OPTIMIZATION UNDER UNCERTAINTY

LECTURE NOTES

2001

R. T. Rockafellar

University of Washington

CONTENTS

1. INTRODUCTION

Problems of optimization under uncertainty are characterized by the necessity of making decisions without knowing what their full effects will be. Such problems appear in many areas of application and present many interesting challenges in concept and computation. For a beginner, it's most important at the outset to gain some appreciation of just how optimization under uncertainty differs from other branches of optimization, and what the basic modeling issues are. A review of terms will lay the foundation for this.

Optimization: This refers to the analysis and solution of problems in which a single choice must be made from a range of "feasible" ones. Feasible choices are modeled as the elements of some set—the *feasible set*. The goal is to find a "best" choice (not necessarily unique), or at least a "better" one than might readily be detected. Choices are compared according to the values they give to a certain function—the *objective function* in the problem. The goal may be maximization or minimization of such values. For simplicity, minimization is taken to be the standard.

Discrete optimization: There are only finitely many options to compare, but the number of them could be enormous. The feasible set is a finite (discrete) set, hopefully with enough structure to support some degree of effective analysis.

Continuous optimization in finite dimensions: The choices are describable in terms of the values of finitely many real variables. The feasible set can therefore be modeled as a subset of a space $\mathbb{R}^n$.

Continuous optimization in infinite dimensions: Not just vectors of variables, but entire functions, scalar-valued or vector-valued, have to be chosen. These unknowns, often interpreted as "policies" of response or control to ongoing situations, may be regarded as elements of an infinite-dimensional vector space.

Sources of infinite dimensionality: Problems typically have infinitely many "degrees of freedom" when the decisions need to be represented as depending on continuous time or on observations of continuous random variables.

Finite-dimensional approximations: Infinite-dimensional problems can't ordinarily be solved numerically without converting them somehow to problems in finite dimensions.

Uncertainty: Decisions must often be taken in the face of the unknown. Actions decided upon in the present will have consequences that can't fully be determined until a later stage. But there may be openings for corrective action later or even multiple opportunities for recourse as more and more becomes known.

Potential applications of optimization under uncertainty: Here are some prime examples. Note that time is almost inevitably involved, because exact knowledge of the future is what most typically is lacking.

Generation of electrical power: An electrical utility has to decide each day how much power to produce without yet knowing the demand, which may depend on weather developments (as could affect heating, cooling, etc.) Longer range decisions may concern the amount of coal to purchase or the kinds of buying/selling contracts set up with other utilities.

Operation of reservoirs: Systems of water reservoirs and their connecting flows have to be managed in a reasonable manner for drinking water, irrigation, hydropower, flood control, recreation, and the protection of fish runs. As if these conflicting goals didn't already pose difficulty, there is climate uncertainty affecting both the inflows into the system and the intensity of irrigation needs and the rest.

Inventory management: Supplies of retail goods or spare parts must reliably be maintained despite the vagaries of demand and the costs of restocking.

Portfolio selection: Investments must be made in stocks, bonds, foreign currencies and the like without knowing for sure how their values will rise or fall. At the root of the uncertainties are interest rates and political developments.

Facility planning: Where should new facilities be located, and what capacity should they have, if they are designed to serve needs that are anticipated from guesswork? Similar: problems of reconfiguration or expansion of existing facilities.

Pollution control: What types and sizes of treatment plants should now be built to ensure water quality in a basin where population growth and future manufacturing requirements can only be estimated roughly, and where anyway a lot depends on what will happen with rainfall over a long term?

Stabilization of mechanisms: An antenna must be kept fixed as steadily as possible on the source of signals it's supposed to receive, despite random wind gusts that can push it out of line. What configuration of forces should be brought into play to restore misalignments as rapidly as possible, without neglecting the potential of additional wind gusts before the process is complete?

Analysis of biological systems: Are the feeding and breeding strategies of animals optimal in the sense of "maximizing survival" in a particular biological niche with respect to its uncertainties of climate, food supply, predation, and so forth? Note that here the issues are theoretical rather than numerical.

The modeling of uncertainty: The uncertainties in a problem have to be represented in such a manner that their effects on present decision-making can properly be taken into account. This is an interesting and challenging subject.

Stochastic modeling: The uncertain elements in a problem can often be modeled as random variables to which the theory of probability can be applied. For this purpose such elements have to have a "known" probability distribution.

Degrees of knowledge: Such a distribution may be available from statistical data (as with weather variables), or it may be no more than an educated guess (as with interest rates or election prospects)—"subjective" probabilities. Either way, the mathematical treatment is the same.

Deterministic modeling: This refers to mathematical formulations in which uncertainty plays no role. In practice, such modeling often prevails even in situations with obviously important uncertainties, because the modelers don't know how to cope with those features, or don't have adequate data to work with, or don't yet have good software available for getting numerical solutions.

Deterministic versus stochastic: These terms are often contrasted with each other in the description of mathematical models.

Range modeling: Sometimes, when a deterministic model is clearly inappropriate, yet there are few clues to the probabilities that would support a stochastic model, it's useful to work with *ranges* of uncertainty. Various quantities that would otherwise be data parameters are neither given specific values nor probability distributions but merely viewed as restricted to particular intervals. One tries to guard against whatever might happen by thinking of the actual values to be faced as chosen from these intervals by an adversary, as in a game setting.

Uncertain probabilities: This notion can be combined with stochastic modeling by supposing that a probability distribution is present but is incompletely known. The actual distribution to be faced will be chosen by an adversary—like Mother Nature—from some limited set of distributions, described perhaps by ranges on statistical parameters.

The role of scenarios: A common tool in planning for the future is to work with *scenarios*, which are particular representations of how the future might unfold. Some kind of probabilistic model or simulation is used to generate a batch of such scenarios. The challenge then, though, is how to make good use of the scenarios in coming up with an effective decision.

A common but faulty approach: Often, planners just solve, for each scenario that is generated, an optimization problem which arises from taking that scenario to be the path the future truly will follow. These problems are themselves deterministic in character. Although each yields a prescription of what should be done here and now, there's no theoretical guidance about the compromise between those prescriptions that should actually be adopted, even when probabilities can be assigned to the individual scenarios. Indeed, the separate prescriptions obtained for the individual scenario problems may be inconsistent with each other and very fragile—not adequately *hedged*. They're optimal only in a context where one can act with perfect foresight.

The need for modeling the evolution of information: The crucial feature demanded for serious applications is the effective modeling of how observations at various future stages increase the store of knowledge on which decisions can properly be based, not only in the present, but in subsequent stages as well. In the setting of discrete probability, we'll formalize this later in terms of a "scenario tree."

Problem data in an elementary framework: It will help in understanding the main issues if we first look at an uncluttered formulation. A basic problem of optimization in $\mathbb{R}^n$ takes the form:

$$\text{minimize } f_0(u) \text{ over all } u \in C$$

for a function $f_0 : \mathbb{R}^n \to \mathbb{R}$ and a set $C \subset \mathbb{R}^n$. This set might be specified as consisting of the points u in an underlying subset $U \subset \mathbb{R}^n$ that satisfy some constraints on the values of additional expressions $f_i(u)$, say

$$f_i(u) \begin{cases} \leq 0 & \text{for } i = 1, \ldots, s, \\ = 0 & \text{for } i = s+1, \ldots, m, \end{cases}$$

but for now it will be better not to focus on such a representation and to think instead of the condition $u \in C$ as expressing feasibility more abstractly.

Essential objective function: Even simpler notation, very convenient in theoretical discussions, results from the device of enforcing constraints by infinite penalties. For the basic problem just described, the *essential objective function* is

$$f(u) := \begin{cases} f_0(u) & \text{if } u \in C, \\ \infty & \text{if } u \notin C. \end{cases}$$

Regardless of the representation eventually given to C, the problem can be written equivalently in terms of this function as

$$\text{minimize } f(u) \text{ over all } u \in \mathbb{R}^n.$$

Cost interpretation: For the most part it works well to think of $f_0(u)$—and $f(u)$ too—as the "cost" associated with choosing u. An infinite cost for a choice u means that it is infinitely undesirable and must be avoided. A negative cost corresponds to a reward.

Uncertain constraints and costs: Imagine now a situation in which we have a feasible set $C(\omega) \subset \mathbb{R}^n$ along with a cost expression $f_0(u, \omega)$, in which ω stands for something uncertain—an element ranging over a space Ω representing the "states of the world" that might be faced. (For instance, Ω might be a probability space generated by certain random variables; more about this later.) By the device of infinite penalties, it's equally good to think of the given data as embodied in a single function f, given by

$$f(u, \omega) := \begin{cases} f_0(u, \omega) & \text{if } u \in C(\omega), \\ \infty & \text{if } u \notin C(\omega). \end{cases}$$

Indeed, there's no need even to speak of f_0; we could proceed directly from the idea that we are given an extended-real-valued cost function $f : \mathbb{R}^n \times \Omega \to \overline{\mathbb{R}}$, in which case we could regard $C(\omega)$ as consisting by definition of the points $u \in \mathbb{R}^n$ with $f(u, \omega) < \infty$.

But where is the optimization? Here we have "problem data" involving uncertainty, but we don't yet actually have a "problem," not to speak of an "optimization problem." For that, we still have to come up with a prescription of exactly *what* should be minimized over *what*. Above all, we have to clarify the interplay between the decision process and the uncertainty. How are we supposed to think of one of these as possibly affecting the other?

Reversion to a deterministic formulation: Most frequently by far in these circumstances, the uncertainty is pushed out of sight through the replacement of the random element ω by some particular *estimate* $\widehat{\omega}$. The problem solved is:

$$\text{minimize } f(u, \widehat{\omega}) \text{ over all } u \in \mathbb{R}^n.$$

What happens for other choices of $\omega \in \Omega$ is disregarded.

Reason: The justification usually given for such a move is that problems are hard enough to solve as it is, without getting entangled in difficult matters like uncertainty, and that anyway this step is in the tradition of "approximation" which most applications of mathematics anyway have to follow sooner or later.

Criticism: Sympathy is well earned for the first part of this assertion, but not for the second. Approximation only makes sense when there is, to begin with, a clear idea of the underlying situation that's being approximated.

Hedging: To be effective in dealing with uncertainty, an optimization model ought to somehow involve *hedging* of decisions, i.e., balancing risks by "not putting all the eggs in one basket." But hedging is impossible when only a single element $\widehat{\omega}$ is made the focus, no matter how well selected it might be. A decision based on condensing the uncertainty just to $\widehat{\omega}$ can't do a good job of reflecting the realities.

Decision versus observation: The question of which comes first, the fixing of the decision or the making of an observation that dissipates the uncertainty, is all-important. Both cases can very well arise, and in fact we'll soon be exploring situations in which partial decisions and partial observations are made in stages and are interspersed with one another. For now, however, we'll keep to the simplest picture, where it's just one or the other.

Observation first: If the decision can be put off until after the value of ω becomes known, what we have really is an optimization problem in u that depends on ω as a parameter:

$$\text{minimize } f(u,\omega) \text{ over all } u \in \mathbb{R}^n.$$

We could either regard this as a single problem, to be solved only for the ω value eventually realized, or as a family of problems yielding an optimal value

$$\psi(\omega) := \inf_u f(u,\omega)$$

and an optimal solution set

$$\Psi(\omega) := \operatorname{argmin}_u f(u,\omega),$$

both of which need to be analyzed in their dependence on ω. We might want to treat this optimal value and optimal solution set as random elements whose probabilistic behavior is induced by that of ω.

Modeling of future actions: The parametric version is basic to evaluating responses that might be made to future events. It will tie in with the notion of "recourse" that we'll eventually be making much of. A function $\omega \mapsto u(\omega) \in \Psi(\omega)$ will in that case play a major role. We'll go into this in considerable detail later, especially when we come to numerical methods.

Decision first: If the decision has to be taken in advance of any information about ω other than its probability distribution, we know what space we must be minimizing over—namely $\mathbb{R}^n$ (or one of its subsets)—but what should be minimized? For each choice of u we have not just a single cost, but an uncertain value $f(u,\omega)$; in other words, we have a function $\omega \mapsto f(u,\omega)$ on Ω. To get any further, we have to pass somehow from a *function* depending on u to a *number* depending on u. Only then will we be in position to minimize with respect to u.

Worst-case approach: In line with range modeling, we can take the problem to be that of minimizing

$$\tilde{f}(u) := \max_\omega f(u,\omega).$$

We look only at the worst outcome that might result from choosing u and make no attempt to distinguish between outcomes according to their likelihood.

Notation: Of course, it's not correct here to write "max" unless we have some assurance that the maximum of $f(u,\omega)$ with respect to $\omega \in \Omega$ for fixed u is attained, as for instance when Ω is finite. Otherwise we ought to write "sup" for the least upper bound.

