Chapter 2

Modeling Feasibility and Dynamics

> That man is prudent who neither hopes nor fears anything from the uncertain events of the future.
> – *Anatole France*

As was illustrated in our News Mix example in Chap. 1, it is not straightforward to pass from a deterministic to a stochastic formulation. We need to rethink the whole model, very often by changing both variables and constraints. Although many reformulations may make sense mathematically, they may in fact be rather peculiar in terms of interpretations. The purpose of this section is to discuss some of these issues, partly in terms of examples. The goal is not to declare some formulations generally superior to others, but rather to help you think carefully about how you rewrite your problems in light of uncertainty.

2.1 The Knapsack Problem

As an example, let us look at the knapsack problem. The problem is simple to write down:

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^{n} c_i x_i \\ &\text{such that} \quad \begin{cases} \sum_{i=1}^{n} w_i x_i \leq b, \\ x_i \in \{0,1\} \ , \quad i = 1, \ldots, n, \end{cases} \end{aligned} \tag{2.1}$$

where

c_i is the value of item i
w_i its weight and
b is the capacity of the knapsack

A.J. King and S.W. Wallace, *Modeling with Stochastic Programming*,
Springer Series in ORFE, DOI 10.1007/978-0-387-87817-1_2,

The goal is to fill the knapsack with as many valuable items as possible, but without exceeding the weight limit. Of course, w_i might also be viewed as "size," in which case the volume of the knapsack is the capacity in question.

Assume now that the weights are uncertain, so that, in fact, we are facing a vector of random variables $[\boldsymbol{w_1}, \ldots, \boldsymbol{w_n}]$. How are we to interpret this situation? The first question to be asked is always:

What is the inherent stage structure, and how many stages are there?

A clear clue to the stage structure is *when will we learn the weight of an item*? Obvious suggestions are:

1. We learn the weight *of each item just before* we decide whether or not to put it into the knapsack.
2. We learn the weight *of each item just after* putting it into the knapsack.
3. We learn the weight *of the full set of items just after* we decide what items to put into the knapsack.

The first two interpretations can lead to both *inherently two-stage problems* and *inherently multistage problems* with as many stages as there are items. The last interpretation will normally lead to an inherently two-stage formulation, in that we first decide which items to put in and only thereafter observe if they in fact fit. An additional aspect of stage structure is how potential infeasibilities are handled. After all, even though weights are uncertain, the capacity of the knapsack is fixed.

2.1.1 Feasibility in the Inherently Two-Stage Knapsack Problem

Let us list some potential ways of handling these stage-structure questions. For the moment we limit our discussion to the inherently two-stage cases where all items are picked (or listed in a specific order) before we learn anything about their weights.

1. We may require that the chosen set of items must always fit in the knapsack.
2. We may list the items in a certain order and pick them up until we come to one that does not fit. Then we stop. (So the decision is the list.)
3. We may do as described above, but if a later item fits (as it is light enough), then we take it.
4. We may list the items and keep adding items until we have added one that does not fit. We then pay a penalty for the overweight.
5. We may pick a set of items such that if the items do not fit in the knapsack after we have learned their weights, we pay a penalty for the total overweight.
6. We may pick a set of items of maximal value so that the probability that the items will not fit in the knapsack is below a certain level.

There are certainly more variations, but let us stop here for now. You should think about what these cases imply before reading on. One way to structure the analysis is to ask a central question: When do we learn the weights of the items?

In Case 2 above, we list the items and stop putting them into the knapsack when we find one that does not fit. This implies that we learn the weight of an item *before* it is actually put into the knapsack. Case 3 is a variant of Case 2 since it amounts to stopping when an item does not fit and then continuing down the list until we find ones that do fit. Case 4 implies that we need to actually put the item into the knapsack before observing its weight. So the second and fourth cases represent quite different interpretations of when we learn the weights. In Case 5 we learn the weights after we have decided on the selection of items.

Note that putting items into the knapsack is a very passive action in these cases since we have already decided on the order in which items will be picked up (for Case 5 we simply pick them all up). If we want the order to depend on the actual observed sizes, then we end up with an inherently multistage formulation, which we will discuss a bit later.

Cases 2, 3, and 4 result in inherently two-stage models since we define the lists before we start putting items into the knapsack. Stage 1 is to find the list of items, whereas Stage 2 is to passively put them into the knapsack until the stopping rules are satisfied. But to set up the lists, you must anticipate the different situations that can occur. Hence the models will be multistage with an inherently two-stage structure.

Case 5 is also an inherently two-stage formulation, leading to an inherently two-stage model. The model will have only two actual stages as there is no question of ordering the items.

Case 6 results in a chance-constrained formulation that we will discuss shortly.

Case 1, requiring the chosen set of items to always fit in the knapsack, corresponds to a worst-case analysis. Since we need to find a set of items that always fit in the knapsack, we can replace the random variables $\boldsymbol{w_i}$ by their maximal values. Of course, this formulation makes sense only if there is an upper bound on the weights.

The worst-case analysis corresponds to a very “pessimistic” view of the world: we can *never* accept overweight. Whether or not this is reasonable is a *modeling* question. We must look at the situation in which the model is being used and ask ourselves if it is really the case that we cannot handle an overweight item.

If we plan to put an item into the knapsack, is there nothing we can do to get it to “fit”? If the knapsack is a truck, and the items are the loads we plan to send, could we not send some items with the next truck? Maybe we could put a package in the passenger seat? Maybe we could send it by mail?

Requiring feasibility in this way is extremely strong, and we must be sure we really wish to imply a worst-case situation.

Finally, of course, our estimates of the maximal weights may be incorrect. This may lead to an actual situation with overweight, even if the model said it would not happen! Then what will we do? Will the world end? Will the company go broke for sure? Probably not. But if we *can* handle overweight when it really happens, then why did we model the problem as if it could not be allowed to happen under any circumstances? You should really be able to answer these questions if you wish to use worst-case analysis.

2.1.2 Two-Stage Models

So some of the models, while being inherently two-stage, are multistage in nature. Those that are not are the worst-case analysis (which is always particularly risky to use if the worst case is not well defined) and the last two cases—the one with a penalty for total overweight and the one looking at the probability of overweight. Let us first look at a penalty case.

Let $\mathcal{S}$ be the set of scenarios describing the uncertainty in an appropriate way. Then we obtain

$$\begin{array}{ll} \max & \displaystyle\sum_{i=1}^{n} c_i x_i - d \sum_{s \in \mathcal{S}} p^s z^s \\ \text{such that} & \left\{ \begin{array}{rl} \sum_{i=1}^{n} w_i^s x_i - z^s \leq b, & \forall s \in \mathcal{S}, \\ z^s \geq 0, & \forall s \in \mathcal{S}, \\ x_i \in \{0,1\}, & 1 = 1, \ldots, n, \end{array} \right. \end{array} \tag{2.2}$$

where d is the unit penalty for overweight. A more general penalty could be a function $f(z^s)$ describing a nonlinear dependence on the total overweight. This model might be good or bad, depending on how well it describes our case. It has, however, a clear interpretation, as it has a clear information structure (we learn about the weights after having decided which items to use), it has a clear description of the goal (to maximize the value of the items selected minus the expected penalty for overweight), and it states what happens if we get it wrong—we pay a penalty. The penalty may mean exactly that, a financial penalty for being wrong. But it may also mean a cost for sending an item with a later truck, the extra cost of using a competitor, or possibly a rejection cost.

