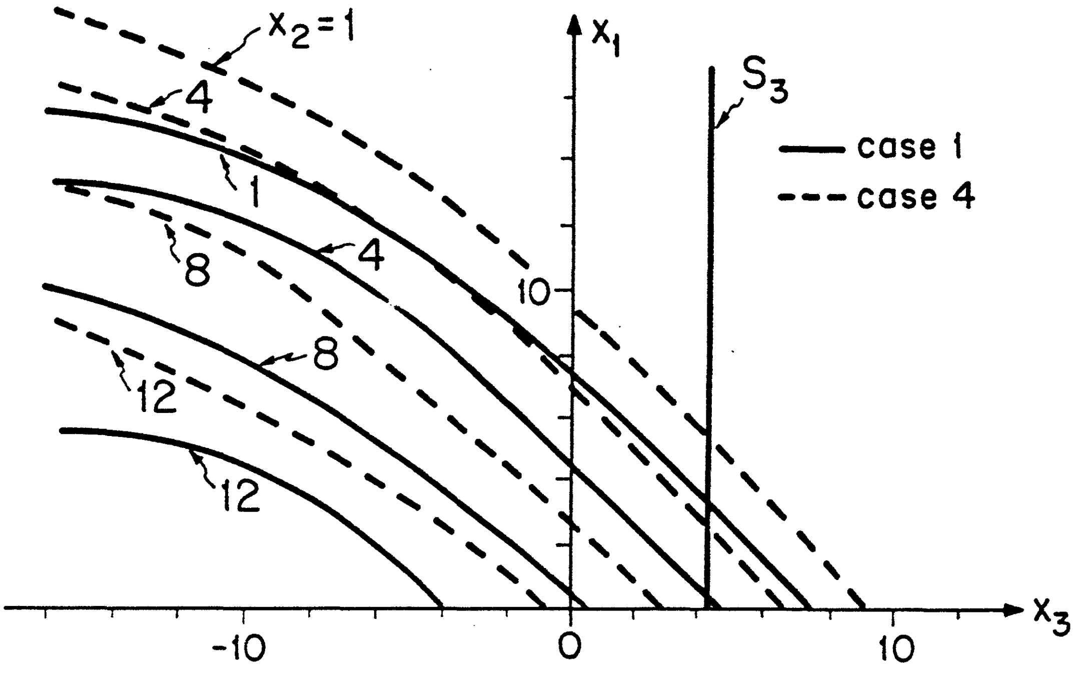


Figure 15: The effect of the machine reliability

WS_{i+1}, *i.e.* x_{i+1}, does not increase without bound ? The optimal control takes care of this. Consider Fig. 13 again. Usually, x_2 can not be larger that x_{2m} (the equilibrium point of the system) because WS_1 would be stopped and WS_2 would be starved. The same is true for the three *WS* case. **The optimal control for WS_i automaticly restricts the inventory of WS_{i+1}.**

This result suggests that one might consider grouping several *WS*s together if the storage costs among them are indifferent. One could then control only the loading process of the first *WS*. For the remaining *WS*s in the group, parts are processed as fast as possible.

6. The effects of the probabilities.

   It is expected that the more reliable the *WS* is, the fewer parts it will store. In Fig. 15, the first set of curves (dotted curves) represent a system with an unreliable WS_1 but the same WS_2 and WS_3. This shift of curves is expected.

5 CONTROL RULE

Based on the above discussion, we propose a control rule which represent a suboptimal solution for a job shop with series–connected work stations (a flow shop).

TWO BOUNDARY CONTROL FOR ALL WORK STATIONS.

For each work station, we

1. Compute its inventory x_i, *i.e.* the number of parts in its buffer[3].

[3] We have noted that the delay (long processing time) issue had been addressed in [19]. If that is the case, we use, as a suboptimal solution, the *total number of parts* in WS_i, *i.e.* the number of parts in the buffer plus the number of parts being precessed as the x_i. See [19, 22 and 23] for detail.