Stochastic approach: When Ω is a probability space, we can interpret $f(u,\omega)$ for fixed u as a random variable that inherits its distribution from ω. This random variable, dependent on u, can be converted in various ways to a numerical quantity dependent on u, but we'll concentrate on the easiest scheme, where the random variable is replaced by its "expected value," as signaled by $\mathbb{E}$. The problem then is that of minimizing

$$\widehat{f}(u) := \mathbb{E}_\omega\big\{f(u,\omega)\big\}.$$

For the time being, we'll keep to the case where the probability distribution of ω is the same regardless of the choice of u, but the more complex case, where decisions might influence underlying probabilities, can come up in some applications as well.

Our focus in these notes: Faced with limited time, we'll put our energies into the stochastic approach. The worst-case approach is also interesting and valuable, and it's good for many applications as well.

Probability and expectation: Although we aren't going to go into probability theory to the extent needed to make everything mathematically rigorous in the context of Ω being a really general probability space, a brief discussion of the different levels of the subject will be beneficial.

Discrete probability: Imagine that Ω is just a finite set, and that for each $\omega \in \Omega$ the probability of ω occurring is known. Denote this probability by $\pi(\omega)$; we have $\pi(\omega) \geq 0$ and $\sum_{\omega\in\Omega} \pi(\omega) = 1$. Consider now a quantity $a(\omega) \in \mathbb{R}$ that depends on $\omega \in \Omega$; this is what is meant by a (real) *random variable* over Ω. The *expected value* of this random variable $a(\omega)$ is the weighted sum

$$\mathbb{E}_\omega\{a(\omega)\} := \sum_{\omega\in\Omega} a(\omega)\pi(\omega).$$

This can be thought of as the average outcome experienced in the limit when the random variable is sampled over and over again according to its distribution. The expected value is often called the *mean* and denoted by μ.

Another basic quantity is the *variance* of the random variable $a(\omega)$, which is denoted by σ^2 and defined as the expected value of $|a(\omega) - \mu|^2$:

$$\sigma^2 = \sum_{\omega\in\Omega} |a(\omega) - \mu|^2\pi(\omega).$$

The variance quantifies the spread of the random variable around its mean. The same formulas apply for vector-valued random variables, i.e., when $a(\omega)$ is a vector in $\mathbb{R}^m$, say, instead of just a real number. (The absolute value is then the Euclidean norm.)

Note: Probability theorists also speak of "discrete probability" when Ω isn't necessarily being comprised of finitely many points, but could consist of an infinite sequence of points. A sum giving an expectation is then an infinite series. For us in these notes, however, the term will always refer to finite Ω, for short.

Continuous probability: Alternatively, Ω could be $\mathbb{R}^d$ or some region in such a space. The probability distribution might then be represented by a density function $\rho(\omega)$ over Ω, in which case the expected value of $a(\omega)$ would come out as a d-dimensional integral:

$$\mathbb{E}_\omega\{a(\omega)\} = \int_\Omega a(\omega)\rho(\omega)d\omega.$$

General probability: More broadly, Ω can be taken to be a probability space in the general mathematical sense—a set supplied with an algebra of subsets ("events") on which there's a *probability measure* π (nonnegative, with $\pi(\Omega) = 1$). The expected value expression is then

$$\mathbb{E}_\omega\{a(\omega)\} = \int_\Omega a(\omega)d\pi(\omega).$$

We'll hold back from this level of sophistication because it would require too much background. Ultimately, though, it's the way to go with the full theory.

Random variables with infinite values: The usefulness of allowing costs to be infinite requires us to deal with random variables $a(\omega)$ having values in $\overline{\mathbb{R}}$ rather than just $\mathbb{R}$. Are the expectation sums or integrals then well defined? There's no difficulty—as long as there's no conflict between opposite infinities.

Rules: Whenever a finite value is added to ∞ the result is taken to be ∞. Similarly whenever a finite value is added to $-\infty$, the result is taken to be $-\infty$. The product of 0 with either ∞ or $-\infty$ is taken to be 0.

In handling an integral, this means that the values of $a(\omega)$ on a subset of Ω having probability 0 can be disregarded, and that for the rest we can form separately the integrals of $\max\{0, a(\omega)\}$ and $\min\{0, a(\omega)\}$. If one or both of these integrals is finite, we simply add the two values together to get the full integral of $a(\omega)$. In the remote event that the two separate integrals turn out to be ∞ and $-\infty$, respectively, we follow the convention that the full integral of $a(\omega)$ is ∞. (This convention comes from minimization being dominant in our framework.)

A technical issue: Also important for rigor in integration are the "measurability" properties that are required for an expectation to make sense. In many areas of applied mathematics there's no serious reason to worry about such things because the functions involved are continuous, say, but don't forget that here we are obliged to work with expressions $f(u, \omega)$ that aren't continuous and may well be *defined* in terms of operations like maximization and minimization, which don't behave well by classical standards.

Daunting questions come up, as they do also incidentally in probability theory itself for related reasons. Fortunately, mathematicians have found satisfying answers in most cases. We won't go into details in this introductory exposition of the subject, except to point out where the difficulties lie. But this doesn't mean that sound technical underpinnings aren't important, just that the understanding of them doesn't have to be the top priority here.

Blanket assumption henceforth about probabilities: As far as these notes are concerned, our underlying assumption *for the sake of mathematical rigor* will always be that Ω is a (finite) *discrete* probability space. But we'll employ language and notation that could operate more generally, and we'll comment from time to time on where the complications lie in moving to a broader framework.

Discretization: If a given problem naturally involves "continuous" random variables, it can be discretized in various ways. One possibility is *simulation*: one can sample the random variables in accordance with their known distribution and thus replace the given probability space with a finite space consisting of the samples (equally weighted, say). Another possibility is to partition the given space into finitely many disjoint subsets, each of which becomes a single element in a new probability space (its probability being the probability of that element). It will be important in multistage problems, however, to discretize properly in a multistage manner. That will be discussed later.

Expected costs: In stochastic modeling, which will be our chief preoccupation, the optimization problem in our simplified setting is to

$$\text{minimize } \widehat{f}(u) := \mathbb{E}_\omega\big\{f(u,\omega)\big\} \text{ over all } u \in \mathbb{R}^n,$$

where the expectation could be given by a sum or an integral according to the nature of the underlying probability space, as explained above.

Alternatives: The emphasis here is on "average outcome" and, from some perspectives, makes the most sense in situations where the same problem might be faced over and over again (although probabilists may disagree with this). Sometimes people like to bring in variance along with mean. In writing the mean of the random variable $f(u,\omega)$ as $\widehat{f}(u)$ and denoting its variance similarly by $\widehat{\widehat{f}}(u)$, one can envision instead a problem of the form

$$\text{minimize } \varphi(u) := \Phi\big(\widehat{f}(u), \widehat{\widehat{f}}(u)\big),$$

where Φ furnishes a way of balancing these two quantities.

Risk: Such is often seen for instance in financial applications, where higher variance is associated with higher volatility. In maximizing returns, some investors are risk-averse and happy to settle for a lower expected value if it comes with lower variance. Other investors are gamblers who enjoy the prospect of big gains even if accompanied by a risk of big losses. These different attitudes could be served by different choices of Φ in formulating the objective. We won't follow up on this here, however.

Constraints in expected costs: In the minimization of $\widehat{f}(u) = \mathbb{E}_\omega\big\{f(u,\omega)\big\}$ over all $u \in \mathbb{R}^n$, the set of feasible solutions is implicitly

$$\widehat{C} := \big\{u \,\big|\, \widehat{f}(u) < \infty\big\}.$$

Any u outside of this set is excluded from consideration. What does that signify?

Insights are available from the motivating case where $f(u,\omega)$ agrees with a given $f_0(u,\omega)$ when $u \in C(\omega)$ but is ∞ otherwise. Suppose that f_0 is "nice" in the sense that the expected value $\widehat{f}_0(u) := I\!E_\omega\big\{f_0(u,\omega)\big\}$ is finite for all u. We then have

$$\widehat{f}(u) = \begin{cases} \widehat{f}_0(u) & \text{if } u \in C(\omega) \text{ almost surely,} \\ \infty & \text{otherwise,} \end{cases}$$

where the term *almost surely* refers to the possible exception of a set of ω values whose total probability is 0. (Students of measure theory will recognize the probabilistic equivalent of "almost everywhere.") Put another way, our scheme assigns an infinite penalty to choosing u if that choice opens up even the slightest (positive) chance of an outcome ω for which $u \notin C(\omega)$. We have

$$\widehat{C} = \big\{u \,\big|\, u \in C(\omega) \text{ almost surely } \big\}.$$

The general case of $f(u,\omega)$ might be more subtle, but the idea is much the same. Any $u \in \widehat{C}$ must in particular be such that $f(u,\omega) < \infty$ almost surely.

Implications for constraint modeling: Suppose, for purposes of discussion, that $C(\omega)$ is specified by a system of constraints of the conventional form

$$u \in U \text{ and } f_i(u,\omega) \begin{cases} \le 0 & \text{for } i = 1,\dots,s, \\ = 0 & \text{for } i = s+1,\dots,m. \end{cases}$$

To sharpen the focus, let's assume that the expressions $f_i(u,\omega)$ really do depend on ω. (Constraints in which ω doesn't actually enter could be absorbed into the specification of U.) How then would we get $\widehat{C}$? Obviously $\widehat{C} \subset U$, but to the extent that $\widehat{C}$ may be smaller than U the situation could be troublesome and may suggest poor modeling.

Inequality constraints: For $i = 1,\dots,s$ we must have $f_i(u,\omega) \le 0$ almost surely, or u won't be in $\widehat{C}$. That is very close to requiring

$$\tilde{f}_i(u) \le 0, \quad \text{where } \tilde{f}_i(u) := \sup_{\omega\in\Omega} f_i(u,\omega),$$

and indeed it's generally the same as that (aside from odd circumstances in which there's a subset $\Omega_0 \subset \Omega$ of probability 0 such that the supremum of $f(u,\omega)$ over ω in the complementary set $\Omega \setminus \Omega_0$ is less than the supremum over all $\omega \in \Omega$). From this perspective, a requirement that $f_i(u,\omega) \le 0$ almost

surely is seen to have a "worst-case" aspect. In imposing it, we are taking a conservative approach and introducing a hurdle that could be costly.

Equality constraints: For $i = s+1, \ldots, m$ we are demanding that $f_i(u, \omega) = 0$ almost surely. The more you think about it, the more you wonder whether this even makes sense. The requirement on u is that it has to make the random variable $\omega \mapsto f_i(u, \omega)$ be not random at all but merely have the constant value 0 (almost surely). That seems to conflict with there being much real dependence of $f_i(u, \omega)$ on ω.

Note that the trouble comes from u having to be chosen *before* ω is observed. Later, when we consider constraints on recourse actions taken *after* ω is observed, equations won't be so questionable.

Conclusion to be drawn: Conventional constraint formulations may be inappropriate in dealing with randomness. Uncertain equality constraints should be avoided, and uncertain inequality constraints should be used only in situations where a worst-case condition is truly intended.

The elimination of uncertain constraints is often possible through closer attention to the underlying problem. A condition $f_i(u, \omega) \leq 0$ has the character of imposing an infinite penalty if, after u has been chosen, there is even a slight chance of an ω coming up that would render $f_i(u, \omega) > 0$, and so it treats such a situation as if "the world comes to an end." Usually, though, the situation isn't so severe and just hasn't been thought out fully.

Penalty substitutes for constraints: A better approach for many applications is to work with finite penalty expressions instead of hard constraints with their infinite penalty aspect. Those expressions can be incorporated into the objective.

For example, in place of a constraint $f_i(u, \omega) \leq 0$ it might make more sense to invoke, whenever this inequality is violated, an additional cost that's proportional to the amount of violation, say at a rate $\lambda_i > 0$. An interpretation might be that when the existing resources in a problem's environment are inadequate it's possible to acquire, at the price λ_i per unit, any supplement that's needed. In this case we would drop the constraint $f_i(u, \omega) \leq 0$ from the problem's formulation and instead add a term $\theta_i\big(f_i(u, \omega)\big)$ to the cost, where

$$\theta_i(\alpha_i) = \begin{cases} 0 & \text{when } \alpha_i \leq 0, \\ \lambda_i \alpha_i & \text{when } \alpha_i > 0. \end{cases}$$

Of course, θ_i needn't be a so-called *linear penalty function* like this; it could have all sorts of other forms.