This formulation can be viewed as replacing a constraint with a penalty since it can be written as

$$\max_{x_i \in \{0,1\}} \quad \sum_{i=1}^{n} c_i x_i - d \sum_{s \in \mathcal{S}} p^s \left[\sum_{i=1}^{n} w_i^s x_i - b \right]_+ , \tag{2.3}$$

where $[x]_+$ is equal to x if $x \geq 0$, and zero otherwise. We call this a *penalty formulation*.

Case 1 in our listing, the worst-case analysis, can be formulated as ensuring that each item's maximal weight $w^{\max}$ will fit into the knapsack:

$$\begin{array}{ll} \max\limits_{x} & \sum\limits_{i=1}^{n} c_i x_i \\ \text{such that} & \begin{cases} \sum_{i=1}^{n} w_i^{\max} x_i \leq b, & \\ \qquad x_i \in \{0,1\}, & 1 = 1, \ldots, n. \end{cases} \end{array}$$

There is not much to say about this one. It is very pessimistic, and the model is, technically speaking, deterministic. Of course, in some cases, this is exactly what we need, so the model may be appropriate. Note, however, as mentioned above, that this is a very sensitive model unless $w^{\max}$ is well understood. Therefore, although mathematically well defined, this model may be pessimistic *and* risky at the same time. So in general, this model is hard to defend. On the one hand, we claim that the items *must* fit in the knapsack; on the other hand, we risk that they do not unless we know $w^{\max}$ precisely. The average behavior of the solution coming from this model might be bad, as we do not attempt to control it.

2.1.3 Chance-Constrained Models

Let us pass to Case 6 in our listing, a model that tries to get around the problem of feasibility by requiring that the items fit in the knapsack with a certain probability. The standard model in this case within stochastic programming is a chance-constrained model. It would take the following form:

$$\begin{array}{ll} \max\limits_{x_i \in \{0,1\}} & \sum\limits_{i=1}^{n} c_i x_i \\ \text{such that} & \begin{cases} \sum_{s \in W(x)} p^s \geq \alpha, \\ W(x) = \{s : \sum_{i=1}^{n} w_i^s x_i \leq b\}, \end{cases} \end{array} \tag{2.4}$$

where α is the required probability of feasibility. This model is clear with respect to the objective function and how to treat infeasibilities.

Chance-constrained models say nothing about what happens if we have overweight. In a sense, a chance-constrained problem is a slight relaxation of a worst-case analysis. In the truck example, this model means that we plan which items to put on the truck and we require that our plans work out α

percent of the time. What we do when the items do not fit is not clear. Perhaps we ship them off on another slower channel, and the probability level is in fact a service level for a quick transfer.

If the monetary cost is reasonably well connected to α, we also have controlled the costs. But a danger with this formulation is that the costs associated with lack of feasibility may be connected to the size of the violation, not just the probability, and a chance-constrained model does not have any control over this aspect of the solution.

2.1.4 Stochastic Robust Formulations

Chance-constrained problems (particularly with discrete variables) can be hard to solve. That is especially so for problems more complicated than what we discuss here. Stochastic robust optimization (discussed in Sect. 1.8.3.1) offers an alternative. The worst-case analysis discussed above is a type of robust formulation—we are looking for the most profitable solution that is always feasible. However, there are more sophisticated formulations with a trade-off between loss in income and increase in probability of feasibility. Also in these models we totally disregard what lack of feasibility actually costs. Let us write the model in a way that is reasonably easy to understand, although this is not the format we would use to solve it. Let there be N items available, such that item i has a weight coming from a symmetric distribution over $[w_i - \hat{w}_i, w_i + \hat{w}_i]$, and let Γ be an integer between 1 and N. Let $\mathcal{N} = \{1, 2, \ldots, N\}$. The following formulation will give us the set of items with maximal value under the constraint that if at most Γ of the random weights work against us, we still have a feasible solution, i.e., we can still fit the items in the knapsack, whatever values these weights take. The general probability of feasibility is also high. Bounds are complicated but are given in the underlying paper by Bertsimas and Sim [5]:

$$\max_{x \in \{0,1\}} \quad \sum_{i=1}^{n} c_i x_i$$

$$\text{such that} \quad \begin{cases} \sum_{i=1}^{n} w_i x_i + \psi(x, \Gamma) \leq b, \\ \qquad\qquad \psi(x, \Gamma) = \max\limits_{S \subset \mathcal{N}, |S| = \Gamma} \sum\limits_{i \in \mathcal{S}} \hat{w}_i x_i. \end{cases}$$

As we increase Γ, we get closer and closer to a worst-case situation. Also in this model, there is no statement about what actually happens when items do not fit. As before, that might or might not be a problem.

A major reason these robust models do not control average profits is, of course, that they do not use probabilities at all, just supports of the random variables (or, generally, intervals over which we wish to be protected). In its own right that may be good, but it certainly carries with it potential surprises when solutions are implemented. The combination of considering neither the costs of overweight nor the probabilities thereof is not without potential risks.

Note that the worst-case formulation given earlier and the preceding case with $\Gamma = N$ are not the same, as the x variables are not defined in exactly the same way.

2.1.5 Two Different Multistage Formulations

When making the knapsack problem stochastic, there are two different multistage settings; one is inherently two-stage, the other inherently multistage. In one case, we ask for a decision rule of the following type: items $i_1, i_2, \ldots, i_k$ have been added and the weights turned out to be $w_1, w_2, \ldots, w_k$, so which item should I put in next? This problem is inherently multistage. This is an extremely difficult problem if you require optimality.

The alternative multistage problem is as follows. Give me the (ordered) list of items and follow certain rules for stopping. Here we do not change our minds on the order as we fit them in based on observations, but we take uncertainty into account when setting up the list. This formulation is inherently two-stage. A good test for you is to formulate this latter problem as a decision-tree problem under the assumption that each item has only a limited number of possible weights. Try it!

2.2 Overhaul Project Example

This example is taken from Anderson et al., Sect. 10.4. Let us first repeat the problem and the analysis as given in the reference. We have an overhaul project with five activities, labeled A–E. The example is as given in Table 2.1. The activity-on-arc network for this little project is given in Fig. 2.1.

Table 2.1: Activity names, expected durations, and immediate predecessors

		Immediate	Expected duration
Activity	Description	predecessor	(days)
A	Overhaul machine I	–	7
B	Adjust machine I	A	3
C	Overhaul machine II	–	6
D	Adjust machine II	C	3
E	Test system	B,D	2

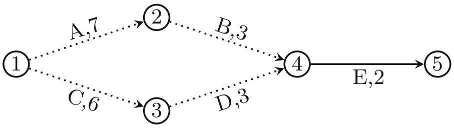

Fig. 2.1: Example network for overhaul example; each arc is labeled with its activity code and duration

The longest path through this network is given by the sequence of activities A, B, and E with a completion time of 12. This path is called the *critical path* as any delay on an activity on this path will delay the whole project. The partial sequence C–D has a slack of 1, indicating that if either activity (but not both!) is delayed by 1 day, the project completion will not be delayed.

Suppose that it has become evident that the overhaul project must be completed within 10 days. With the data presented in Table 2.1, this is not possible, and the company is willing to invest money to reduce the project duration. The company may reduce the durations of the activities for a cost. For each activity, the reduction costs and the maximum possible reductions are given in Table 2.2.