2. Compute its surplus s_i. To determine the surplus, one can calculate **EITHER** of the following two numbers.

    - the difference between the actual and planned productions. That is, the total number of parts loaded into WS_i starting from some initial time—the cumulative production minus the total parts planned to produce since the initial time—the cumulative planned production.
    - the summation of the inventories of the down stream *WS*s (use surplus for the last work station), $x_i + x_{i+1} + \cdots + x_N$.

3. Compare x_i and s_i with two predetermined numbers, ***inventory hedging point*** h_i and ***surplus hedging point*** h_{si}. If WS_i is in the working condition, $x_i \leq h_i$ and $s_i \leq h_{si}$, load WS_i at full speed (considering the capacity constraints and the previous buffer contents). Otherwise do not load WS_i.

This single rule actually contains all the rules we observed in the last Section. It implies,

- Last *WS* follows a Simple Surplus rule. As we pointed out, the Simple Surplus rule is the same as the Simple Inventory rule for the last *WS*.
- Only down stream *WS*s are taken into account. This is reflected in the way we calculate the surplus.
- WIP determines the control. Machine states play a secondary role. As an approximation, here the on line control rules are independent of the work station states— we do not alter the hedging points h_i and h_{si} at all when work station states change.

6 SIMULATION RESULTS

To compare the Two–Boundary control with other production control approaches, let us consider the following example. In this example, four work stations with exponentially distributed down and up times are connected in series. Their parameters are shown in Table 6.

work station	CLEAN	PHOTO	OXID	TEST
Ave-down-time	5	2	40	5
Ave-up-time	200	8	80	200
Process-time	5	7	12	2
Time-between-load	1	2	12	1
capacity	4	20	50	3

In order to model work stations in real life (the parameters of this example were from a real VLSI wafer fabrication facility), we also considered their processing time. The principle for treating work stations with finite processing time can be found in [19]. Here we simply use the total number of parts being processed in WS_i plus the number of parts in its buffer as our x_i. Note also, due to the different natures of the work stations, the minimum time between successive loadings of one work station is different from that of another. For example, Work Station 3 was designed to model a furnace. Since no parts can be loaded into a furnace unless it finishes a batch, the time between loadings must be larger than or equal to the total processing time. On the other hand, other work stations in this example were supposed to contain number of machines in series. The time between loadings is therefore less than the total processing time.

The cost was computed according to Eq. 6 where the weighting factors are $c_1 = 1.0, c_2 = 1.2, c_3 = 1.4, c_4^+ = 1.6, c_4^- = 5.0$. The constant c_4^+ is the weighting for inventory and c_4^- is the weighting for backlog at the last work station. Notice, we penalize backlog three times more than the inventory. In this example, only one part type is produced and the production unit is lot. The target production has a constant rate of two lots per hour.

An Event–Driven simulator designed for job shop production simulation was used (see [??]). The time horizon for each simulation run is 1,000 time units (hours). Four different cases were simulated. Case 1 uses the Two–Boundary control rule. It is assumed that there are infinite number of lots at the buffer before the first work station with no storage cost. Case 2 places 200 lots of parts in the buffer before the first work station at every 100 hours. It is similar to the stratege being used in some companies and called Uniform–Loading in this paper. The third and fourth cases make use of One–Boundary control. Specifically, in Case 3 the Local–Inventory control is used. Namely, for each work station there is a pre-calculated threshold (hedging point). If the x_i is below this threshold, we try to load parts as many as we can. Otherwise we do not load. In case 4 the Surplus Control, which compares the surplus of each work station with certain predetermined threshold to determine the loading, is used.

First let us see the difference between the Two–Boundary Control and the Uniform–Loading. The costs for seven simulation runs of both cases are shown in Table 6. The average total cost is 308.35 for the former and 690.47 for the latter. In other words, the Two–Boundary Control performs two times better than Uniform–Loading.