The upshot is that, instead of defining $f(u,\omega)$ in terms of $f_0(u,\omega)$ and a feasible set $C(\omega)$ given by $u \in U$ and uncertain constraints $f_i(u,\omega) \le 0$ and $f_i(u,\omega) = 0$ as above, one could have

$$f(u,\omega) = \begin{cases} f_0(u,\omega) + \sum_{i=1}^m \theta_i\big(f_i(u,\omega)\big) & \text{when } u \in U, \\ \infty & \text{when } u \notin U, \end{cases}$$

for some choice of penalty functions θ_i tailored to the case at hand. The function $\widehat{f}(u) = I\!E_\omega\big\{f(u,\omega)\big\}$ would then be different, and most notably the feasible set implicit in the minimization of $\widehat{f}(u)$ would simply be

$$\widehat{C} = U.$$

Unproblematical constraints: This picture was sketched under the continuing assumption that $f_i(u,\omega)$ depends on ω. But it's apparent now that one could also proceed by passing to penalties only for the "uncertain constraints" and retaining in the conventional format any constraints where ω doesn't really enter.

Uncertain penalties: There's no difficulty here in generalizing to penalties that aren't fully known at the time when u must be chosen, as for instance when the λ_i coefficient in the linear penalty expression above has to be viewed as a random variable $\lambda_i(\omega)$. Then $\theta_i(\alpha_i)$ becomes $\theta_i(\alpha_i,\omega)$, and terms $\theta_i\big(f_i(u,\omega)\big)$ extend to the form $\theta_i\big(f_i(u,\omega),\omega\big)$.

Chance constraints: Another way of getting around the severe consequences of modeling with uncertain hard constraints is to relax them probabilistically. Suppose that, along with $u \in U$, we would like to have $f_i(u,\omega) \le 0$ for $i = 1,\dots,m$ but know that requiring these inequalities to hold almost surely is too much. We might consider just asking them to hold "most of the time." This can be quantified in terms of

$$\text{prob}\big\{f_i(u,\omega) \le 0\big\} := \text{ probability that this inequality holds (for a given } u).$$

Requiring this probability to equal 1 is the same as requiring $f_i(u,\omega) \le 0$ almost surely. Instead we could require it to be at least some level p_i, for instance $p_i = .99$. The problem could thereby be formulated as one in the conventional format:

$$\begin{aligned} &\text{minimize } F_0(u) := I\!E_\omega\big\{f_0(u,\omega)\big\} \text{ over all } u \in U \text{ satisfying} \\ &\qquad 0 \ge F_i(u) := p_i - \text{prob}\big\{f_i(u,\omega) \le 0\big\} \text{ for } i = 1,\dots,m. \end{aligned}$$

Drawbacks: Many people find this approach very appealing, but in most applications (although not all) it's questionable both from the modeling angle and because of its technical ramifications.

In modeling, the trouble is that there's little guidance for what probability levels p_i ought to be insisted on. It seems that p_i should be higher when the consequences of the inequality $f_i(u,\omega) \le 0$ are more serious, but how much higher? No quantitative assessment is being made of the consequences of having $f_i(u,\omega) > 0$. The penalty approach is based on a closer inspection of such consequences, but here we still regard $f_i(u,\omega) > 0$ as signifying that "the world comes to an end." We merely don't want it to end with a probability greater than $1 - p_i$.

That also raises the issue of constraint interactions. Does it make sense to require $\text{prob}\big\{f_1(u,\omega) \le 0\big\} \ge .99$ while requiring $\text{prob}\big\{f_2(u,\omega) \le 0\big\} \ge .95$, for example, when a violation of either inequality would mean the end of things? What would be the resulting probability of avoiding either calamity? Shouldn't we at least be working with a joint condition of the type

$$\text{prob}\big\{f_i(u,\omega) \le 0 \text{ for } i = 1,\dots,m\big\} \ge p?$$

But how then are we to distinguish between the levels of seriousness with which the constraints should be enforced?

Then too there are the technical hurdles of dealing with the functions $F_i(u) = p_i - \text{prob}\big\{f_i(u,\omega) \le 0\big\}$ numerically. Just what is the nature of F_i, and to what extent can its properties be derived from those of f_i? A common difficulty, for instance, is that convexity of $f_i(u,\omega)$ with respect to u need not carry over to convexity of $F_i(u)$ with respect to u.

Reasonable situations: Chance constraints do make good sense in some cases. An example is the design of a system that necessarily will exhibit some randomness in its behavior, but where one is trying to place an upper bound on one or more variance aspects of this behavior in order to enhance stability of performance.

Properties of expected cost functions: In minimizing $\widehat{f}(u) = \mathbb{E}_\omega\big\{f(u,\omega)\big\}$, the mathematical properties of the function $\widehat{f}$ are bound to be crucial.

Continuity? Something like continuity is needed in ascertaining the existence of optimal solutions and the convergence of sequences that might be generated by a numerical method, but ordinary continuity itself could be too strong a property. The reason is that we are relying on ∞ to signal infeasibility, and a sudden jump of $\widehat{f}(u)$ from a finite value to ∞ could well occur. A substitute property for many purposes is the closedness of the epigraph of $\widehat{f}$; more about such matters later.

Differentiability? To the extent that continuity of $\widehat{f}$ could fail, differentiability could fail all the more. But as a seed for future developments, suppose that $f(u,\omega)$ happens

to be differentiable with respect to u (there being no constraints represented by having $f(u,\omega)$ take the value ∞). Then $\widehat{f}(u)$ ought to be differentiable with respect to u as well, in fact with the gradient formula

$$\nabla \widehat{f}(u) = I\!E_\omega\big\{\nabla_u f(u,\omega)\big\}.$$

Why? In the case of discrete probability, where the expectation is a finite sum, this simply follows from differentiating term by term. More generally, where an integral is involved, it ought to follow similarly under minor assumptions.

This is more than just a formula, however; it's a fertile idea. It suggests that in cases where the probability space Ω is big and complicated we might be able to bypass a full computation of $I\!E_\omega\big\{\nabla_u f(u,\omega)\big\}$ and obtain a working approximation to $\nabla \widehat{f}(u)$ by sampling a relatively small number of elements ω and taking the average of the corresponding gradients $\nabla_u f(u,\omega)$.

Subgradients: When $\widehat{f}$ isn't differentiable, we may be able to use "subgradient" notions as a substitute for gradient. The same idea of simulation through sampling might then be brought into play.

Convexity? Fortunately, convexity does carry through well and provide solid support. If $f(u,\omega)$ is convex in u for each ω, then $\widehat{f}(u)$ is convex in u. When Ω is finite, this comes from the definition of $\widehat{f}(u)$ as a weighted sum. More generally it follows from elementary properties of integration. Convexity of extended-real-valued functions will be discussed in the next chapter.

Probabilities affected by decisions: Tacit in our discussion of the probability space Ω has been the assumption that the probabilities are fixed and independent of the choice of u. What if this were not true? In the setting of discrete probability we could think of the probability associated with ω as having the form $\pi(u,\omega)$. From a cost $f(u,\omega)$ we would then get an expected cost

$$\widehat{f}(u) = \sum_{\omega\in\Omega} f(u,\omega)\pi(u,\omega).$$

Much the same would hold for continuous probability. This kind of generalization makes sense mathematically but greatly escalates the difficulty of working with $\widehat{f}$.

Failure of convexity: In particular, the expected cost function $\widehat{f}$ is highly unlikely to be convex in such circumstances, despite $f(u,\omega)$ being convex in u for each ω. This can be a very serious impediment to solving problems numerically.

2. BASIC STOCHASTIC PROGRAMMING

Two of the main contenders in the methodology of optimization under uncertainty are "stochastic programming" and "dynamic programming." Both take a probabilistic approach to uncertainty and contemplate not just one decision and one observation, but a possible interplay between decisions and observations in stages. They differ significantly, however, in their philosophy, scope and practicality. We'll look now at basic stochastic programming and later at dynamic programming. Eventually we'll examine an advanced stochastic programming setup that combines some of the best features of the two.

Distinguishing features: In overview, the chief characteristics of stochastic programming are the following.

Focus on the here-and-now: Despite heavy involvement of the future, everything is aimed at enhancing the decision that must be made in the present.

Recourse decisions: The attitude is adopted that a decision can't properly be made in the present without taking into account, at least to some extent, the opportunities for modification or correction at later times.

Information through observations: Decisions at later times can respond to information that has become available since the initial decision. This information is modeled by observations of random variables.

Optimization technology: The context is that of large-scale optimization and associated numerical techniques, as adapted and elaborated to handle uncertainty. This isn't to be taken for granted; dynamic programming, in contrast, doesn't make such a vital connection with the rest of the world of optimization.

Large-scale: A term referring to high dimensionality—extremely many variables.

Convexity: Properties of convexity are assumed and made the most of. This is the prime way of coping with the otherwise formidable complications of problems in this area. In principle one could hope someday to cover nonconvexity, but for now that seems out of practical reach. Convexity is essential above all in decomposing a large-scale problem iteratively into digestible pieces.

Probabilities independent of decisions: It's assumed that the probability structure is unaffected by any of the decisions that are taken. Understandably this goes hand in hand with the desire to maintain convexity, but it also draws a crucial dividing line between problems that can be tackled with stochastic programming, in its current state, and those that can't.

Illustration: Typical sources of uncertainty are "weather" or "demand." In hydropower applications, one has to worry on a week-to-week basis about whether rainfall will be adequate to fill reservoirs and, on the other hand, whether a heat wave may cause a surge in the use of air conditioners, say. In the second case the demand for power is in question, but it's weather-dependent. Clearly the weather is driven by forces that aren't going to be affected by a decision to operate turbines in one way or another.

In deciding how big to build a new dam and reservoir, however, it's quite conceivable that population growth and industry could be altered in the long run, and that could well mean a change in the statistics of demand. Especially that would be likely if the utility that constructs the dam seeks to recover costs by advertising to encourage people to use more electricity. (This was common in the industry for many years.)

Ambiguous situations: It's quite possible that when a situation is modeled in the manner that seems most natural, decisions will appear to affect probabilities, but through more careful attention to the setup, this may turn out to be an artifact that can be removed.

Underlying process: The basic ingredient to a problem of stochastic programming, which we need to think about before such a problem can even be formulated, is a "process" in which decisions alternate with observations:

$$
\begin{aligned}
&u_0 \in \mathbb{R}^{n_0} && \text{initial decision}\\
&\omega_1 \in \Omega_1 && \text{observation}\\
&u_1 \in \mathbb{R}^{n_1} && \text{recourse decision}\\
&\omega_2 \in \Omega_2 && \text{observation}\\
&u_2 \in \mathbb{R}^{n_2} && \text{recourse decision}\\
&\quad\vdots\\
&\omega_N \in \Omega_N && \text{observation}\\
&u_N \in \mathbb{R}^{n_N} && \text{recourse decision.}
\end{aligned}
$$

Here we have an *initial stage*, centered on the choice of u_0, followed by N *recourse stages*, each consisting of an observation and a decision which we wish to think of as capable of responding to that observation. Note that the spaces from which the decisions are to be chosen are taken as finite-dimensional but of possibly varying dimensionality. The number of stages gives a so-called *finite horizon* to the process; it doesn't go on indefinitely. When it has finished, we have a record of decisions

taken, expressible as a "supervector"

$$u := (u_0, u_1, \ldots, u_N) \in \mathbb{R}^n \quad \text{for } n := n_0 + n_1 + \cdots + n_N,$$

and a history of observations made, representable by

$$\omega := (\omega_1, \ldots, \omega_N) \in \Omega := \Omega_1 \times \cdots \times \Omega_N.$$

The outcome is in these terms a pair $(u, \omega) \in \mathbb{R}^n \times \Omega$.

Cost of outcome: Another essential ingredient is the cost ultimately associated with the outcome of this process. We take it to be given by a function f on $\mathbb{R}^n \times \Omega$, with

$$f(u, \omega) \;=\; f(u_0, u_1, \ldots, u_N, \omega_1, \ldots, \omega_N).$$

We allow this cost to be infinite, of course, since constraints can conveniently be captured in that fashion at our current level of theory. The specification of f carries with it the specification of the set

$$C(\omega) = C(\omega_1, \ldots, \omega_N) := \big\{u = (u_0, u_1, \ldots, u_N) \,\big|\, f(u, \omega) < \infty\big\}.$$

Implicitly we'll want to have $(u_0, u_1, \ldots, u_N) \in C(\omega_1, \ldots, \omega_N)$.

Convexity assumption: We suppose that $f(u, \omega)$ is convex as a function of u for each fixed ω. Then too, $C(\omega)$ is a convex set for each ω. The precise meaning of convexity in this framework of extended-real-valuedness will be explained below.

Costs and constraints in stages: Details can be added to the specification of f to account for the stage-by-stage evolution of costs and constraints. For instance, f can be expressed as a sum of terms, some of which are independent of ω or depend only on certain initial components of ω. This will be taken up shortly.

Probability structure: The space Ω is regarded as supplied with an overall probability distribution for $\omega = (\omega_1, \ldots, \omega_N)$. The individual components ω_k, thought of as random variables, *aren't* required to be independent of each other in the statistical sense. If they were, the distribution of ω in Ω would just be the "product" of the distributions of the ω_k's in the spaces Ω_k. We allow, however, for a more complicated interdependence.