Table 2.2: Maximum possible reduction and corresponding costs

		Maximal	Cost
Activity	Description	reduction	per day
A	Overhaul machine I	3	100
B	Adjust machine I	1	150
C	Overhaul machine II	2	200
D	Adjust machine II	2	175
E	Test system	1	250

Let y_{A} denote the number of days by which we reduce the duration of activity A, which will cost $100y_{\mathrm{A}}$. With variables for the other activities similarly defined, the following linear program can be used to determine the minimum cost of reducing the project duration to 10 days, as required. The variable x_i denotes the time at which event i begins. For example, event 4 is the time when both activities D and B have finished and, hence, E is ready to start.

$$\begin{aligned} \min \quad & 100y_{\mathrm{A}} + 150y_{\mathrm{B}} + 200y_{\mathrm{C}} + 175y_{\mathrm{D}} + 250y_{\mathrm{E}} \\ \text{such that} \quad & \begin{cases} x_2 \geq x_1 + 7 - y_{\mathrm{A}}, \\ x_3 \geq x_1 + 6 - y_{\mathrm{C}}, \\ x_4 \geq x_2 + 3 - y_{\mathrm{B}}, \\ x_4 \geq x_3 + 3 - y_{\mathrm{D}}, \\ x_5 \geq x_4 + 2 - y_{\mathrm{E}}, \\ x_1 = 0, \\ x_5 \leq 10, \\ y_{\mathrm{A}} \leq 3, \\ y_{\mathrm{B}} \leq 1, \\ y_{\mathrm{C}} \leq 2, \\ y_{\mathrm{D}} \leq 2, \\ y_{\mathrm{E}} \leq 1, \\ x, y \geq 0. \end{cases} \end{aligned} \tag{2.5}$$

The minimal investment necessary to complete the project within 10 days is 350 and is obtained by setting $y_{\mathrm{A}} = 1$ and $y_{\mathrm{E}} = 1$ with all other investments zero. All activities are now on a critical path. This is a typical result from this type of model. Our investment results in a large number of critical paths, and the extreme case is what we observed here: all activities sit on critical paths. The new network is given in Fig. 2.2.

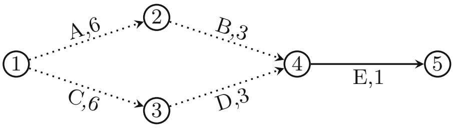

Fig. 2.2: Example network after investments in activity durations; all paths are critical

2.2.1 Analysis

How are we to interpret this result? At a cost of 350 we have reduced the project duration to 10, as required, and all activities have become critical. What does that mean? Does it mean that any delay in any activity will cause a project delay? The answer depends on the nature of the data presented in Tables 2.1 and 2.2.

Suppose that the activity durations given in Figs. 2.1 (before investments) and 2.2 (after investments) are not really deterministic. Although many possibilities exist, let us assume that the given numbers are expected durations and that the distributions are independent and as indicated in Table 2.3. That is, suppose that with the exception of activity E, all activity durations will be their expected values, plus one of three values, $\{-1, 0, 1\}$, with all values being equally likely. Activity E is assumed to have a deterministic duration.

Table 2.3: Probability distribution for activity durations relative to mean

		Probability of deviation from expected value		
Activity	Description	-1	0	$+1$
A	Overhaul machine I	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
B	Adjust machine I	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
C	Overhaul machine II	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
D	Adjust machine II	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
E	Test system	0	1	0

Given that activity durations are random variables, the project duration is also a random variable. Since this is such an easy network, we can calculate the distribution of the project duration exactly. We begin by calculating the distribution of the duration of activities A and B together and then do the same for C and D. Event 4 will then take place when the last of (A and B) and (C and D) is finished. Finally, we add the duration of E.

Since the duration of A (after investments) is 5, 6, or 7, while that of B is 2, 3, or 4, activities A *and* B will have finished after 7, 8, 9, 10, or 11 days. The distribution can be found by examining all possible combinations of the completion times:

Duration of A	5	5	5	6	6	6	7	7	7
Duration of B	2	3	4	2	3	4	2	3	4
Duration of A *and* B	7	8	9	8	9	10	9	10	11

Each of these events occurs with probability $\frac{1}{3} \times \frac{1}{3} = \frac{1}{9}$. Hence, the distribution for the duration of A *and* B is given by

Duration of A *and* B	7	8	9	10	11
Probability	$\frac{1}{9}$	$\frac{2}{9}$	$\frac{3}{9}$	$\frac{2}{9}$	$\frac{1}{9}$

This is also the duration of C *and* D. Event 4, which corresponds to the start of activity E, occurs when all of the first four activities have finished. Formally, we may state that

$$\text{Start time for event 4} = \max\{\text{Duration of A and B, Duration of C and D}\}$$

To calculate the distribution of this maximization, we simply look at all 25 combinations of durations of (A and B) and (C and D). The project duration is simply the start time for event 4, plus the duration of activity E (which is 1 day). Thus, we obtain the following distribution for the duration of the project:

Duration of project	8	9	10	11	12
Probability	$\frac{1}{81}$	$\frac{8}{81}$	$\frac{27}{81}$	$\frac{28}{81}$	$\frac{17}{81}$

The expected project duration is therefore 10.642, well above the required duration of 10. In fact, there is a probability of 56 % that the project will take longer than 10. Hence, it seems that our investment of 350 has not brought the duration down to 10. It has not even brought the expected duration down to 10.

At this point you should worry about the sequencing of decisions and about the information that is available as individual decisions are made. Our initial analysis assumed that all activities ended up with their expected duration and that all decisions were made initially. That is, our initial analysis focused exclusively on one scenario that had a probability of $\frac{1}{81}$ of occurring.

2.2.2 A Two-Stage Version

In the preceding discussion, we solved a problem where all activities were assumed to have an average duration in order to find a possible investment. Thereafter, we checked how this solution/investment would behave in a random environment. But when searching for the solution, we did not consider the uncertainty in the activity durations. As a result, one of the effects was that we did not even achieve an expected project duration of 10.

Let us now reconsider the investment, this time recognizing the uncertainty in the activity durations. Suppose that the investment must be determined before the actual durations can be known. This is inherently two-stage. Let d^s_{A} be the duration of activity A in scenario s, and let the other durations be defined accordingly. Since the duration of activity E is not subject to uncertainty, we do not define such an entity for this activity. Let the investment variables, y, be defined as before, and let x^s_i be the time of event i if scenario s occurs. We have now $3^4 = 81$ equally likely scenarios corresponding to the 81 possible realizations of durations of activities A–D.

A question that occurs at this time is what we are to mean by the project taking 10 days. Is it going to be an average of 10 days, or is 10 days a hard constraint? Or maybe 10 days is the goal, but, at a penalty, we are allowed to be late? In reality, of course, this type of constraint is not hard. We can never guarantee that a project cannot be late. We could certainly find an investment that, with our 81 scenarios, guaranteed that we were never late, but reality will always be different. Hence, let us instead assume that if we are late, a penalty of 275 per day is incurred.

Note first that if we had added the possibility of being late at a penalty of 275 to the deterministic problem, the solution would not have changed, as it is cheaper to invest in reducing activity durations than to pay the penalties. Also, note that the expected cost associated with the deterministic solution is now the initial investment of 350 plus an expected penalty for being late of 210, at a total of 560. So the initial cost estimate of 350 was far off the actual cost.