Two–Bound	289.34	280.12	295.95	290.44	336.90	291.17	374.51
Uni–Load	595.28	663.81	820.17	703.17	657.12	743.49	650.22

It seems more convincing if we look at the sample paths, shown in Fig. 16 and 17, for these two cases with identical work station states variations (the same sample path of work station ups and downs). In the figures, the time variations of x_i for $i = 2, 3, 4$ are shown. We see from the figures, that

- The time horizon can approximately be divided into two periods. In the first period (from t=0 to t=400), since the system has just started from zero inventories and the down time of WS_3 from t=110 to 200 is relatively long, the system is behind the schedule (with a negative x_4) most of the time. In the second period, there is no major breakdowns and the system is in a relatively stable state.

- It is evident that Case 1 overperforms Case 2 in the first period. The reason is also quite obvious. In this period, all the work stations are having very small or negative surpluses most of the time. Therefore for Case 1, only the Local–Inventory thresholds are active. So that a large amount of parts are pumped into the system, which helps the system to catch up with the target production. On the other hand, uniformly loading parts in Case 2 results in a part shortage, which in turn causes the long delay before the system eventually catchs up.

- In the second period when t is greater than 400, the first case again overperforms the second by achieving a smoother inventory variations. (Note, the x_i is the summation of the number of parts being processed and the number of parts in the buffer at the i^{th} work station. Therefore some positive x_i are certainly necessary to keep a smooth production). The inventories of Case 2 in Fig. 16 present wild fluctuations. Further, the x_4 is always less than zero. In other words, system are always having backlog. Only at every hundred hours the production reachs its target.

Now, let us consider the differences between the Two–Boundary control and the one boundary controls, *i.e.* the Local–Inventory Control in Case 3, that has the same Inventory hedging points as in Case 1 but the infinite Surplus hedging points and Surplus Control in Case 4, that has the same Surplus hedging points as in Case 1 but infinite Inventory hedging points. Fig. 18 and 19 are corresponding sample paths for those two cases (again, work stations' ups and downs follow the same sample path as in Fig. 16 and 17). We notice from the figures:

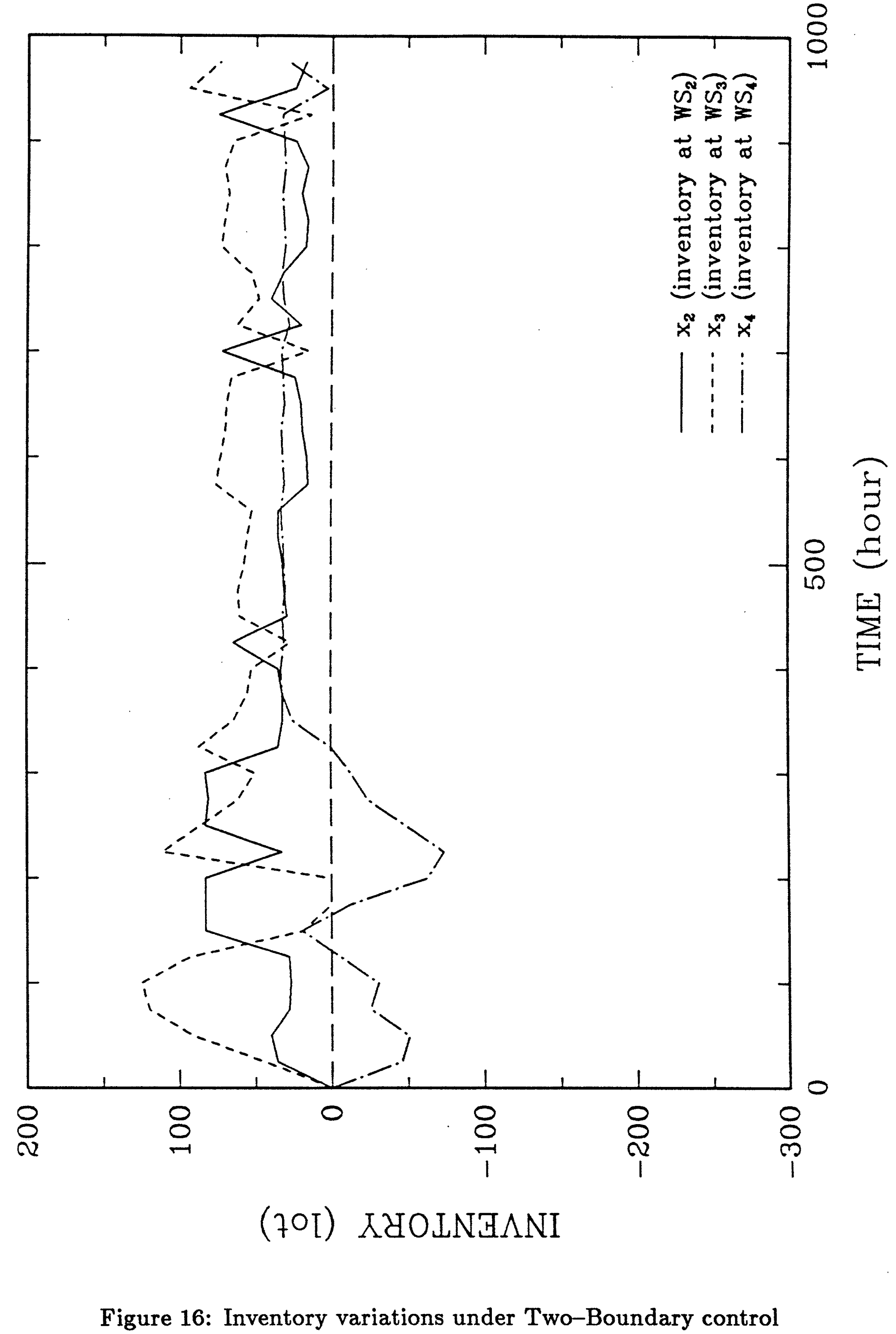


Figure 16: Inventory variations under Two–Boundary control

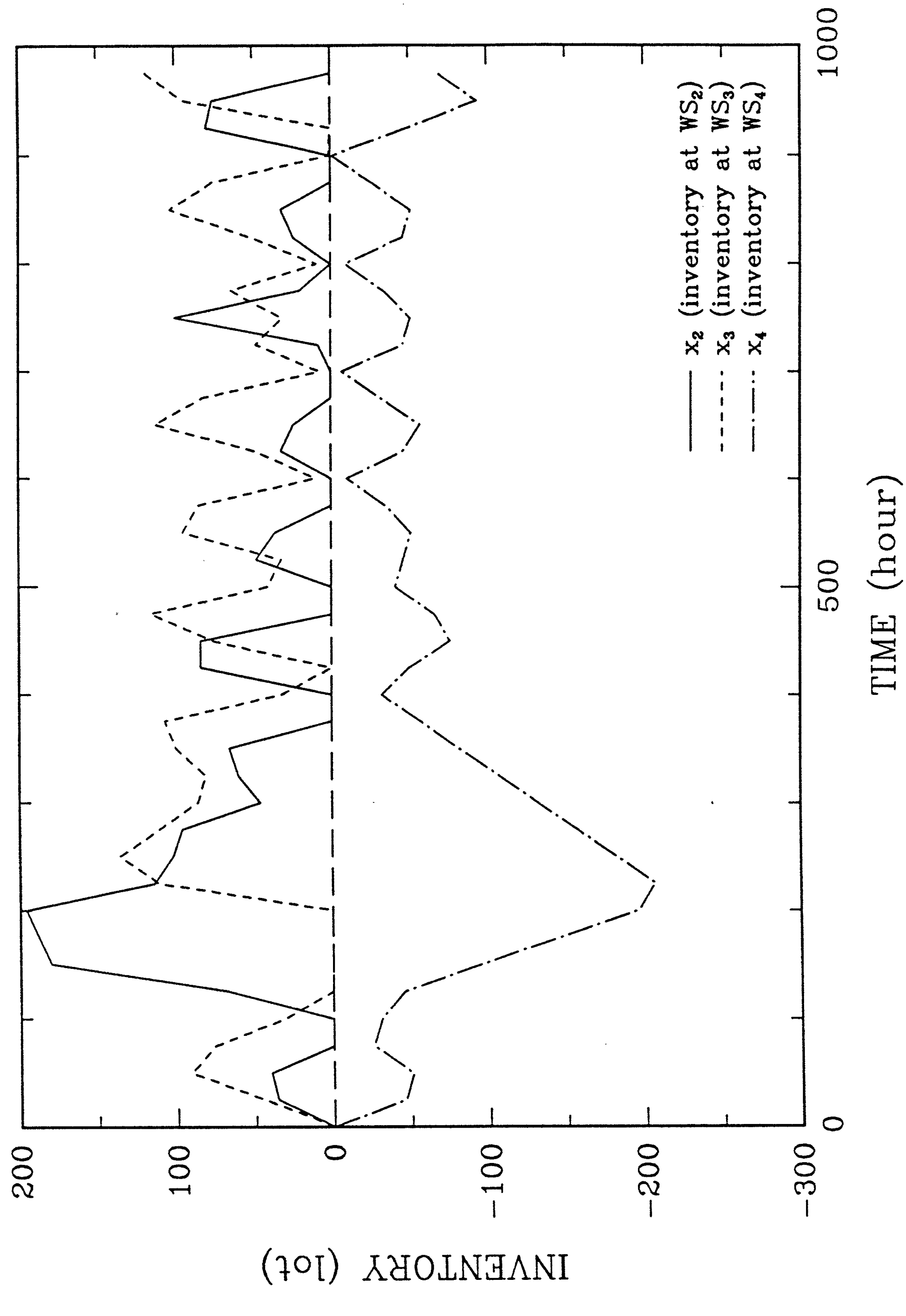


Figure 17: Inventory variations under Uniform–Loading

- In period 1, the behavior of the Local–Inventory control is very similar to that of the Two–Boundary control (Case 1, Fig. 16) when the surpluses are very small or negative. This is because, as we have just mentioned, in Case 1 only the local inventory hedging points are active. In Period 2 however, the Surplus hedging points become active that keep a relatively low WIP throughout the system while in Case 3 the same Local Inventory hedging points still maintain a higher WIP. Note, although reducing the Local Inventory hedging points in Case 3 will lower down the WIP in period 2, it will also slow down the catching up speed in period 1 and worsen the total behavior.

- In period 2, Case 1 and 4 behave similarly since only the Surplus hedging points are active. The difference occur only when the system is behind its schedule, *i.e.* with a negative x_4. We notice that, in period 1 of Case 4, x_2 and x_3 have larger peaks than that of Case 1 in the same period.

7 SUMMARY

In this paper we described the flow rate control model. Then, we computed the optimal control for a flow job shop. We combined our results into a single rule—the Two–Boundary–Control rule —which is sub-optimal for the flow job shop. The detailed analysis and an algorithm to compute the hedging points are presented in [23].

It should be pointed out that the control strategy and observations made in [21, 16] is closely related to the results shown in this paper. More specificly, as pointed out earlier, if the holding costs of the work stations prior to some *KEY* station, such as photolithography station in their example, are equal to or less than the cost at this key station, then they can be lumped together as a *SINGLE* station and the Two–Boundary control in this case is similar to what proposed in [21].

REFERENCES

1. R.Akella, Y.F.Choong, and S.B.Gershwin (1984), *Performance of Hierarchical Production Scheduling Policy,* IEEE Transactions on Components, Hybrids, and Manufacturing Technology, Vol. CHMT-7, No.3, September, 1984.