Important qualification: We suppose, though, that the overall distribution of ω is fixed and unaffected by the choice of u.

Big mistake—watch out: It would be easy to fall into the trap of thinking now that the problem to be solved is that of minimizing $\mathbb{E}_\omega\{f(u_0, u_1, \ldots, u_N, \omega_1, \ldots, \omega_N)\}$ over all choices of $(u_0, u_1, \ldots, u_N)$, but no!

That would put us back in the pattern of having to choose $u = (u_0, u_1, \ldots, u_N)$ before we know $\omega = (\omega_1, \ldots, \omega_N)$, but here we are treating a pattern that's really different, where the components of u and ω are interspersed in the decision process. The recourse decision in stage k, as viewed from stage 0, must be allowed to respond to information that becomes available between stage 0 and stage k. It has to be modeled as a function of that information instead of as a lone vector.

Evolution of information: Random elements have a before-and-after character, and it's crucial to keep track of their status in any discussion. In the "before" mode, ω_k is known only in a probabilistic sense, whereas in the "after" mode it has become a fixed piece of data. The transition between these modes is called *observation*.

At the time when u_0 must be chosen, nothing about the random elements in our process has been pinned down, but in making a recourse decision u_k we have more at our disposal. At that point the elements $\omega_1, \ldots, \omega_k$ have been observed and we have to think of $\omega = (\omega_1, \ldots, \omega_N)$ as partitioned into two parts with different status:

$$(\omega_1, \ldots, \omega_k) \text{ constituting the } \textit{current information},$$
$$(\omega_{k+1}, \ldots, \omega_N) \text{ representing the } \textit{residual uncertainty}.$$

The probability space has thus been reduced to $\Omega_{k+1} \times \cdots \times \Omega_N$. The corresponding distribution for $(\omega_{k+1}, \ldots, \omega_N)$ in this space is its *conditional probability* distribution given $(\omega_1, \ldots, \omega_k)$. (Note once more that although this distribution depends in general on the preceding *observations*, it isn't permitted, under our assumptions, to depend on the preceding *decisions*.)

Effect of information on the decision process: The data environment in which the decision u_k is taken is richer than that of earlier decisions. Somehow this has to be reflected in our formulation of an optimization problem, which hasn't yet been achieved beyond listing "ingredients.".

A major step comes in understanding that we can't get very far with the naive notion that "the decisions u_0, $u_1, \ldots, u_N$ can be made optimally as we go along," or in other words, through a sort of *sequence* of optimization problems. The shortcoming there is in failing to appreciate that in every decision stage, except the last, *we can't properly optimize without taking into account the recourse opportunities we might have later.*

Anyway, our true goal is a good choice of u_0. For that, we have to put ourselves squarely in the present and handle the future through its influence on the present, not just leaving it to take care of itself.

Recourse functions: We model the ability of the decision u_k to respond to the current information $(\omega_1, \ldots, \omega_k)$ in recourse stage k by thinking of this decision not in terms of choosing a vector in $\mathbb{R}^{n_k}$, but rather in terms of choosing a *recourse function* on the space $\Omega_1 \times \cdots \times \Omega_k$:

$$u_k(\cdot) : \ (\omega_1, \ldots, \omega_k) \ \mapsto \ u_k(\omega_1, \ldots, \omega_k) \in \mathbb{R}^{n_k}.$$

Here we write $u_k(\cdot)$ rather than u_k as a way of emphasizing that we are looking at the decision in question as a function instead of a plain vector. (In getting used to the subject, that distinction could otherwise be a source of confusion.)

The adoption of the function point of view is dictated by our need to consider the kth recourse decision *from the perspective of the present*, where we are occupied still with u_0 and don't yet know which $(\omega_1, \ldots, \omega_k)$ will emerge from the future observations. In selecting a particular function $u_k(\cdot)$, we are specifying in advance exactly how we would respond to all outcomes of the first k observations!

Policies: By a *policy* we'll mean a choice of the initial decision u_0 together with a choice of recourse functions $u_1(\cdot), \ldots, u_N(\cdot)$. Such a policy will be denoted by $u(\cdot)$ and regarded as a function from Ω to $\mathbb{R}^n$, with

$$u(\omega) = (u_0, u_1(\omega_1), u_2(\omega_1, \omega_2), \ldots, u_N(\omega_1, \omega_2, \ldots, \omega_N)),$$

even though its components depend in varying degrees on $\omega = (\omega_1, \ldots, \omega_N)$. We'll denote by $\mathcal{U}$ the collection of functions $u(\cdot)$ of this type, calling it the *policy space*.

Nonanticipativity: In insisting that the $u_k(\omega)$ component of $u(\omega)$ can only depend on $(\omega_1, \ldots, \omega_k)$ and not on $(\omega_{k+1}, \ldots, \omega_N)$, we are imposing a condition on $u(\cdot)$ called nonanticipativity. Recourse decisions can react to the past, but they can't be based on knowing the future before it happens. In this terminology, the space $\mathcal{U}$ consists of all functions $u(\cdot) : \Omega \to \mathbb{R}^n$ that are *nonanticipative*.

Stochastic programming problem: The problem we come down to is that of minimizing, over all $u(\cdot)$ in the policy space $\mathcal{U}$, the expected cost

$$\begin{aligned} J\big[u(\cdot)\big] \,&:= \mathbb{E}_\omega \Big\{ f\big(u(\omega), \omega\big) \Big\} \\ &= \mathbb{E}_\omega \Big\{ f\big(u_0, u_1(\omega_1), u_2(\omega_1, \omega_2), \ldots, u_N(\omega_1, \omega_2, \ldots, \omega_N), \omega_1, \omega_2, \ldots, \omega_N\big) \Big\}. \end{aligned}$$

Constraints are implicit; a policy $u(\cdot)$ is deemed *feasible* when $J[u(\cdot)] < \infty$. This definitely requires having

$$(u_0, u_1(\omega_1), u_2(\omega_1,\omega_2), \ldots, u_N(\omega_1, \omega_2, \ldots, \omega_N)) \\ \in C(\omega_1, \omega_2, \ldots, \omega_N) \text{ almost surely,}$$

and in the case of discrete probability, at least, is fully captured by that.

Remark: The expression $f(u_0, u_1(\omega_1), u_2(\omega_1, \omega_2), \ldots, u_N(\omega_1, \omega_2, \ldots, \omega_N), \omega_1, \ldots, \omega_N)$ could, in the context of continuous probability, be finite almost surely with respect to ω, yet its integral $J[u(\cdot)]$, could nonetheless be ∞. Issues about additional assumptions that may be needed on $u(\cdot)$ for the integral to be well defined come up too—and are indicated below.

Optimization context: We are minimizing here over a particular space $\mathcal{U}$ of *functions* $u(\cdot) : \Omega \to \mathbb{R}^n$, distinguished by the special rules of dependence that go into the definition of a policy, namely *nonanticipativity*. Minimization over a function space puts us at a more advanced level of mathematics, at least conceptually, and it raises further questions about the degree to which conventional optimization methodology may be applicable. From this angle, there's a major division between discrete and continuous probability.

Case of discrete probability: In this setting, which we fall back on whenever mathematical rigor is called for in these notes, not only Ω but each of the component spaces Ω_k is supposed to be finite. Expectations are given by weighted sums rather than more complicatedly by integrals relative to a density, and there are important simplifications as well in the way we can look at recourse functions.

A function on a finite set can always be interpreted as a "supervector" in a finite-dimensional space. First consider $u_1(\cdot)$, which goes from Ω_1 to $\mathbb{R}^{n_1}$. For concreteness, let's think of the elements of Ω_1 as indexed by a superscript, say ω_1^q for $q = 1, \ldots, q_1$. To specify $u_1(\cdot)$, we merely have to specify the finitely many vectors it assumes as its values, namely $u_1(\omega_1^q)$ for $q = 1, \ldots, q_1$. Thus we have a correspondence

$$u_1(\cdot) \longleftrightarrow (u_1^1, u_1^2, \ldots, u_1^q, \ldots, u_1^{q_1})$$

in which u_1^q stands for $u_1(\omega_1^q)$. This identifies $u_1(\cdot)$ with a vector in $[\mathbb{R}^{n_1}]^{q_1}$.

Next consider $u_2(\cdot)$, which goes from $\Omega_1 \times \Omega_2$ to $\mathbb{R}^{n_1}$. Thinking of Ω_2 as consisting of $\omega_2^1, \ldots, \omega_2^{q_2}$, we can view the elements of $\Omega_1 \times \Omega_2$ as pairs $(\omega_1^q, \omega_2^{q'})$, where $q = 1, \ldots, q_1$ and $q' = 1, \ldots, q_2$. This gives us a correspondence

$$u_2(\cdot) \longleftrightarrow (u_2^{11}, u_2^{12}, \ldots, u_2^{1q_2}; u_2^{21}, u_2^{22}, \ldots, u_2^{2q_2}; \ldots; u_2^{q_1 1}, u_2^{q_1 2}, \ldots, u_2^{q_1 q_2})$$

in which $u_2^{qq'}$ stands for $u_2(\omega_1^q, \omega_2^{q'})$. With such double indexing one might think of $u_2(\cdot)$ as corresponding to a sort of matrix whose components are vectors, but it can also be thought of as a "supervector" in $[\mathbb{R}^{n_2}]^{q_1 q_2}$. This indexing procedure can obviously be extended to stage k to identify $u_k(\cdot)$ with an element of $[\mathbb{R}^{n_k}]^{q_1 q_2 \cdots q_k}$, where q_k is the number of elements in Ω_k.

Sophisticated view: There's really no need to introduce such *ordered* indexing except when it comes to manipulating data in a computer. For theoretical purposes, it's easy to get accustomed to thinking directly of the values of $u_k(\cdot)$ as indexed by the finitely many elements $(\omega_1, \ldots, \omega_k)$ themselves.

Inevitably high dimensionality: It's worth noting in this the tremendous effect of "future branching" on the size of the optimization problem under consideration. Suppose, for example, that each decision space $\mathbb{R}^{n_k}$ is modestly 3-dimensional, and each uncertain element ω_k is allowed to take on just 10 possible values; i.e., $q_k = 10$. (Perhaps ω_k has been obtained from a single random real variable by discretizing its range into 10 subintervals—deciles.) With 6 recourse stages, the dimension of the policy space $\mathcal{U}$ will be $d = 3{+}30{+}300{+}\cdots{+}3\cdot 10^6 = 3,333,333$! The challenge of solving problems in so many variables is an overriding characteristic of stochastic programming.

Case of continuous probability: When each of the spaces Ω_k is a continuum, we can't escape from the function context to finite-dimensional supervectors. Policies $u(\cdot)$ have to be treated as elements of a function space that's *infinite-dimensional*. But anyone having experience in working with function spaces is aware that many technical questions come up, not the least of which is how the space should be narrowed down. Should the functions be continuous? piecewise continuous? just "measurable"? What norm should describe convergence? And so forth.

Restrictions may already be required in making sure that the integral behind the expectation $J[u(\cdot)]$ is well defined; we won't really be able to take the policy space $\mathcal{U}$ to consist of *all* nonanticipative functions $u(\cdot)$. However, such restrictions also have a troublesome "a priori" quality in seeming to dictate properties of a solution before we know whether those properties are legitimate. For instance, if we restrict recourse functions to be continuous, because that appears convenient, we may find ourselves unable to verify that an optimal policy exists. Furthermore, we may be excluding from consideration numerical techniques that operate outside of continuity, e.g. with step functions.

Finite-dimensional approximations: Two approaches to passing from an infinite-dimensional space to a finite-dimensional one have already been mentioned. Random variables can be discretized by partitioning their domains. Alternatively, random variables can be "sampled" a number of times, and the given probability space can in that way be replaced by a sample space.

The classical means of approximation of functions aren't available, however. In much of applied mathematics a function can be replaced by a truncated Taylor series or trigonometric series, but that's not helpful for policy functions because of the need to attend to constraints and, in many situations, cope with cost discontinuities.

Convexity of costs: The expression $f(u,\omega) = f(u_0, u_1, \ldots, u_N, \omega_1, \ldots, \omega_N)$ has been assumed to be convex as a function of $u = (u_0, u_1, \ldots, u_N)$ for each $\omega = (\omega_1, \ldots, \omega_N)$, but just what does this mean in the presence of extended-real-valuedness, and what are the consequences?

Extended definition of convexity: For a function φ on $\mathbb{R}^n$ with values in $\overline{\mathbb{R}}$, convexity can be viewed equivalently in different ways. For one thing, it corresponds to the usual convexity inequality being satisfied, provided that one keeps to the extended arithmetic of $\pm\infty$ described earlier. There's nothing special to say about this except that if, for some reason, both ∞ and $-\infty$ should both pop up together, their sum is taken as ∞. (Ordinarily we don't encounter this because we know for some reason that $\varphi > -\infty$ everywhere.)