The following two-stage model will minimize the expected cost of achieving a project duration of 10, *provided all investment decisions are made before the project starts and we are allowed to be late.* Lateness in scenario s is measured by t^s.

$$\min \quad 100y_{\mathrm{A}} + 150y_{\mathrm{B}} + 200y_{\mathrm{C}} + 175y_{\mathrm{D}} + 250y_{\mathrm{E}} + \frac{275}{81}\sum_{s=1}^{81} t^s$$

$$\text{such that} \quad \left\{ \begin{array}{rcll} x_2^s & \geq & x_1^s + d_{\mathrm{A}}^s - y_{\mathrm{A}} & \text{for all } s, \\ x_3^s & \geq & x_1^s + d_{\mathrm{C}}^s - y_{\mathrm{C}} & \text{for all } s, \\ x_4^s & \geq & x_2^s + d_{\mathrm{B}}^s - y_{\mathrm{B}} & \text{for all } s, \\ x_4^s & \geq & x_3^s + d_{\mathrm{D}}^s - y_{\mathrm{D}} & \text{for all } s, \\ x_5^s & \geq & x_4^s + 2 - y_{\mathrm{E}} & \text{for all } s, \\ x_1^s & = & 0 & \text{for all } s \\ x_5^s - t^s & \leq & 10 & \text{for all } s, \\ y_{\mathrm{A}} & \leq & 3, & \\ y_{\mathrm{B}} & \leq & 1, & \\ y_{\mathrm{C}} & \leq & 2, & \\ y_{\mathrm{D}} & \leq & 2, & \\ y_{\mathrm{E}} & \leq & 1, & \\ x, y & \geq & 0, & \\ t^s & \geq & 0 & \text{for all } s. \end{array} \right. \tag{2.6}$$

What distinguishes Problem (2.6) from Problem (2.5) is found in the representation of the constraint on the project duration. In (2.5), this was represented by

$$x_5 \leq 10.$$

In (2.6), we have

$$x_5^s - t^s \leq 10,$$

indicating that the duration should be at most 10 but can be allowed to be longer. Furthermore, all constraints that are used to calculate the event times, such as

$$x_2^s \geq x_1^s + 7 - y_\mathrm{A},$$

are now indexed by scenario. Solving this we find $y_\mathrm{A} = 1$ as the only investment. The cost is 100. The expected penalty for delays is as high as 455, yielding a total of 555, a reduction of 5 from the expected value of the deterministic solution. (As this is just an artificial example, the numbers themselves are not important; the main point is that we obtain a different and cheaper solution.)

What happened here is that as we realized explicitly that the world was stochastic and that delays were in fact feasible (at a cost), we ended up investing much less initially to reduce the project duration. Instead, we preferred to pay the penalty for being late. With $y_\mathrm{A} = 1$ as the only investment, there is a probability that event 4 will take place later than time 8 (so that we finish the whole project later than 10) of 89 %. Hence, the penalty is incurred in 89 % of the cases. As 89 % of 275 equals less than 250 (the unit investment cost in activity E), we prefer the penalty cost. So in this case, there was flexibility in *waiting*. Instead of securing the project duration initially, it was better to wait and see what happened.

This formulation is an inherently two-stage formulation, leading to a two-stage model, as we first make investments, then observe the activity durations, and finally (Stage 2) calculate the project duration and delay costs.

2.2.3 A Different Inherently Two-Stage Formulation

The ideal formulation of this project scheduling problem is to take into account the actual float of information. Initially, we must decide on y_A and y_C. Then, y_B is decided when activity A is finished and y_D when activity C is finished. However, we do not know which of these events will occur first. Modelingwise, this creates a lot of difficulty if we are to formulate the problem as a stochastic programming problem. It makes us unable to define stages. Stage 1 is to define y_A and y_C, but what is Stage 2? It is to define y_B if A finishes before C, but to define y_D if B finishes before A. And which of these will happen first depends on both the randomness and our first-stage decisions. Stage 4 is in any case to determine y_E.

Hence, let us analyze a somewhat simpler case. Let us assume that investments in activities A–D must be determined initially, but that activity E can wait until the activity is to start. This will make y_{E} scenario dependent. In addition, we can be late at a penalty.

$$\min \quad 100y_{\mathrm{A}} + 150y_{\mathrm{B}} + 200y_{\mathrm{C}} + 175y_{\mathrm{D}} + \frac{250}{81}\sum_{s=1}^{81} y_{\mathrm{E}}^s + \frac{275}{81}\sum_{s=1}^{81} t^s$$

$$\text{such that} \quad \left\{\begin{array}{rcl} x_2^s \geq x_1^s + d_{\mathrm{A}}^s - y_{\mathrm{A}} & & \text{for all } s, \\ x_3^s \geq x_1^s + d_{\mathrm{C}}^s - y_{\mathrm{C}} & & \text{for all } s, \\ x_4^s \geq x_2^s + d_{\mathrm{B}}^s - y_{\mathrm{B}} & & \text{for all } s, \\ x_4^s \geq x_3^s + d_{\mathrm{D}}^s - y_{\mathrm{D}} & & \text{for all } s, \\ x_5^s \geq x_4^s + 2 - y_{\mathrm{E}}^s & & \text{for all } s, \\ x_1^s & = 0 & \text{for all } s, \\ x_5^s - t^s & \leq 10 & \text{for all } s, \\ y_{\mathrm{A}} & \leq 3, & \\ y_{\mathrm{B}} & \leq 1, & \\ y_{\mathrm{C}} & \leq 2, & \\ y_{\mathrm{D}} & \leq 2, & \\ y_{\mathrm{E}}^s & \leq 1 & \text{for all } s, \\ x, y & \geq 0, & \\ t^s & \geq 0 & \text{for all } s. \end{array}\right. \tag{2.7}$$

Based on what we have learned so far, it is a challenge to guess what the solution will be. Without calculations, you should be able to see that the investment will be $y_{\mathrm{A}} = 1$ as the only investment. Furthermore, since investments in E can be delayed until we are ready to start the activity, we will choose to invest in E if we start later than time 8. This is so since the cost of investing in E is lower than the penalty cost of being late. If we are ready to start activity E later than time 9, we will invest in a one-unit decrease in E and take the rest of the delay as a penalty. The total expected cost is down to 533.

2.2.4 Worst-Case Analysis

An obvious way to make sure the duration is at most 10 (in fact, the only way) is to perform a worst-case analysis. We then resolve (2.5), but with maximal durations rather than average durations. Hence, the durations of activities

A, B, C, and D will increase by one. The result will be $y_\mathrm{A} = 3$, $y_\mathrm{B} = 0$, $y_\mathrm{C} = 0$, $y_\mathrm{D} = 2$, and $y_\mathrm{E} = 1$, with a total cost of 900. Of course, in this case the expected delay cost is zero. But be aware that this is assured only if our description of uncertainty is correct. Hence, worst-case analysis can be both conservative and risky at the same time; we are careful, pay a lot to be *sure* that we are always feasible, but then, due to errors in estimating the data, we are not so sure after all. Worst-case solutions do not handle measurement errors.

2.2.5 A Comparison

There are a few points to be made here. The first concerns feasibility. In the deterministic model, it was reasonable and meaningful to say that we had to finish in ten time periods. But if we kept that requirement in the stochastic setting, we were brought to a worst-case analysis. If we have an inherently two-stage model (i.e., all investments are made before the project is started), the only way to guarantee a duration of ten time periods is to plan as if everything were against you. We saw that the cost would be 900. Very often, such strict interpretations of feasibility are not reasonable. Instead, it is necessary to ask if a constraint is *really* hard? Very often the answer is no. If the softness that is therefore brought into the model can be described by penalties, then we end up with a *recourse* model. We gave two examples of such models. In a two-stage setting with penalties, we ended up with expected costs down from 900 to 555. If, in addition, we allowed the second stage to also contain a genuine investment, the expected cost dropped to 533. The latter drop is simply caused by the new investment opportunity's being cheaper than the delay cost. If we solved the deterministic model, it claimed that the investment cost would be 350, whereas, in fact, the total expected cost was 560. With a strict interpretation of feasibility, the deterministic solution was infeasible with a probability of 0.56.