2. R. Akella and P.R. Kumar (1986), *Optimal control of production rate in*

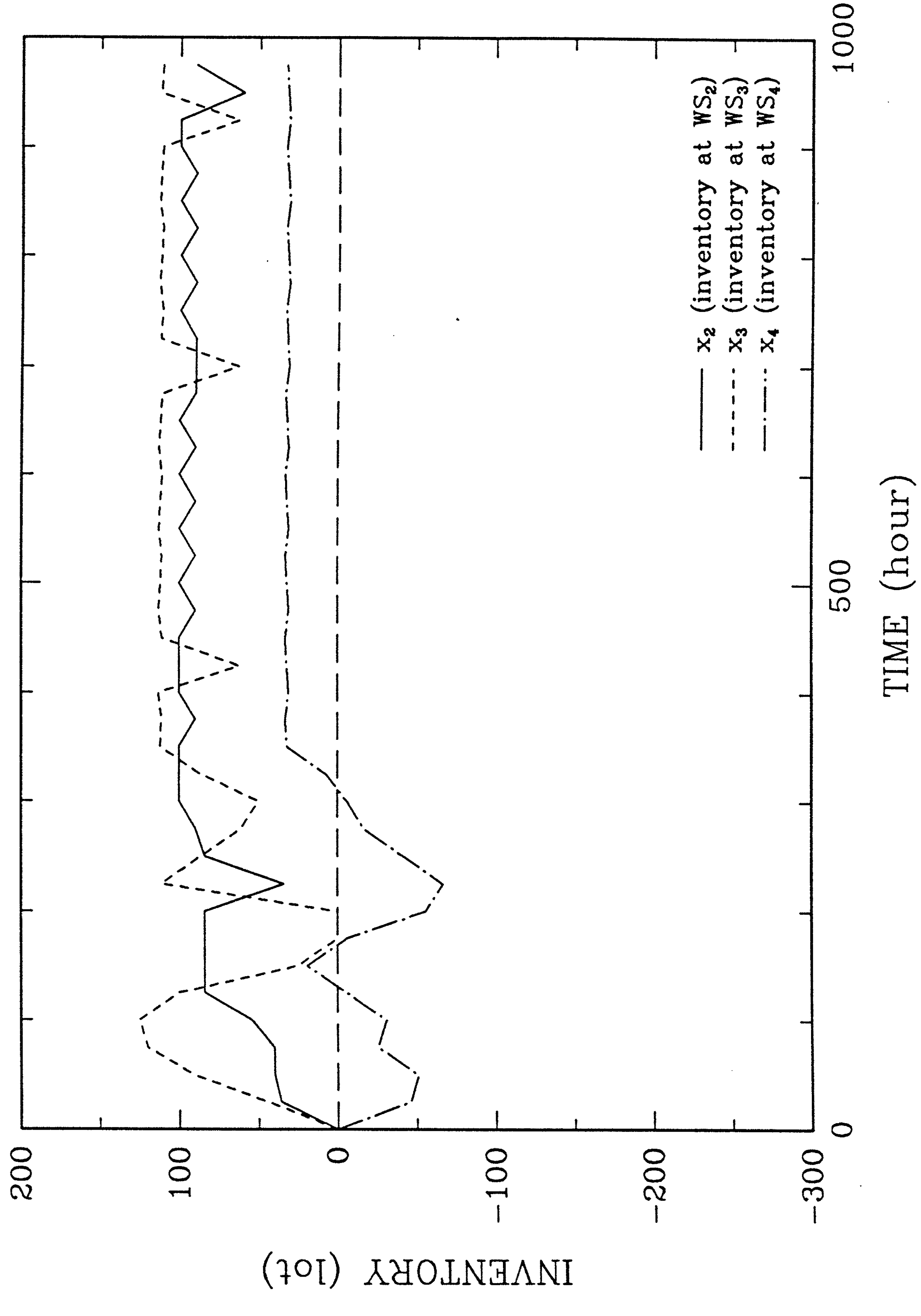


Figure 18: Inventory variations under Local–Inventory control