Alternatively, the convexity of φ can be identified with the convexity of the epigraph of φ, which is the set of all pairs $(u,\alpha) \in \mathbb{R}^n \times \mathbb{R}$ having $\varphi(u) \le \alpha$. Note that the α components of these pairs are always finite; by definition, no infinities enter the epigraph itself.

When φ is a convex function on $\mathbb{R}^n$, the set $\{u \,|\, \varphi(u) < \infty\}$ is a convex subset of $\mathbb{R}^n$. On the other hand, if one is given a convex subset C of $\mathbb{R}^n$ and a real-valued function φ_0 that's convex on C, then by setting $\varphi(u) = \varphi_0(u)$ for $u \in C$ but $\varphi(u) = \infty$ elsewhere, one obtains a convex function φ on all of $\mathbb{R}^n$.

Convexity in expectations: Under our convexity assumption on f, the expected cost $J[u(\cdot)]$ is convex as a function of the policy $u(\cdot) \in \mathcal{U}$.

Proof: Consider any two policies $u(\cdot)$ and $u'(\cdot)$ and any $\tau \in (0,1)$. Let $u''(\cdot) = (1-\tau)u(\cdot) + \tau u'(\cdot)$, so that the kth component of $u''(\cdot)$ is $(1-\tau)u_k(\cdot) + \tau u'_k(\cdot)$. Since $u_k(\omega)$ and $u'_k(\omega)$ depend only on $(\omega_1, \ldots, \omega_k)$, the same is true of $u''(\omega)$; this confirms that $u''(\cdot)$ is indeed another policy, i.e., satisfies the requirement

of nonanticipativity. (More generally, $\mathcal{U}$ is a *linear space* of functions—it contains all linear combinations of its various elements.) Recall now that

$$J\big[u''(\cdot)\big] = I\!E_\omega\big\{f\big(u''(\omega),\omega\big)\big\} = I\!E_\omega\big\{f\big((1-\tau)u(\omega)+\tau u'(\omega),\omega\big)\big\}.$$

By assumption $f\big((1-\tau)u(\omega)+\tau u'(\omega),\omega\big) \le (1-\tau)f\big(u(\omega),\omega\big)+\tau f\big(u'(\omega),\omega\big)$ for each ω. From this it follows that $I\!E_\omega\big\{f\big((1-\tau)u(\omega)+\tau u'(\omega),\omega\big)\big\} \le (1-\tau)\, I\!E_\omega\big\{f\big(u(\omega),\omega\big)\big\}+\tau\, I\!E_\omega\big\{f\big(u'(\omega),\omega\big)\big\}$, or in other words that we have $J[u''(\cdot)] \le (1-\tau)J[u(\cdot)]+\tau J[u'(\cdot)]$, as claimed.

Implication for optimal policies: Any locally optimal solution $\bar{u}(\cdot)$ to our problem of stochastic programming has to be globally optimal.

Two-stage stochastic programming: Problems with only two stages have long been a workhorse in stochastic programming. Such problems involve an initial decision u_0, an opportunity for gaining additional information through observation, and then a single recourse decision u_1. Time is divided starkly into just the "present" and the "future." That could make rough sense, for instance, in planning for a new facility like a shopping center or factory. Initially one has to decide on its location, scale, etc., but only later does one have to decide how best to operate it.

Beyond the practicality of its applications, two-stage stochastic programming has been—and continues to be—important as a testing ground for concepts and solution techniques. It will serve those purposes here as well.

Specialized formulation: In dealing with just one uncertain element, it would be tedious to denote it by ω_1 and speak of it lying in a space Ω_1, in the manner of our general notation. We'll simply work with ω and Ω. At the same time, though, we'll introduce more structure in costs and constraints. We take the decision-observation process to be:

$$\begin{aligned}&\text{choose } u_0 \in U_0 \subset I\!R^{n_0},\ \text{ paying } f_0(u_0),\\ &\text{observe } \omega \in \Omega,\\ &\text{choose } u_1 \in U_1 \subset I\!R^{n_1},\ \text{ paying } f_1(u_0,u_1,\omega).\end{aligned}$$

The functions f_0 and f_1 are still extended-real-valued; the "payments" could be ∞, but we suppose these functions don't take on $-\infty$. We'll eventually look at how constraints and penalties may enter their definitions.

It should be clear that we're merely specializing our general stochastic programming model to the case of $N = 1$, $\omega = \omega_1$, and total cost

$$f(u_0,u_1,\omega) = \begin{cases} f_0(u_0)+f_1(u_0,u_1,\omega) & \text{if } u_0 \in U_0 \text{ and } u_1 \in U_1 \\ \infty & \text{otherwise.}\end{cases}$$

This cost formula can be written compactly in terms of functions δ_{U_0} and δ_{U_1} which enforce the set membership through infinite penalties: $\delta_{U_0}(u_0) = 0$ if $u_0 \in U_0$ but $\delta_{U_0}(u_0) = \infty$ if $u_0 \notin U_0$, and similarly for $\delta_{U_1}(u_1)$. It comes out then as

$$f(u_0, u_1, \omega) \;=\; f_0(u_0) + \delta_{U_0}(u_0) + f_1(u_0, u_1, \omega) + \delta_{U_1}(u_1).$$

Indicators: The functions δ_{U_0} and δ_{U_1} are called the *indicators* of U_0 and U_1.

Convexity: To be sure that $f(u_0, u_1, \omega)$ is convex with respect to (u_0, u_1), we assume that the sets U_0 and U_1 are convex, that $f_0(u_0)$ is convex with respect to u_0, and that $f_1(u_0, u_1, \omega)$ is convex with respect to (u_0, u_1).

Implicit constraints: In order to avoid invoking an infinite penalty, u_0 must belong to the *initial set*

$$C_0 := \big\{u_0 \in U_0 \,\big|\, f_0(u_0) < \infty\big\},$$

while u_1 must belong to the *recourse set*

$$C_1(u_0, \omega) := \big\{u_1 \in U_1 \,\big|\, f_1(u_0, u_1, \omega) < \infty\big\}.$$

Under our convexity assumptions, these sets that describe feasibility are convex subsets of $\mathbb{R}^{n_0}$ and $\mathbb{R}^{n_1}$, respectively.

Two-stage policies and optimization: In this setting a policy $u(\cdot)$ is a pair $(u_0, u_1(\cdot))$, where $u_0 \in \mathbb{R}^{n_0}$ and $u_1(\cdot)$ is a function from Ω to $\mathbb{R}^{n_1}$. Its expected total cost is

$$\begin{aligned} J[u(\cdot)] = J[u_0, u_1(\cdot)] &= \mathbb{E}_\omega\big\{f(u_0, u_1(\omega), \omega)\big\} \\ &= f_0(u_0) + \delta_{U_0}(u_0) + \mathbb{E}_\omega\big\{f_1(u_0, u_1(\omega), \omega) + \delta_{U_1}(u_1(\omega))\big\}. \end{aligned}$$

The stochastic programming problem we wish to solve is

$$(\mathcal{P}) \qquad \text{minimize } J[u(\cdot)] \text{ over the space } \mathcal{U} \text{ of policies } u(\cdot) = (u_0, u_1(\cdot)).$$

Feasibility: The feasibility of a policy $u(\cdot) = (u_0, u_1(\cdot))$, i.e., the property $J[u(\cdot)] < \infty$, necessitates having $u(\cdot)$ satisfy the conditions

$$\begin{cases} u_0 \in C_0, \\ u_1(\omega) \in C_1(u_0, \omega) \text{ almost surely.} \end{cases}$$

Here C_0 and $C_1(u_0, \omega)$ are the feasible sets for the costs in the two stages, as introduced above. In the case of discrete probability, the feasibility of $u(\cdot)$ is *equivalent* to these conditions being satisfied.

Projected costs: It's instructive, and eventually very fruitful in seeing how two-stage stochastic programming problems can be solved, to investigate how costs in the recourse stage can be projected back to the initial stage. This requires first looking at the recourse stage *parametrically*.

Recourse subproblem: Once we have chosen u_0 and observed ω, the optimization problem in u_1 that we are confronted with has the form

$$(\mathcal{P}_1(u_0,\omega)) \qquad \text{minimize } f_1(u_0,u_1,\omega) \text{ over } u_1 \in U_1.$$

In terms of u_0 and ω as parameters, let's denote by $\varphi(u_0,\omega)$ the optimal value in this problem and by $\Phi(u_0,\omega)$ its optimal solution set; thus,

$$\varphi(u_0,\omega) := \inf_{u_1\in U_1} \big\{f_1(u_0,u_1,\omega)\big\},$$
$$\Phi(u_0,\omega) := \operatorname*{argmin}_{u_1\in U_1} \big\{f_1(u_0,u_1,\omega)\big\}.$$

Of course, we haven't yet introduced assumptions that would tell us whether the optimal value is finite or that an optimal solution actually exists.

Argmin notation: In general, "argmin" refers to the minimizing arguments, i.e., the points where the minimum in question is attained, whereas "inf" refers only to the greatest lower bound of the expression being minimized, regardless of whether or not it's attained. Because of our way of using ∞, however, there's a small exception we need to make—when inf $= \infty$. In that case there are no points that give a value less than ∞ in the minimization, so there are no points considered to be feasible. We don't want to regard any points as optimal then either, so we prefer to think of the "argmin" set in that case as empty. The convention is therefore followed that argmin $= \emptyset$ when inf $= \infty$.

Projected problem: Relative to any choice of u_0, but prior to the observation of ω, the optimal value $\varphi(u_0,\omega)$ in the recourse subproblem is a random variable. Its expected value

$$\widehat{\varphi}(u_0) := I\!E_\omega\big\{\varphi(u_0,\omega)\big\}$$

is the *projected cost of recourse*, from the perspective of the initial stage. It tells us all that's essential about the future effects of u_0. It focuses us on solving

$$(\mathcal{P}_0) \qquad \text{minimize } f_0(u_0) + \widehat{\varphi}(u_0) \text{ over } u_0 \in U_0.$$

How does this projected problem, in u_0 alone, coordinate with solving the original two-stage problem $(\mathcal{P})$?

Note: In the literature of two-stage stochastic programming, the projected problem is often called the "deterministic equivalent problem" because it no longer involves a random element. This terminology may be misleading, however, because the problem in question is fully based on stochastic modeling and not in any way on getting rid of the stochastic structure and passing to a deterministic model in its place.

Characterization of an optimal policy: The optimal value in the original problem $(\mathcal{P})$ equals the optimal value in the projected problem $(\mathcal{P}_0)$. As long as this is finite, a policy $\bar{u}(\cdot) = (\bar{u}_0, \bar{u}_1(\cdot))$ is optimal in $(\mathcal{P})$ if and only if

$$\begin{cases} \bar{u}_0 \text{ is an optimal solution to } (\mathcal{P}_0), \text{ and} \\ \bar{u}_1(\omega) \in \Phi_1(\bar{u}_0, \omega) \text{ almost surely in } \omega. \end{cases}$$

Argument: Denote the optimal values in $(\mathcal{P})$ and $(\mathcal{P}_0)$ by $\bar{\alpha}$ and $\bar{\alpha}_0$. Consider any policy $u(\cdot) = (u_0, u_1(\cdot))$ with $f_0(u_0) < \infty$, $u_0 \in U_0$ and $u_1(\omega) \in U_1$. We have

$$f_1(u_0, u_1(\omega), \omega) \;\geq\; \varphi(u_0, \omega),$$

where the condition $u_1(\omega) \in \Phi(u_0, \omega)$ characterizes the case where the two sides are equal and not ∞. Taking the expectation on each side and adding $f_0(u_0)$ to each, we obtain

$$J[(u(\cdot)] \;\geq\; f_0(u_0) + \widehat{\varphi}(u_0) \;\geq\; \bar{\alpha}_0.$$

By varying the $u_1(\cdot)$ part of the policy $u(\cdot)$ with the u_0 part fixed, we can make the value on the left as close as we please to the value in the middle; when $\widehat{\varphi}(u_0) < \infty$, the two are equal if and only if $u_1(\omega) \in \Phi(u_0, \omega)$ almost surely. (When $\widehat{\varphi}(u_0) = \infty$, they both must be ∞.) On the other hand, by varying u_0 we can bring the middle value down to $\bar{\alpha}_0$. To say that u_0 is an optimal solution to $(\mathcal{P}_0)$ is to say that these values are equal and not ∞. Thus, the infimum $\bar{\alpha}$ of $J[u(\cdot)]$ over all feasible policies equals $\bar{\alpha}_0$ and, when not ∞, is attained precisely by the policies $\bar{u}(\cdot)$ of the kind stipulated.