In addition to the issue of feasibility, we observe that there is again a value for delaying decisions. We saw that the total expected cost depended on how we defined these possible delays. We cannot say that one model is better than the other unless we actually know the real decision context. But we see how the modeling choices affect decisions and costs.

2.2.6 Dependent Random Variables

In Sect. 2.2.1, we assumed that all the random durations were independent. We then found that using the investment from the deterministic model $y_\mathrm{A} = y_\mathrm{E} = 1$, the expected cost was not 350, as indicated by the deterministic model, but rather 560 if the late penalty were set at 275 per day. The probability of being late was as high as 56 %. If the durations had instead been correlated, the deterministic model would have been the same, hence the investments would have been the same, but the expected costs would have been different, and so would the probability of delays.

Let us see what would happen if activities A and B were perfectly negatively correlated and activities C and D perfectly positively correlated. These are, of course, extreme assumptions, but they serve to illustrate a few points.

First, the duration of A and B would be deterministically equal to 9. The perfect negative correlation (correlation coefficient of -1) has removed the uncertainty on that path. As always, a negative correlation has helped us control the variation. A negative correlation of -1 is, of course, rather special (as it implies we have only one random variable, not two), but any negative correlation between the durations of A and B would reduce overall uncertainty and be useful to us.

> Negative correlations are always potentially useful, and you should think carefully about how they might help you.

For the other path, a perfect positive correlation would cause the duration of C and D to be 7, 9, or 11, each with a probability of $\frac{1}{3}$. This is worse than in the deterministic case in the sense that while the probability of being above 9 (causing the project to have a duration above 10) has stayed at $\frac{1}{3}$, the expected delay, given that there is a delay, has increased. Here we see that a positive correlation has caused us trouble, as it normally does.

So the starting time for event 4 is again the maximum of the duration of these two paths plus one, leading to the following distribution for the project duration:

Project duration	10	12
Probability	$\frac{2}{3}$	$\frac{1}{3}$

So the probability of being late is now 33 % and the expected project duration 10.67. The expected cost is now 533, down from 560.

The point of this is to understand that there is not just the question of stochastic/deterministic but also *which* stochastic model we are facing. Correlations (and other measures of covariation) have no counterparts in deterministic modeling, making several rather different stochastic settings being represented by the same deterministic model.

So the question is not that one type of stochastics (like the uncorrelated durations we started out with) is “better” than others. Rather, it is that the effects of solving a deterministic model depends on the stochastic setting. We saw that when we changed from uncorrelated random variables to correlated ones (in this one specific way), the expected costs went down, the expected duration went up, and the probability of delay did not change. Again, this is not good or bad; it is simply an observation telling us that issues that do not even come up in deterministic modeling, such as covariation, can be important in valuing the actual performance of a deterministic model.

2.2.7 Using Sensitivity Analysis Correctly

In this example, we assumed that there was a penalty cost of 275 per day if we were late. We simply used it as a number. Is that appropriate in the light of the discussions in this book? You may want to think about that question for a few minutes!

If the number is a specific estimate of the cost of delay based on market activities, like a contractual penalty, *and nothing else*, such as lost image, we may face two situations:

- If this is an estimate of what the future value of the penalty will be, based on its present value, then it should be treated as a random variable, and our approach is not valid in light of our own discussions: we have used expected values instead of the actual distribution.
- If this is a known entity, which we know will not change during the life of the project, then our approach is indeed valid.

But most likely, even if there is a contractual penalty for lateness, there will also be a question of lost goodwill, lost reputation. And the size of that loss is not really known; it is anybody's guess. So 275 is our guess, our chosen value for the overall costs. But it is a guess, a choice, not because we are facing a random variable but because it is up to us to define the penalty: the penalty is in its own right a policy parameter for the company. If that is the case, our approach is appropriate, but it should possibly be accompanied by a parametric analysis on the level of the penalty.

So the right answer depends on the setting. However, most likely, this is a case where parametric analysis is appropriate because the penalty is like a decision variable: it is up to us to set it. We leave it to you to check what would happen if the penalty were slightly different from 275 (in particular a bit higher).

2.3 An Inventory Problem

Let us turn to another very classical model. We are responsible for a production and inventory system where for the next T periods we know the demand. For practical reasons, we do not like production to vary too much from one period to the next, so we have defined an upper bound on changes in production levels ℓ. We are in a setting where demand must be met, but it can be met from outside sources (which means, technically speaking, that we allow demand to be rejected *inside* the model). We have formulated the following inherently multistage production and inventory model for our situation, not taking uncertainty in demand into account.

$$
\begin{array}{ll}
\min & \displaystyle\sum_{t=1}^{T} (c_t x_t + f_t I_t + b_t u_t) \\
\text{such that} & \left\{ \begin{array}{lll} I_t - I_{t-1} = x_t - d_t + u_t, & & t = 1, \ldots, T, \\ |x_t - x_{t-1}| & \leq \ell, & t = 2, \ldots, T, \\ x_t, u_t, I_t & \geq 0 & t = 1, \ldots, T, \end{array} \right.
\end{array}
$$

with I_0 given. Here I_t is the inventory at the end of period t, x_t the production of period t (determined at the start of the period), c_t the production costs, f_t the unit inventory costs, and d_t the demand in period t. The variable u_t represents external orders or the impact of lost sales measured by b_t. Of course, there are other versions of this model, representing backorders, for example. But the basic model that requires production plus inventory to satisfy demand is common. The first inequality expresses our requirement that production must not change by more than ℓ from one period to the next.

Note that such an inequality can be written linearly as

$$
\begin{aligned}
x_t - x_{t-1} + w_t - z_t &= 0, \\
0 \leq w_t, z_t &\leq \ell.
\end{aligned}
$$

However, we will continue to use the absolute value formulation for easier reading.

2.3.1 Information Structure

The first question is always what is random and when do we learn the value of the random variables? Production costs, inventory costs, demand, and production volumes can all be random. For simplicity in this discussion we assume that the only relevant random variables are the demands. From the perspective of stages in stochastic programs, and the corresponding modeling, we have two major choices with respect to when demand becomes known:

1. Demand for a period becomes known before production for that period is determined.
2. Demand for a period becomes known after production for that period is determined.

Neither of these is better than the other, and probably both are incorrect since most likely we learn little by little. But we need to make a choice modelingwise, so let us assume we learn the demand before production is determined. The time line of our interpretation can be found in Table 2.4.

Table 2.4: Time line for our interpretation of the inventory model

	$t = 1$		$t = 2$		$t = 3$	$\cdots$	$t = T - 1$		$t = T$	
x_0, d_1	x_1, u_1	d_2	x_2, u_2	d_3	x_3, u_3	$\cdots$	x_{T-1}, u_{T-1}	d_T	x_T, u_T	
I_0		I_1		I_2		$\cdots$		I_{T-1}		I_T

Let $\mathcal{S}$ be a scenario tree describing how $\boldsymbol{d}$ develops randomly over time. (Chap. 4 discusses the making of such trees.) A path of demands in $\mathcal{S}$ is called a scenario, is expressed by $d^s = (d_1^s, \ldots, d_T^s)$, and occurs with probability p^s. Similarly, production and inventory are described by x_t^s and I_t^s. We require that all variables be *implementable* (also called nonanticipative—if you can pronounce that), in the sense that when two scenarios s and σ are such that $d_t^s = d_t^\sigma$ for $t = 1, 2, \ldots, \tau$, then $x_\tau^s = x_\tau^\sigma$ and $I_\tau^s = I_\tau^\sigma$. The reason is that since the two scenarios are indistinguishable when the production for period τ is made, the production and inventory decisions must be the same.