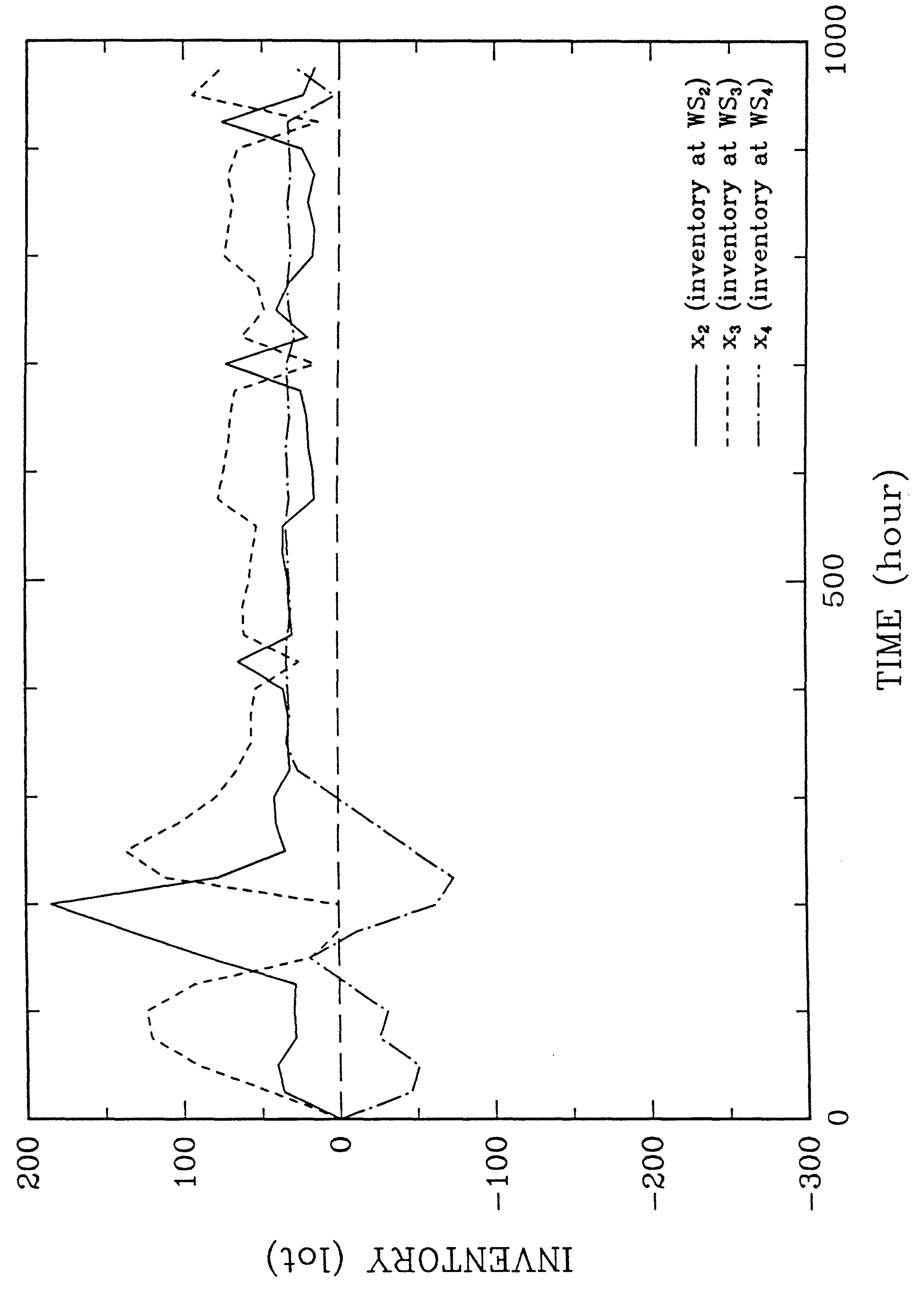


Figure 19: Inventory variations under Surplus control

failure prone manufacturing system, IEEE Trans. on Automatic Control, Vol. AC-31, No. 2, pp.116–126, February 1986.