Rigor: Beyond discrete probability, where only finite sums are involved in taking an expectation, more care would need to be exercised in arguing about what happens when the recourse function $u_1(\cdot)$ is varied. The difficulty arises from the need to stay within the category of "measurable" recourse functions. Further assumptions about f_1 have to be added for this purpose.

Convexity of projected costs: The infimum $\varphi(u_0, \omega)$ is convex with respect to u_0, and the same holds for its expected value $\widehat{\varphi}(u_0)$.

Proof: The second assertion follows from the first, as noted earlier. The first assertion is based on the general fact that *when a convex function of two vector variables is minimized in one of these variables, the optimal value, as a function of the other variable is again a convex function.* We'll supply the argument now in the special notation at hand; ω is fixed throughout.

Consider any two points u_0 and u_0' and any $\lambda \in (0,1)$. Let $u_0'' := (1-\lambda)u_0 + \lambda u_0'$. It will suffice to demonstrate for arbitrary $\alpha > \varphi(u_0,\omega)$ and $\alpha' > \varphi(u_0',\omega)$ that the value $\alpha'' := (1-\lambda)\alpha + \lambda\alpha'$ satisfies $\alpha'' \geq \varphi(u_0'',\omega)$.

Since $\varphi(u_0,\omega) < \alpha$, there exists by definition of the function φ some $u_1 \in U_1$ with $f_1(u_0,u_1,\omega) < \alpha$. Likewise, there exists some $u_1' \in U_1$ with $f_1(u_0',u_1',\omega) < \alpha'$. Then, in terms of $u_1'' := (1-\lambda)u_1 + \lambda u_1'$, which still belongs to U_1 by the convexity assumed for that set, we have by the convexity of f_1 in its first two arguments that

$$\begin{aligned}(1-\lambda)\alpha + \lambda\alpha' \;&>\; (1-\lambda)f_1(u_0,u_1,\omega) + \lambda f_1(u_0',u_1',\omega)\\ &\geq\; f_1(u_0'',u_1'',\omega) \;\geq\; \varphi(u_0'',\omega),\end{aligned}$$

and since the first expression equals α'' this chain gives us what we want.

Implications for two-stage stochastic programming: The projected problem $(\mathcal{P}_0)$ is one of *convex* optimization. It differs from classical problems of convex programming primarily in the nature of the function $\widehat{\varphi}$ and how it might be handled, and in that respect the convexity of $\widehat{\varphi}$ is an extremely important feature. By approximating $\widehat{\varphi}$ in some manner or another, one can hope to approximate $(\mathcal{P}_0)$ and in that way come close to determining an optimal initial decision $\bar{u}_0$.

Projected constraints: A vector u_0 is a feasible solution to the projected problem $(\mathcal{P}_0)$ if and only if it satisfies $u_0 \in U_0$, $f_0(u_0) < \infty$, and $\widehat{\varphi}(u_0) < \infty$. The first and second of these conditions are known directly from the data in the original problem $(\mathcal{P})$ and have already been lumped together in the notation $u_0 \in C_0$. The third condition, however, is only known indirectly as a by-product of analyzing the recourse subproblem $(P_1(u_0,\omega))$ parametrically. It doesn't itself involve uncertainty, but it arises out of future uncertainty. It necessitates having $u_0 \in \widehat{D}$, where

$$\begin{aligned}\widehat{D} &:= \big\{u_0 \,\big|\, u_0 \in D(\omega) \text{ almost surely}\big\} \text{ for}\\ D(\omega) &:= \big\{u_0 \,\big|\, \exists\, u_1 \in U_1 \text{ with } f_1(u_0,u_1,\omega) < \infty\big\},\end{aligned}$$

and it is equivalent to this in the context of discrete probability, in which $\widehat{D}$ is sure to be the effective domain of $\widehat{\varphi}$. Note that $\widehat{D}$ is a convex set, so the condition $u_0 \in \widehat{D}$ is a convex type of constraint.

Relatively complete recourse: This term refers to the case where the implicit constraint $\widehat{\varphi}(u_0) < \infty$ in $(\mathcal{P}_0)$ imposes no additional restriction, i.e., where

$$\widehat{\varphi}(u_0) < \infty \text{ for all } u_0 \in C_0.$$

So-called *complete recourse* requires $\widehat{\varphi}(u_0) < \infty$ for *all* $u_0 \in \mathbb{R}^{n_0}$.

Trouble otherwise: Without relatively complete recourse, the projected problem can be much harder to solve. Numerical methods are harder to set up, and they waste considerable effort just in trying to ensure feasibility. It's important therefore to build this property into a problem's formulation if possible.

The role of penalty expressions: In order to achieve relatively complete recourse in the formulation of a problem, it's essential to be careful about constraints in the recourse stage. Appropriate penalties should be introduced when an inequality or equality constraint might create uncertainty about recourse feasibility.

Exploring this more closely, we see that the recourse subproblem $(\mathcal{P}_1(u_0, \omega))$ will generally need to be viewed in a form like the following:

$$\text{minimize } g_0(u_0, u_1, \omega) + \textstyle\sum_{i=1}^{r} \theta_i\big(g_i(u_0, u_1, \omega)\big) \text{ subject to}$$
$$u_1 \in U_1 \text{ and } g_i(u_0, u_1, \omega) \begin{cases} \le 0 & \text{for } i = r+1, \dots, s, \\ = 0 & \text{for } i = s+1, \dots, m, \end{cases}$$

which corresponds to taking

$$f_1(u_0, u_1, \omega) = \begin{cases} g_0(u_0, u_1, \omega) + \sum_{i=1}^{r} \theta_i\big(g_i(u_0, u_1, \omega)\big) \text{ when} \\ \qquad\qquad g_i(u_0, u_1, \omega) \begin{cases} \le 0 & \text{for } i = r+1, \dots, s, \\ = 0 & \text{for } i = s+1, \dots, m, \end{cases} \\ \qquad\qquad u_1 \in U_1, \\ \infty \qquad \text{otherwise.} \end{cases}$$

Relatively complete recourse will be present if for each $u_0 \in U_0$ with $f_0(u_0) < \infty$ there exists, almost surely with respect to ω, a $u_1 \in U_1$ giving a finite value to the expression $g_0(u_0, u_1, \omega) + \sum_{i=1}^{r} \theta_i\big(g_i(u_0, u_1, \omega)\big)$ and satisfying the listed inequalities and equations on the $g_i(u_0, u_1, \omega)$'s for $i = r+1, \dots, m$.

Typically that would be because the objective expression is finite for all $u_0 \in U_0$, $u_1 \in U_1$ and $\omega \in \Omega$, and the inequality and equality constraints can be always fulfilled. (For instance, the components of u_1 involved in these constraints might merely be "slack variables" which don't enter any other expressions.)

Convexity considerations: The required convexity of $f_1(u_0, u_1, \omega)$ in (u_0, u_1) is assured if all expressions $g_i(u_0, u_1, \omega)$ are convex in (u_0, u_1) and the functions θ_i are convex and *nondecreasing*. When $g_i(u_0, u_1, \omega)$ is affine in (u_0, u_1), the associated θ_i doesn't need to be nondecreasing; its convexity is enough.

3. LINEAR PROGRAMMING MODELS

Stochastic programming problems may appear to be so complicated that there is little hope of solving them, but important progress has been made. Naturally, when people first put their minds to devising numerical methods they concentrated on two-stage models that were amenable to the use of techniques in large-scale linear programming. We'll begin there as well.

In examining the ideas behind such methods, we'll have opportunities to gain insights into aspects of optimization beyond stochastic programming. We'll see how duality can be put to work. We'll also look at "subgradients" of convex functions and the ways they can substitute for gradients in the absence of differentiability.

Two-stage stochastic linear programming: The two-stage stochastic programming theory that we've been examining will be applied now to the case where each stage fits the traditional patterns of linear programming.

Description: In terms of vector inequalities, we take the process to have the form

choose $u_0 \geq 0$ with $A_{00}u_0 \geq b_0$, paying $c_0 \cdot u_0$,

observe $\omega \in \Omega$,

choose $u_1 \geq 0$ with $A_{10}(\omega)u_0 + A_{11}(\omega)u_1 \geq b_1(\omega)$, paying $c_1(\omega) \cdot u_1$.

Here $u_0 \in \mathbb{R}^{n_0}$ and $u_1 \in \mathbb{R}^{n_1}$, so the conditions $u_0 \geq 0$ and $u_1 \geq 0$ are shorthand for $u_0 \in \mathbb{R}^{n_0}_+$ and $u_1 \in \mathbb{R}^{n_1}_+$. We think of A_{00} and b_0 as belonging to $\mathbb{R}^{m_0 \times n_0}$ and $\mathbb{R}^{m_0}$, so that the condition $A_{00}u_0 \geq b_0$ summarizes m_0 linear inequality constraints on u_0. Likewise we take $A_{10}(\omega) \in \mathbb{R}^{m_1 \times n_0}$, $A_{11}(\omega) \in \mathbb{R}^{m_1 \times n_1}$ and $b_1(\omega) \in \mathbb{R}^{m_1}$, so that the condition $A_{10}(\omega)u_0 + A_{11}(\omega)u_1 \geq b_1(\omega)$ summarizes m_1 additional constraints that come into play only in the recourse stage.

In the scheme previously adopted for two-stage stochastic programming, U_0 and U_1 are the orthants $\mathbb{R}^{n_0}_+$ and $\mathbb{R}^{n_1}_+$, $f_0(u_0)$ equals $c_0 \cdot u_0$ plus the indicator of the set of points u_0 satisfying $A_{00}u_0 \geq b_0$, and $f_1(u_0, u_1, \omega)$ equals $c_1(\omega) \cdot u_1$ plus the indicator of the set of points u_1 satisfying $A_{10}(\omega)u_0 + A_{11}(\omega)u_1 \geq b_1(\omega)$.

Feasiblity in the two stages: The feasible set C_0 associated with the initial stage specializes here to the polyhedral set

$$C_0 = \big\{u_0 \geq 0 \,\big|\, A_{00}u_0 \geq b_0\big\},$$

whereas the feasible set $C_1(u_0, \omega)$ for the recourse stage is the polyhedral set

$$C_1(u_0, \omega) = \big\{u_1 \geq 0 \,\big|\, A_{11}(\omega)u_1 \geq b_1(\omega) - A_{10}(\omega)u_0\big\}.$$

Problem statement: As always in two-stage stochastic programming, we optimize over policies $u(\cdot) = (u_0, u_1(\cdot))$. Because the expected cost $J[u(\cdot)]$ is ∞ unless $u_0 \in C_0$ and $u_1(\omega) \in C_1(u_0, \omega)$ almost surely, but otherwise is given by an expression depending only on the linear cost terms in the two stages, we can state the problem as:

$$(\mathcal{P}) \qquad \begin{array}{l} \text{minimize } c_0 \cdot u_0 + I\!E_\omega\{c_1(\omega) \cdot u_1(\omega)\} \text{ subject to} \\ \begin{cases} u_0 \geq 0,\ A_{00}u_0 \geq b_0, \\ u_1(\omega) \geq 0,\ A_{10}(\omega)u_0 + A_{11}(\omega)u_1(\omega) \geq b_1(\omega) \text{ almost surely.} \end{cases} \end{array}$$

Comments on the constraint setup: Equality constraints could be incorporated along with, or instead of, inequalities, but we've left this out to keep the notation more manageable. A bigger issue is why inequality or equality constraints are allowed to depend on ω at all, in light of the earlier criticism of such uncertain constraints. Shouldn't penalty expressions be brought in?

The answer is "historic." This is the format suggested by linear programming, and it's therefore the one that people were led to adopt. Linear programming doesn't allow for penalty expressions—directly. But such expressions, if of piecewise linear type, can often be converted to the linear programming format through the introduction of additional variables, and so forth. That's in fact what's usually behind the circumstances in which one is able, in the current context, to claim relatively complete recourse.

Comments on the depiction of uncertainty: In writing $A_{10}(\omega)$, $A_{11}(\omega)$, $b_1(\omega)$ and $c_1(\omega)$, we indicate that these arrays might not be fully known initially, when we have to choose u_0. The "observation of ω" is code language for getting the information needed to fill in the knowledge gaps. We are supposing that the information will be available before the deadline for choosing u_1.

At one extreme, every component of each array might be a random variable—a total of $d = m_1n_0 + m_1n_1 + m_1 + n_1$ such variables. Those random variables could to some degree be independent of each other, but in principle might have a joint probability distribution. We could think of them as comprising a "supervector" which ranges randomly over a region in $I\!R^d$. The region could then be identified with Ω, and the supervector itself with ω.

More commonly, only some of the components might be uncertain. Perhaps ω just refers to those, or ω is a vector of two or three random variables (e.g. current

interest rates) which aren't themselves components of $A_{10}(\omega)$, $A_{11}(\omega)$, $b_1(\omega)$ or $c_1(\omega)$, but enter the formulas for several of the components.