It is worth a few minutes of your time to be sure you understand this way of formulating a stochastic program. Earlier, in Fig. 1.1, you saw a (simple) scenario tree. That tree branched each time we learned something, and each node in the tree had its own decision variables. A scenario is a path in that tree, and the scenarios interact directly by all scenarios sharing the top node (Stage 0). But there is an alternative formulation. It is sometimes chosen because it provides a better way to outline the problem structure, sometimes because we plan to use a solution method that is based on the formulation. For this book, we want to emphasize that it might at times be easier to read for people not used to stochastic programs. Consider Fig. 2.3, where we challenge you to fill in what is missing.

As you can see, each scenario, represented by a column in Fig. 2.3, has its own set of variables. The first line shows information that just became known when we reached this node (incoming inventory and this period's demand), the second row what must be determined (production and external orders). That means that the basic part of the model is as in the deterministic situation—all information is known. But in addition we impose some requirements. These are what we call implementability or nonanticipativity constraints. Consider the first row in the figure. All boxes have been connected by horizontal lines. That means that decisions in those eight boxes must take on the same values. In particular:

$$x_1^1 = x_1^2 = \cdots = x_1^7 = x_1^8$$

and

$$u_1^1 = u_1^2 = \cdots = u_1^7 = u_1^8.$$

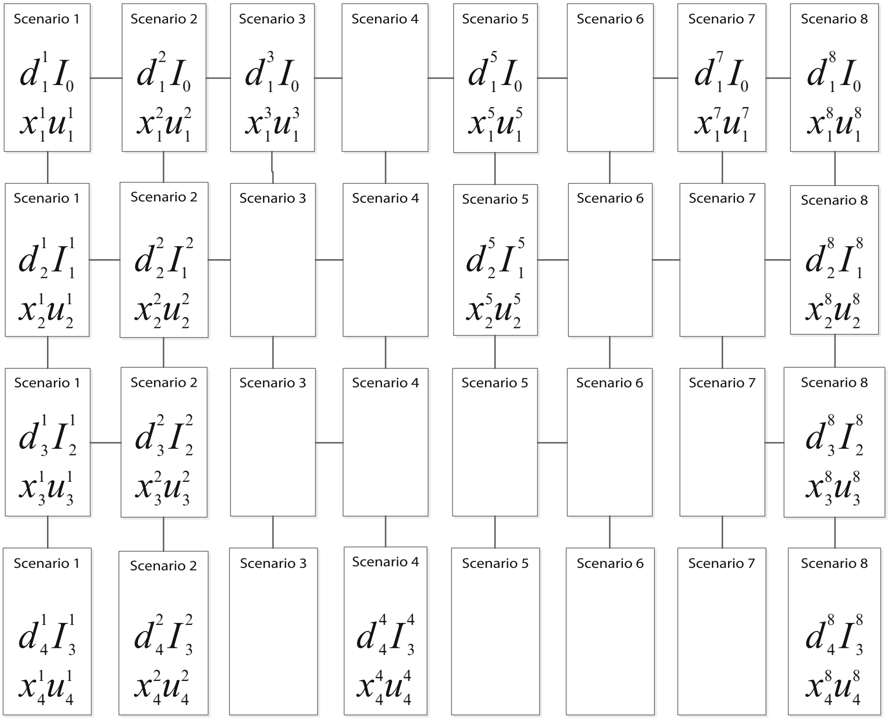

Fig. 2.3: Information structure of the inventory problem outlined scenario by scenario for $T = 4$. A column represents a scenario. Two boxes with a horizontal line between them must have the same values for all the variables

The meaning of this is that since at this point in time we cannot know which of the eight scenarios we are on, we cannot allow decisions to depend on scenarios. Remember that we assumed that demand became known *before* production decisions were made. Hence, in fact, we have assumed that

$$d_1^1 = d_1^2 = \cdots = d_1^7 = d_1^8$$

and, more obviously, that incoming inventory I_0 is the same for all scenarios.

Here you can see the meaning of the two, possibly slightly peculiar, terms *implementability* and *nonanticipativity*. A decision is implementable, i.e., can be implemented, if it does not depend on values that are not yet revealed. "Buy IBM stock now if its value goes up next month" is *not* implementable, while "Buy IBM stock now" is. In the same way, "Buy IBM stock now if its value goes up next month" is anticipative, i.e., it anticipates (uses) information that is not yet known. Nonanticipativity then means that a decision does not use such unavailable information.

The term nonanticipativity is most used. But be a bit careful about how you understand it. The whole point of stochastic programming is to be anticipative, that is, to look into the future and consider what *might* happen. However, you must not anticipate what *will* happen.

In the second row in the figure, the four left nodes are connected, as are the right four. This implies that since $d_2^1 = d_2^2 = d_2^3 = d_2^4$, all the first four scenarios must have the same second-stage decisions since we cannot know which of them we are on. The same applies to the right four nodes. Then, in the third row, only two and two nodes are connected, and then at the end, you know exactly which scenario you are on.

So to say that the variables are implementable, or nonanticipative, is to say that they must satisfy the information structure inherent in Fig. 2.3. A straightforward formulation then becomes (with $x_0^s = x_0$ given for all $s \in \mathcal{S}$):

$$\begin{array}{ll} \min & \displaystyle\sum_{s\in\mathcal{S}} p^s \sum_{t=1}^{T} (c_t x_t^s + f_t I_t^s + b_t u_t^s) \\ \text{such that} & \left\{ \begin{array}{ll} I_t^s - I_{t-1}^s = x_t^s - d_t^s + u_t^s, & t = 1, \ldots, T; s \in \mathcal{S}, \\ |x_t^s - x_{t-1}^s| \leq \ell, & t = 2, \ldots, T; s \in \mathcal{S}, \\ x_t^s, I_t^s, u_t^s \geq 0, & t = 1, \ldots, T; s \in \mathcal{S}, \\ x_t^s, I_t^s, u_t^s \ \text{ implementable} & t = 1, \ldots, T; s \in \mathcal{S}. \end{array} \right. \end{array} \tag{2.8}$$

Hopefully you can see that, although Fig. 2.3 is perhaps a bit involved until you get used to it, the advantage of being able to write down the problem in terms of scenarios can be substantial in terms of clarity. This is especially true for multistage problems.

2.3.2 Analysis

Let us now see what this model implies, particularly relative to the deterministic version. First, the inventory constraint, combined with nonnegativity on inventory, has turned the model into something like a worst-case model: however high the demand, we *must* be able to satisfy it. As a modeler, you might ask: was this what we wanted? Of course, the deterministic model also had this property, but since demand (most likely) was set at its mean value, the actual effect was less dramatic.

On the other hand, in light of this we might ask what the deterministic model means (unless the modeler really thinks the world is deterministic). The model requires that we *must* meet average demand and does not allow for *any* violation of that requirement. But at the same time, obviously, if the results from the model are to be used in reality, then a shortage will occur. The model does not say anything about what will happen then.

Observe also that while demands up to the mean must be met at any cost, demands above that level do not matter at all! Such an arbitrary setup can lead to rather strange results in terms of behavior in the real world. So we see a deterministic model that makes good sense in its own setting—a deterministic world—but that becomes rather strange when we start to think about its use in a real setting. The problem is that deterministic models often do not answer major questions about what to do under difficult conditions, simply because they assume them away.