3. D.P. Bertsekas, *Dynamic Programming:Deterministic and Stochastic Models,* Prentice-Hall, Inc. 1987.

4. T. Bielicki and P.R. Kumar (1987), *Optimality of zero–inventory policies for unreliable manufacturing systems,*

5. J. H. Blackstone Jr, D. T. Phillips and G. L. Hogg, *A State-of-the Art Survey of Dispatching Rules for Manufacturing Job Shop Operations*, Int. J. Prod. Res., 1982, Vol. 20, No. 1, 27-45.

6. R. W. Conway, W. L. Maxwell, L. W. Miller, Theory of Scheduling, Addison-Wesley, Reading, Mass., 1967.

7. S.E. Dreyfus and A.M.Law, *The Art and Theory of Dynamic Programming,* Academic Press, New York, 1977.

8. M. Fox, *Constraint-Directed Search: A Case Study of Job-Shop Scheduling,* Ph. D. Thesis, Carnegie-Mellon University, 1983.

9. S.B. Gershwin, R.Akella, and Y.F.Choong (1985), *Short-Term Production Scheduling of and Automated Manufacturing Facility,* IBM Journal of Research and Development, Vol. 29, No. 4, pp 392-400, July, 1985.

10. S.C.Graves, H.C.Meal, D.Stefek, A.H.Zeghmi (1983), *Scheduling of Re-Entrant Flow Shops,* Journal of Operations Management, Vol.3, No.4, August 1983, pp 197-207.

11. S. C. Graves, *A Tactical Planning Model for a Job Shop*, Working Paper, Alfred P. Sloan School of Management, MIT, 1985.

12. R.A. Howard, *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, Massachusetts, 1960.

13. J. Kimemia, S. B. Gershwin, *An Algorithm for the Computer Control of a Flexible Manufacturing System*, IIE Transactions, Vol. 15, No. 4, December 1983, pp. 353-362.

14. Rene de Koster, and Jacob Wijingaard, *Local and Integral control of Workload,* Report TUE/BDK/ORS/87/01, Eindhoven University of Technology, 1987.

15. E. W. Lawler, *Recent Results in the Theory of Machine Scheduling.*

16. R.C.Leachman and R. Glassey, *Preliminary Design and Development of a Corporate Level Production Planning System for the Semiconductor Industry*, IEEE International Conference on Robotics and Automation, Raleigh, April 1987.

17. J. K. Lenstra, *Sequencing by Enumerative Methods*, Mathematical Centre Tract 69, Mathematisch Centrum, Amsterdam, 1977.

18. S.X.C. Lou, *Job Shop Scheduling Using Flow Rate Control*, Proceedings of International Computers in Engineering Conference, New York N.Y., June 1987.

19. S.X.C. Lou, G. Van Ryzin, and S. B. Gershwin, *Scheduling Job Shops with delays*, Proceedings of the IEEE International Conf. on Robotics and Automation, Raleigh, North Carolina, March, 1987.

20. S.C.X. Lou, *An Event-Driven Job Shop Production Simulator,* LIDS Report, MIT, Cambridge, Dec., 1987.

21. C.Roger Glassey and Mauricio G. C. Resende, *Closed-Loop Release Control for Semiconductor Wafer Manufacturing.* DARPA/SRC Workshop on CIM for Integrated Circuits, Boston, June 1987.

22. G.Van Ryzin, *Control of Manufacturing Systems With Delays* M.S. Thesis, Laboratory for Information and Decision Systems, MIT, 1987.

23. G.Van Ryzin and Sheldon X.C. Lou, *Flow Rate Control Approach to Job Shop Scheduling*, Paper under preparation.