Random right sides only: Especially noteworthy is the case where the matrices and costs in the recourse stage are certain, hence expressible as A_{10}, A_{11}, and c_1, so that only $b_1(\omega)$ is uncertain. Much of the early literature of stochastic programming was devoted to that, with $b_1(\omega)$ as a vector of demands, say.

Recourse subproblems: With u_0 and ω as parameters, the problem faced in the second stage has the form

$$(\mathcal{P}_1(u_0,\omega)) \qquad \begin{array}{l} \text{minimize } \; c_1(\omega)\cdot u_1 \;\; \text{subject to} \\ \qquad\qquad u_1 \geq 0, \;\; A_{11}(\omega)u_1 \;\geq\; b_1(\omega) - A_{10}(\omega)u_0. \end{array}$$

The effects of u_0 are felt only through the term $A_{10}(\omega)u_0$, which has been moved to the right side of the vector inequality in order to make this role clearer. If $c_1(\omega)$ and $A_{11}(\omega)$ didn't really depend on ω, we would have a parametric family of linear programming problems which differ only in these right sides.

Projected problem: The projected problem in u_0 alone comes out in the form

$$(\mathcal{P}_0) \qquad \text{minimize } \; c_0\cdot u_0 + \widehat{\varphi}(u_0) \;\; \text{subject to } \; u_0 \geq 0, \;\; A_{00}u_0 \geq b_0,$$

where the projected cost is defined by

$$\widehat{\varphi}(u_0) := I\!E_\omega\big\{\varphi(u_0,\omega)\big\} \;\; \text{ for } \;\; \varphi(u_0,\omega) := \inf\big(P_1(u_0,\omega)\big).$$

We'll have *relatively complete* recourse available if this cost $\widehat{\varphi}(u_0)$ is finite for all u_0 satisfying the initial linear constraints $u \geq 0$ and $A_{00}u_0 \geq b_0$, and *complete* recourse if it's finite even when u_0 doesn't satisfy those constraints.

Convexity: This two-stage linear programming model meets the general convexity requirements in stochastic programming, so the function $\widehat{\varphi}$ is convex. In the case of relatively complete recourse, and even better in the case of complete recourse, problem $(\mathcal{P}_0)$ is thus one of minimizing a convex function subject to initial linear constraints only.

Expression of the overall problem as large-scale linear programming: Before getting down to the close scrutiny of the projected cost function $\widehat{\varphi}$ that's crucial to the most popular approaches to computation in the projected problem $(\mathcal{P}_0)$, it's instructive to look at how the overall problem $(\mathcal{P})$ can itself be viewed as large-scale

linear programming. For this purpose we concentrate on discrete probability. We take

$$\Omega = \{\omega^q \text{ for } q = 1, \ldots, \bar{q}\}, \qquad \pi^q = \pi(\omega^q),$$

and adopt in place of $u_1(\omega)$, $b_1(\omega)$, $c_1(\omega)$, $A_{10}(\omega)$ and $A_{11}(\omega)$ the compact notation

$$u_1^q = u_1(\omega^q), \quad b_1^q = b_1(\omega^q), \quad c_1^q = c_1(\omega^q), \quad A_{10}^q = A_{10}(\omega^q), \quad A_{11}^q = A_{11}(\omega^q).$$

The specification of a recourse function $u_1(\cdot)$ is tantamount then to the specification of a family of vectors u_1^q for $q = 1, \ldots, \bar{q}$.

The objective and constraints in $(\mathcal{P})$ can be rewritten in this notation, but we have to be careful about the conditions that only need to be satisfied "almost surely." A distinction is being made between elements ω^q with $\pi^q > 0$ and those with $\pi^q = 0$. We could take the position that $\pi^q > 0$ always (else why include w^q in Ω?), but it will be better just to multiply all the linear constraints in the recourse stage by their probability. That has the effect of canceling out the constraints of 0 probability without forcing us to delete elements from Ω. The restated version of the overall stochastic problem $(\mathcal{P})$ is then

$$(\mathcal{P}') \qquad \begin{array}{l} \text{minimize } c_0 \cdot u_0 + \sum_{q=1}^{\bar{q}} \pi^q c_1^q \cdot u_1^q \\ \text{subject to } \begin{cases} u_0 \geq 0, \;\; A_{00} u_0 \geq b_0, \\ u_1^q \geq 0, \;\; \pi^q A_{10}^q u_0 + \pi^q A_{11}^q u_1^q \geq \pi^q b_1^q \text{ for } q = 1, \ldots, \bar{q}. \end{cases} \end{array}$$

Representation in a tableau: In the tableau format for linear programming problems, which assists with determining dual problems while helping to reveal patterns of matrix sparsity, $(\mathcal{P}')$ has the schematic representation:

$$\begin{array}{ccccccccc} [& u_0 & u_1^1 & \ldots & u_1^q & \ldots & u_1^{\bar{q}} &] & \geq & 0 \\ \left[\begin{array}{c} \\ A_{00} \\ \pi^1 A_{10}^1 \\ \vdots \\ \pi^q A_{10}^q \\ \vdots \\ \pi^{\bar{q}} A_{10}^{\bar{q}} \end{array}\right. & & \begin{array}{c} \\ 0 \\ \pi^1 A_{11}^1 \\ \\ 0 \\ \\ 0 \end{array} & & \begin{array}{c} \\ 0 \\ 0 \\ \vdots \\ \pi^q A_{11}^q \\ \vdots \\ 0 \end{array} & & \left.\begin{array}{c} \\ 0 \\ 0 \\ \\ 0 \\ \\ \pi^{\bar{q}} A_{11}^{\bar{q}} \end{array}\right] & \geq & \left[\begin{array}{c} b_0 \\ \pi^1 b_1^1 \\ \vdots \\ \pi^q b_1^q \\ \vdots \\ \pi^{\bar{q}} b_1^{\bar{q}} \end{array}\right] \\ [& c_0 & \pi^1 c_1^1 & \ldots & \pi^q c_1^q & \ldots & \pi^{\bar{q}} c_1^{\bar{q}} &] & \rightarrow & \min \end{array}$$

Potential for direct computation: The matrix sparsity in $(\mathcal{P}')$ suggests that advanced implementations of the simplex method, or some other technique for solving linear programming problems, might be used to solve $(\mathcal{P}')$ and hence $(\mathcal{P})$.

That's indeed how people first looked at two-stage stochastic linear programming. Efforts were made to take advantage of the particular matrix pattern exhibited here. This goes on today, but the alternative approach of solving $(\mathcal{P})$ by way of the projected problem and approximations of the projected cost function $\widehat{\varphi}$, offers a framework in which the same linear programming ideas can be viewed more insightfully, without being obscured by algorithm-specific details. It relies on duality, which we'll pursue now on more than one level.

Duality: The corresponding dual problem of linear programming can be derived from this tableau by associating a "multiplier" with each row of the A matrix. The multipliers are in this case *vectors*, since the rows give *vector* constraints; we'll denote them by v_0 (for the initial row) and v_1^q (for the recourse rows). Instead of keeping with rows, however, it will be more convenient to pass to the transpose tableau, in which the rows have turned into columns. Using $*$ to indicate the transposes of the various submatrices, we get the dual scheme

$$\begin{array}{ccccccccc}
[& v_0 & v_1^1 & \dots & v_1^q & \dots & v_1^{\bar q} &] & \geq & 0 \\
\left[\begin{array}{ccccccc}
A_{00}^* & \pi^1 A_{10}^{1*} & & \pi^q A_{10}^{q*} & & \pi^{\bar q} A_{10}^{\bar q *} \\
0 & \pi^1 A_{11}^{1*} & & 0 & & 0 \\
 & & & \vdots & & \\
0 & 0 & \dots & \pi^q A_{11}^{q*} & \dots & 0 \\
 & & & \vdots & & \\
0 & 0 & & 0 & & \pi^{\bar q} A_{11}^{\bar q *}
\end{array}\right] & & & & & & & \leq & \left[\begin{array}{c} c_0 \\ \pi^1 c_1^1 \\ \vdots \\ \pi^q c_1^q \\ \vdots \\ \pi^{\bar q} c_1^{\bar q} \end{array}\right] \\
[& b_0 & \pi^1 b_1^1 & \dots & \pi^q b_1^q & \dots & \pi^{\bar q} b_1^{\bar q} &] & \to & \max
\end{array}$$

The dual linear programming problem represented in this tableau can be read off now as:

$$(\mathcal{D}') \qquad \begin{array}{l} \text{maximize } \ b_0 \cdot v_0 + \sum_{q=1}^{\bar q} \pi^q b_1^q \cdot v_1^q \\ \text{subject to } \ \begin{cases} v_0 \geq 0, \ \ A_{00}^* v_0 + \sum_{q=1}^{\bar q} \pi^q A_{10}^{q*} v_1^q \leq c_0, \\ v_1^q \geq 0, \ \ \pi^q A_{11}^{q*} v_1^q \leq \pi^q c_1^q \ \text{ for } \ q = 1, \dots, \bar q. \end{cases} \end{array}$$

This dual problem can also be posed in the original notation of $(\mathcal{P})$ if we interpret the vectors v_1^q as the values $v_1(\omega^q)$ of a function $v(\cdot) : \Omega \to \mathbb{R}^{m_1}$. It comes out then in the form:

$$(\mathcal{D}) \qquad \begin{array}{l} \text{maximize } \ b_0 \cdot v_0 + \mathbb{E}_\omega \big\{ b_1(\omega) \cdot v_1(\omega) \big\} \\ \text{subject to } \ \begin{cases} v_0 \geq 0, \ \ A_{00}^* v_0 + \mathbb{E}_\omega \big\{ A_{10}^*(\omega) v_1(\omega) \big\} \leq c_0, \\ v_1(\omega) \geq 0, \ \ A_{11}^*(\omega) v_1(\omega) \leq c_1(\omega) \ \text{ almost surely.} \end{cases} \end{array}$$

Comments: Interestingly enough, this dual problem $(\mathcal{D})$ resembles the primal problem $(\mathcal{P})$, in that the optimization takes place with respect to pairs $v(\cdot) = (v_0, v_1(\cdot))$. The objective functions look similar as well. The constraints follow quite a different pattern, though.

In $(\mathcal{D})$, the constraints tied to the recourse stage don't involve v_0 at all, just $v_1(\omega)$. On the other hand, the constraints in the initial stage involve $v_1(\cdot)$ together with v_0 and, more strikingly, place a bound on a vector of *expectations*. To the extent that multiplier vectors can be interpreted as price vectors, it's evident that we must be dealing, in the initial stage, with a vector of expected future prices, but little more can be said outside the framework of a specific application of the model.

Theoretical consequences: From general linear programming duality, we know that problems $(\mathcal{P}')$ and $(\mathcal{D}')$, or equivalently $(\mathcal{P})$ and $(\mathcal{D})$ (as long as we keep to *discrete* probability), are tightly connected:

> *The optimal values* $\inf(\mathcal{P})$ *and* $\sup(\mathcal{D})$ *are equal except in the remote case where both* $\inf(\mathcal{P}) = \infty$ *and* $\sup(\mathcal{D}) = -\infty$*, i.e., neither of the problems has a feasible solution. Moreover, as long as the common optimal value is finite, both problems are sure to have an optimal solution.*

Lagrangian: Optimal solutions $\bar{u}(\cdot)$ to $(\mathcal{P})$ and $\bar{v}(\cdot)$ to $(\mathcal{D})$ can jointly be characterized in terms of the Lagrangian. In the general notation of these problems, the Lagrangian comes out as

$$\begin{aligned} L\big(u(\cdot), v(\cdot)\big) &= c_0 \cdot u_0 + I\!E_\omega\big\{c_1(\omega) \cdot u_1(\omega)\big\} + v_0 \cdot \big[b_0 - A_{00}u_0\big] \\ &\qquad + I\!E_\omega\big\{v_1(\omega) \cdot [b_1(\omega) - A_{10}(\omega)u_0 - A_{11}(\omega)u_1(\omega)]\big\} \\ &= b_0 \cdot v_0 + I\!E_\omega\big\{b_1(\omega) \cdot v_1(\omega)\big\} + u_0 \cdot \big[c_0 - A_{00}^* v_0 - I\!E_\omega\big\{A_{10}^*(\omega)v_1(\omega)\big\}\big] \\ &\qquad + I\!E_\omega\big\{u_1(\omega) \cdot [c_1(\omega) - A_{11}^*(\omega)v_1(\omega)]\big\} \end{aligned}$$

on the set $U \times V \subset \mathcal{U} \times \mathcal{V}$, where

$$\begin{aligned} U &= \big\{u(\cdot) = (u_0, u_1(\cdot)) \,\big|\, u_0 \in I\!R_+^{n_0},\ u_1(\omega) \in I\!R_+^{n_1} \text{ almost surely}\big\} \\ V &= \big\{v(\cdot) = (v_0, v_1(\cdot)) \,\big|\, v_0 \in I\!R_+^{m_0},\ v_1(\omega) \in I\!R_+^{m_1} \text{ almost surely}\big\}. \end{aligned}$$