We also should look at the constraint on variation in production. As demand now varies more than in the deterministic model, it is now a much more serious constraint. The model might decide on large inventories simply to facilitate cases (possibly with very low probabilities) where demand changes a lot from one period to the next. Was that really the understanding of the underlying problem? Especially the total combination of all constraints results in a very special situation: limited ability to change production plus an absolute need to deliver will give very large inventories, most likely at a level that users will consider unreasonable.

Of course, the foregoing situation might be the right one. It is not our point to say it cannot be. But most likely the starting point will be a situation where shortages (whether they result in lost sales or backorders) are considered bad and where large changes in production should be avoided. But if demand suddenly increases dramatically, it will likely be possible to increase production by more than ℓ. And if inventory is empty, most likely the company does not go directly out of business. There are solutions.

Ways around this could be to put the constraints of smoothness in production into an objective function, with a penalty for all changes or all changes above ℓ. Also, we should think seriously about modeling backorders or lost sales unless we *really, really* must satisfy all possible demand. But if we do not, we face another problem: are we sure our estimates of maximal demands are correct? Most likely we are not. And if so, the model is a bit shaky: while we require feasibility at any cost, at the same time, in reality, shortages might occur. This does not represent good modeling.

2.3.3 Chance-Constrained Formulation

We can also imagine probabilistic constraints here. With discrete distributions, one (but certainly not the only) possibility is to replace $I_t \geq 0$ by

$$\sum_{s \in \mathcal{W}_t(x)} p^s \leq 1 - \alpha, \quad \mathcal{W}_t(x) = \{s | I_t^s < 0\},$$

expressing that at most $1-\alpha$ parts of the time inventory may be negative, time period by time period, in the tree. Apart from obvious problems of solving such a model (remember that I_t depends on all previous demands and production decisions), this model does not really soften the original hard constraints of zero inventory levels. It simply moves them to certain negative levels and then requires that these new limited negative levels must be achieved at any cost, whereas beyond that level, costs are of no concern at all. This is a rather general observation. Chance constraints may appear to be softer than the original constraints. But in general they are not; they are simply different, usually looser, but not softer.

In this example we expressed the chance constraints period by period, so the requirement is not history dependent. An alternative formulation would express the inventory constraint node by node in the scenario tree, rather than period by period.

2.3.4 Horizon Effects

Many problems, particularly those that are inherently multistage, in fact have infinitely many stages. By that we do not imply that the problems cover infinitely long time intervals but that it is not known when the problems end.

A production company knows, of course, that eventually a certain product will go out of production, but they do not know when, so they plan as if the production will go on forever. A financial investor will, sooner or later, go out of business, but he plans as if he will not. That is the nature of most decision problems. We really have no choice but to treat these problems as if they had infinitely many stages. But there is no way we can, technically speaking, handle models with infinitely many stages. We need to make some kind of simplification or approximation to make the model have finitely many stages—if possible, only two.

There is also a modeling-related reason for this: As the number of stages gets very large, we approach a kind of steady-state situation. But as was pointed out previously, stochastic programming is about transient behavior, not steady-state behavior. We are interested in what to do *now*, not what to do when (if) we eventually reach a steady state. So at best we need to represent the steady state in the model to obtain the right transient behavior. But what actually to *do* once we get there is not at all our focus.

2.3.5 Discounting

In problems with many time stages stretching into the distant future, we want to account for the fact that a payment in the future is less valuable than a payment today. For example, we could take a dollar today and invest it in a secure deposit account that pays interest, say, r during each period t. One dollar invested today in this account for t periods would hold

$$\prod_{\tau=1}^{t} (1+r)\,.$$

The inverse of this value represents the amount we would put into the deposit account today in order to receive exactly one dollar at time t. This ratio $\delta = (1+r)^{-1}$ is called a *discount rate.* As time passes, discounting progressively lowers the value of a future dollar. For instance, at 15 % interest rate today's value of a dollar to be received 5 years from now is \$0.50, and 10 years from now it is \$0.25.

The *discounted present value* of a future cash stream $f = [f_1, \ldots, f_T]$ is

$$\sum_{t=1}^{T} \delta^t f_t.$$

This formula simply adds up the various amounts we would have to put into our deposit account to generate each term of the payment stream f. As time goes on, the discounting reduces the impact of the future on present decision making and imposes a kind of "significance" horizon on comparisons between future cash streams. Discounting is a very common practice in planning problems, but we do not intend for you to always accept such smoothing when it appears. For example, is it sensible to discount global warming or nuclear conflict? It does not take much of a discount factor to smooth away the impact of serious events that are far away in time.

2.3.6 Dual Equilibrium: Technical Discussion

So now suppose in the production-inventory problem that there is a time point t in the future when we will not be too concerned about randomness. We could just as well use mean values for the costs and demands. Even if we do not necessarily know what these will be, we do not want the model to assume that they are *zero* since this could introduce strange incentives into the preceding periods. In stochastic programming, for numerical reasons, we cannot use lots and lots of time stages to gently smooth out the impact of a shock like that, but we can develop an approach that approximates this kind of smoothing.

Let us look at the optimization problem to be solved at this final horizon stage t. The data we have are the decisions from the past stage $t-1$ and our best guesses as to the steady-state costs and demands. Let us also introduce a time discount factor $0 < \delta < 1$ that applies to the costs in the horizon objective function:

$$\begin{aligned} c_{t+\tau} &= \delta^\tau c_t, \\ f_{t+\tau} &= \delta^\tau f_t, \qquad (2.9) \\ b_{t+\tau} &= \delta^\tau b_t. \end{aligned}$$

This discounting will have the effect of smoothing the impact of errors in horizon estimates, which is what we desire for this model:

$$\begin{aligned} \min \quad & \sum_{\tau=0}^{T-t} \delta^{\tau} \left(c_{t+\tau} x_{t+\tau} + f_{t+\tau} I_{t+\tau} + b_{t+\tau} u_{t+\tau} \right) \\ \text{such that} \quad & \begin{cases} I_{t+\tau} - I_{t+\tau-1} = x_{t+\tau} - d_{t+\tau} + u_{t+\tau}, & \tau = 0, \ldots, T-t, \\ x_{t+\tau} - x_{t+\tau-1} \leq \ell, & \tau = 0, \ldots, T-t, \\ x_{t+\tau} - x_{t+\tau-1} \geq -\ell, & \tau = 0, \ldots, T-t, \\ x_{t+\tau}, I_{t+\tau}, u_{t+\tau} \geq 0 & \tau = 0, \ldots, T-t. \end{cases} \end{aligned} \tag{2.10}$$

Dual equilibrium (due to Grinold [18]) simplifies the horizon problem by equalizing all future undiscounted shadow prices. Of course, this means that the shadow prices are not going to be set at the level that guarantees constraint satisfaction, so in effect we are choosing to overlook feasibility violations in order to simplify the horizon problem. This may not be the correct approach in every modeling situation, so let us think about what it means in the production-inventory problem. The constraints model the satisfaction of demand while restricting production variations from one period to another. If the constraints are not satisfied in some period past the horizon, should we be concerned? This is problem dependent, of course, but let us proceed as if this were a reasonable assumption.

To implement the dual-equilibrium assumption in (2.10), we start by denoting the dual multipliers (shadow prices) for the first, second, and third constraints in each horizon stage by y_τ, z_τ^+, and z_τ^-, respectively. The first step in the dual-equilibrium approach assumes that the dual multipliers all take the form

$$\begin{aligned} y_\tau &= \delta^\tau\, y_t, \\ z_\tau^+ &= \delta^\tau\, z_t^+, \\ z_\tau^- &= \delta^\tau\, z_t^-, \end{aligned} \tag{2.11}$$

which, as we can see, has the effect of equalizing the undiscounted shadow prices for each time stage. If these dual multipliers were actually optimal, then the primal solution for (2.10) could be recovered by optimizing the following objective function:

$$\begin{aligned} \textstyle\sum_{\tau=0}^{T-t} \delta^\tau \Big[& \big(c_t x_{t+\tau} + f_\tau I_{t+\tau} + b_t u_{t+\tau} \big) \\ & + y_t \big(I_{t+\tau} - I_{t+\tau-1} - x_{t+\tau} + d_{t+\tau} + u_{t+\tau} \big) \\ & + z_t^+ \big(x_{t+\tau} - x_{t+\tau-1} - \ell \big) \\ & + z_t^- \big(x_{t+\tau} - x_{t+\tau-1} + \ell \big) \Big]. \end{aligned} \tag{2.12}$$

Of course, these dual-equilibrium multipliers are not likely to be the optimal ones for (2.10). For one thing, there are only three dual multipliers left, so the model only has three constraints! The next step in the dual-equilibrium procedure is to collect terms in expression (2.12) to see what these constraints are.

To see the constraints, we need to add up all three sets of terms that involve the three dual multipliers (y_t, z_t^+, z_t^-) multiplied by the discount factor δ^τ. To simplify these expressions, let us introduce "integrated" primal variables and right-hand sides as follows:

$$\begin{aligned}
d_t^* &:= \textstyle\sum_{\tau=0}^{T-t} \delta^\tau \, d_{t+\tau}, \\
\ell_t^* &:= \textstyle\sum_{\tau=0}^{T-t} \delta^\tau \, \ell, \\
x_t^* &:= \textstyle\sum_{\tau=0}^{T-t} \delta^\tau \, x_{t+\tau}, \\
I_t^* &:= \textstyle\sum_{\tau=0}^{T-t} \delta^\tau \, I_{t+\tau}, \\
u_t^* &:= \textstyle\sum_{\tau=0}^{T-t} \delta^\tau \, u_{t+\tau}.
\end{aligned}$$

What can these integrated expressions mean? Interpreting these expressions really gets us to the heart of the dual-equilibrium approach. By assuming the form (2.11) for the dual-variables structure, we are essentially going to add up (or integrate) all the future demands and production constraints out to the horizon and then choose production, inventory, and external order decisions that satisfy the integrated demand and production constraints. The "integration" incorporates the discount factor, as it should, in order to appropriately scale the impact of the future periods back to the horizon period.

Using the integrated primal variables and right-hand sides you can verify that (2.12) transforms into the following expression:

$$\begin{aligned}
&c_t x_t^* + f_t I_t^* + b_t u_t^* \\
+&y_t\big(I_t^* - \delta(I_{t-1} + I_t^*) + \delta^{T-t} I_T - x_t^* + d_t^* + u_t^*\big) \\
+&z_t^+\big(x_t^* - \delta(x_{t-1} + x_t^*) + \delta^{T-t} x_T - \ell_t^*\big) \\
+&z_t^-\big(x_t^* - \delta(x_{t-1} + x_t^*) + \delta^{T-t} x_T + \ell_t^*\big).
\end{aligned} \tag{2.13}$$

We performed the following calculation:

$$\sum_{\tau=0}^{T-t} \delta^\tau x_{\tau-1} = \delta(x_{t-1} + x_t^*) - \delta^{T-t} x_T.$$

Is that really correct? We think it is, but it will help you to understand what is going on if you try to verify it yourself. If we assume that T is so large that

δ^{T-t} is practically zero and drop the asterisks on the primal variables, then the dual-equilibrium horizon problem is

$$\begin{aligned} \min \quad & c_t x_t + f_t I_t + b_t u_t \\ \text{such that} \quad & \begin{cases} (1-\delta)I_t - \delta I_{t-1} = x_t - d_t^* + u_t, \\ |(1-\delta)x_t - \delta x_{t-1}| \le \ell_t^*, \\ x_t, I_t, u_t \ge 0. \end{cases} \end{aligned} \tag{2.14}$$

The variables (x_t, I_t, u_t) in the horizon problem can be interpreted as the amounts required to satisfy the integrated demand and production constraints out to the horizon.

Dual equilibrium is a general approach to the modeling of horizon constraints. It can be applied to many types of problems. However, like any general approach, there are assumptions made and the solution must be checked to understand if the assumptions are reasonable.

So, in total, we end up with the following extension of (2.8). Notice that T has changed interpretation from the development of the dual-equilibrium horizon model. T is now the period in which we apply the horizon model.

$$\begin{aligned} \min \quad & \sum_{s\in\mathcal{S}} p^s \sum_{t=1}^{T} (c_t x_t^s + f_t I_t^s + b_t u_t^s) \\ \text{such that} \quad & \begin{cases} I_t^s - I_{t-1}^s = x_t^s - d_t^s + u_t^s, & t = 1,\dots,T-1; s \in \mathcal{S}, \\ (1-\delta)I_T^s - \delta I_{T-1}^s = x_T^s - d_T^* + u_T^s, & s \in \mathcal{S}, \\ |x_t^s - x_{t-1}^s| \le \ell, & t = 2,\dots,T-1; s \in \mathcal{S}, \\ |(1-\delta)x_T^s - \delta x_{T-1}^s| \le \ell_T^*, & s \in \mathcal{S}, \\ x_t^s, I_t^s, u_t^s \ge 0, & t = 1,\dots,T; s \in \mathcal{S}, \\ x_t^s, I_t^s, u_t^s \ \text{ implementable }, & t = 1,\dots,T; s \in \mathcal{S}. \end{cases} \end{aligned}$$

As before, "implementable" refers to the structure discussed in Fig. 2.3.

2.4 Summing Up Feasibility

The three examples, together with the news vendor example from Sect. 1.2, show many different aspects of modeling stochastic decision problems. However, much of the discussion, one way or another, concerns feasibility. Let us try to sum up what we have seen.

What we notice here is that although many deterministic models can be made stochastic, the steps cannot be automated. We must think about what the constraints actually imply in the stochastic setting, and we must be particularly concerned about the handling of feasibility. Very often feasibility is not as strict as we imply by our modeling, and using penalties is better.

Feasibility in this context has two components. First, constraints that seem reasonable in a deterministic setting turn into worst-case analysis in a stochastic environment. Although that might be what we want, it is rarely the case. The difficulty is simply that in a deterministic setting, where parameters normally are expected values, the constraints seem reasonable, even if they in fact represent goals that might be violated. After all, the model only handles the average case.

But in a stochastic setting, if the constraints actually represent wishes or goals and violations can be allowed, though possibly at a high cost, the constraints should be moved into the objective function with a penalty. Except for what we might call bookkeeping constraints—inventory out equals inventory in plus production minus sales—not using constraints in stochastic models should be avoided.

Constraints that remain constraints in the problem formulation are called *hard* constraints, while those moved into the objective function are called *soft* constraints. Our claim is that from a modeling perspective, most constraints are soft.

The second component of feasibility is that many, if not all, deterministic models lack a proper stage structure. Even though a deterministic model may include time periods, the variables are not properly adjusted. A side effect of this is that when the information structure of an event tree is added to a problem, the variables must be redefined. For example, in the news vendor example of Sect. 1.2, we had the same variable for orders and sales. This might be fine in a deterministic world where you never produce something you will not need, but in a stochastic model, you must realize that production and sales belong to different stages of the model and cannot be represented by the same variables. A more straightforward example was seen in the overhaul project example in Sect. 2.2, where we had to impose a scenario index on some variables. That amounts to defining new variables, even though we continued to use the old variable name.