Optimality conditions: In our setting of discrete probability, we can conclude from linear programming theory that $\bar{u}(\cdot)$ and $\bar{v}(\cdot)$ are optimal solutions to $(\mathcal{P})$ and $(\mathcal{D})$ if and only if they furnish a saddle point of L on $U \times V$ in the sense of having

$$\bar{u}(\cdot) \in \operatorname*{argmin}_{u(\cdot) \in U} L\big(u(\cdot), \bar{v}(\cdot)\big), \qquad \bar{v}(\cdot) \in \operatorname*{argmax}_{v(\cdot) \in V} L\big(\bar{u}(\cdot), v(\cdot)\big).$$

Furthermore, this is equivalent to the following conditions of *complementary slackness* being fulfilled:

$$
\begin{aligned}
&v_0 \geq 0, \quad A_{00}u_0 - b_0 \geq 0, \quad v_0 \cdot \big[A_{00}u_0 - b_0\big] = 0,\\
&v_1(\omega) \geq 0, \quad A_{10}(\omega)u_0 + A_{11}(\omega)u_1(\omega) - b_1(\omega) \geq 0,\\
&\qquad v_1(\omega) \cdot \big[A_{10}(\omega)u_0 + A_{11}(\omega)u_1(\omega) - b_1(\omega)\big] = 0 \text{ almost surely},\\
&u_0 \geq 0, \quad c_0 - A_{00}^* v_0 - I\!E_\omega\big\{A_{10}^*(\omega)v_1(\omega)\big\} \geq 0,\\
&\qquad u_0 \cdot \big[c_0 - A_{00}^* v_0 - I\!E_\omega\big\{A_{10}^*(\omega)v_1(\omega)\big\}\big] = 0,\\
&u_1(\omega) \geq 0, \quad c_1(\omega) - A_{11}^*(\omega)v_1(\omega) \geq 0,\\
&\qquad u_1(\omega) \cdot \big[c_1(\omega) - A_{11}^*(\omega)v_1(\omega)\big] = 0 \text{ almost surely}.
\end{aligned}
$$

Recourse analysis: In our discrete probability notation we have, with respect to any choice of the initial vector u_0, just a *finite* family of recourse subproblems $(\mathcal{P}(u_0, \omega^q))$ for determining appropriate responses u_1^q. We can write these subproblems as

$$\big(\mathcal{P}_1^q(u_0)\big) \qquad \text{minimize } \; c_1^q \cdot u_1^q \; \text{ subject to } \; u_1^q \geq 0, \;\; A_{11}^q u_1^q \geq b_1^q - A_{10}^q u_0$$

and denote the corresponding optimal value functions by

$$\varphi^q(u_0) := \inf\big(\mathcal{P}_1^q(u_0)\big)$$

The function $\widehat{\varphi}$ giving the expected cost of recourse is a weighted sum of these functions φ^q, namely

$$\widehat{\varphi}(u_0) = \sum_{q=1}^{\bar{q}} \pi^q \varphi^q(u_0).$$

By analyzing the φ^q's individually, we can hope to understand better how $\widehat{\varphi}$ might be handled in the projected problem $(\mathcal{P}_0)$.

Recourse assumptions: For this purpose we'll assume that the value $\inf(\mathcal{P}_0) = \inf(\mathcal{P})$ is finite and keep to the case of relatively complete recourse, in which

$$
\begin{aligned}
u_0 \geq 0,\ A_{00}u_0 \geq b_0 \quad &\Longrightarrow \quad \widehat{\varphi}(u_0) < \infty\\
&\Longrightarrow \quad \varphi^q(u_0) < \infty \text{ almost surely}.
\end{aligned}
$$

Here "almost surely" refers to the inequality just holding for the indices q with $\pi^q > 0$, but to avoid the hassling with this minor technicality we'll suppose henceforth in this discussion that actually $\pi^q > 0$ for all q. In terms of the

polyhedral set $C_0 := \{u_0 \,|\, u_0 \geq 0,\ A_{00}u_0 \geq b_0\}$, which is nonempty (or the optimal value in $(\mathcal{P}_0)$ would be ∞), we then have that

$$\varphi^q \text{ is finite on } C_0 \text{ for all } q = 1, \ldots, \bar{q}.$$

Indeed, relatively complete recourse makes the functions φ^q be $< \infty$ on C_0, and then they must also be $> -\infty$ on C_0 or there would exist $u_0 \in C_0$ with $\widehat{\varphi}(u_0) = -\infty$ (so that $\inf(\mathcal{P}_0)$ would be $-\infty$, contrary to assumption).

Dualization in the recourse stage: The recourse subproblem $(\mathcal{P}_1^q(u_0))$ is one of linear programming too. Its dual problem, focused on the vector v_1^q, has the form

$$\big(\mathcal{D}_1^q(u_0)\big) \qquad \text{maximize } \big[b_1^q - A_{10}^q u_0\big] \cdot v_1^q \text{ subject to } v_1^q \geq 0,\ A_{11}^{q*} v_1^q \leq c_1^q.$$

Note that $(\mathcal{D}_1^q(u_0))$ depends on u_0 only through its linear objective in v_1^q, not through its constraints on v_1^q.

Dual formula for recourse costs: The optimal values in $(\mathcal{P}_1^q(u_0))$ and $(\mathcal{D}_1^q(u_0))$ have to agree when either is finite, and under our recourse assumptions the optimal value in $(\mathcal{P}_1^q(u_0))$, namely $\varphi^q(u_0)$, is finite when $u_0 \in C_0$, in particular. Therefore we have

$$\varphi^q(u_0) = \max\big(\mathcal{D}_1^q(u_0)\big) \text{ when } u_0 \in C_0,$$

where "max" is justified since linear programming problems have optimal solutions whenever their optimal values are finite.

Envelope interpretation: This has a valuable interpretation as an "envelope formula," i.e., a formula in which the function φ^q is expressed as the pointwise maximum (or supremum) of a collection of simpler functions. Let

$$\begin{aligned} D_1^q &:= [\text{ feasible set in } (\mathcal{D}_1^q(u_0)) \,] = \{v_1^q \,|\, v_1^q \geq 0,\ A_{11}^{q*} v_1^q \leq c_1^q\}, \\ l^q(u_0, v_1^q) &:= \big[b_1^q - A_{10}^q u_0\big] \cdot v_1^q = \big[b_1^q \cdot v_1^q\big] - \big[A_{10}^{q*} v_1^q\big] \cdot u_0. \end{aligned}$$

Obviously $l^q(u_0, v_1^q)$ is an affine function of u_0 for each choice of v_1^q. Our dual formula for φ^q says that

$$\varphi^q(u_0) = \max_{v_1^q \in D_1^q} l^q(u_0, v_1^q) \text{ when } u_0 \in C_0,$$

and this can be interpreted as saying that, relative to the set C_0 at least, φ^q is the supremum of the collection of affine functions $u_0 \mapsto l^q(u_0, v_1^q)$, viewed as "indexed" by the vectors $v_1^q \in D_1^q$.

Consequence for expected costs: From the dual formula for φ^q and the expression of $\widehat{\varphi}$ as a weighted sum of the φ^q functions, we immediately obtain the formula

$$\begin{aligned}\widehat{\varphi}(u_0) &= \max_{\substack{v_1^q \in D_1^q \\ q=1,\ldots,\bar{q}}} \textstyle\sum_{q=1}^{\bar{q}} \pi^q l^q(u_0, v_1^q) \\ &= \max_{\substack{v_1^q \in D_1^q \\ q=1,\ldots,\bar{q}}} \Big\{ \big[\textstyle\sum_{q=1}^{\bar{q}} \pi^q b_1^q \cdot v_1^q\big] - \big[\textstyle\sum_{q=1}^{\bar{q}} \pi^q A_{10}^{q*} v_1^q\big] \cdot u_0 \Big\} \text{ for } u_0 \in C_0.\end{aligned}$$

Corresponding envelope interpretation: This likewise has can be interpreted in terms of the pointwise maximum of a collection of affine functions of $u_0 \in C_0$:

$$\begin{aligned}\widehat{\varphi}(u_0) &= \max_{\substack{v_1^q \in D_1^q \\ q=1,\ldots,\bar{q}}} l(u_0; v_1^1, \ldots, v_1^{\bar{q}}), \quad \text{where} \\ l(u_0; v_1^1, \ldots, v_1^{\bar{q}}) &:= \big[\textstyle\sum_{q=1}^{\bar{q}} \pi^q b_1^q \cdot v_1^q\big] - \big[\textstyle\sum_{q=1}^{\bar{q}} \pi^q A_{10}^{q*} v_1^q\big] \cdot u_0.\end{aligned}$$

In our general probability notation, this takes the form

$$\widehat{\varphi}(u_0) = \max_{\substack{v_1(\cdot): \\ v_1(\omega) \in D_1(\omega)}} \Big\{ I\!E_\omega\big\{b_1(\omega) \cdot v_1(\omega)\big\} - I\!E_\omega\big\{A_{10}^*(\omega) v_1(\omega)\big\} \cdot u_0 \Big\} \text{ for } u_0 \in C_0,$$

where $D_1(\omega)$ denotes the set of feasible solutions to the problem $(\mathcal{D}_1(u_0, \omega))$ that's dual to $(\mathcal{P}_1(u_0, \omega))$ for fixed u_0 and ω.

Refinement utilizing extreme points: We're getting ever deeper into the special properties of linear programming problems, and the next fact in that line is going to be the to taking advantage of the envelope formula for expected cost. Each of the feasible sets D_1^q, being specified by the linear constraint system $v_1^q \geq 0$, $A_{11}^{q*} v_1^q \leq c_1^q$, is a polyhedral set having a finite number of *extreme points* (also called corner points, or vertices). When a linear function is maximized over such a set, the maximum is attained in particular at some extreme point. Thus, in letting

$$\operatorname{ext} D_1^q := [\text{ set of extreme points of } D_1^q \text{ }],$$

we have a nonempty, finite set of points with the property that, for each $u_0 \in C_0$, problem $(\mathcal{D}_1^q(u_0))$ will have an optimal solution v_1^q that belongs to $\operatorname{ext} D_1^q$. It follows that

$$\varphi^q(u_0) = \max_{v_1^q \in \operatorname{ext} D_1^q} \Big\{ \big[b_1^q \cdot v_1^q\big] - \big[A_{10}^{q*} v_1^q\big] \cdot u_0 \Big\} \text{ when } u_0 \in C_0,$$

so that φ^q has an expression, relative to C_0, as the pointwise maximum of *finitely many* affine functions. Correspondingly for $\widehat{\varphi}$ we have

$$\begin{aligned}\widehat{\varphi}(u_0) &= \textstyle\sum_{q=1}^{\bar{q}} \pi^q \left[\max_{v_1^q \in \operatorname{ext} D_1^q} \left\{ \left[b_1^q \cdot v_1^q \right] - \left[A_{10}^{q*} v_1^q \right] \cdot u_0 \right\} \right] \\ &= \max_{\substack{v_1^q \in \operatorname{ext} D_1^q \\ q=1,\ldots,\bar{q}}} \left\{ \left[\textstyle\sum_{q=1}^{\bar{q}} \pi^q b_1^q \cdot v_1^q \right] - \left[\textstyle\sum_{q=1}^{\bar{q}} \pi^q A_{10}^{q*} v_1^q \right] \cdot u_0 \right\} \text{ when } u_0 \in C_0,\end{aligned}$$

which likewise is a formula of such character.

Piecewise linearity: It can be deduced from these refined formulas that, relative to the polyhedral set C_0, the convex functions φ^q and $\widehat{\varphi}$ are *piecewise linear*. A function is said to have that property if C_0 can be expressed as the union of finitely many polyhedral subsets on each of which the function is given by some affine expression.

Subgradients of convex functions: Functions that might not be differentiable can anyway be "subdifferentiated," and this is especially helpful in dealing with envelope formulas like the ones just derived. Consider any function $\varphi : \mathbb{R}^n \to \overline{\mathbb{R}}$ and a point $\bar{u}$ with $\varphi(\bar{u})$ finite. A vector $z \in \mathbb{R}^n$ is a *subgradient* of φ at $\bar{u}$, written

$$z \in \partial\varphi(\bar{u}),$$

if $(z, -1)$ belongs to the normal cone $N_E(\bar{u}, \varphi(\bar{u}))$ to the epigraph E of φ in $\mathbb{R}^n \times \mathbb{R}$ at the point $(\bar{u}, \varphi(\bar{u}))$. Much could be said about this general definition and its consequences, but we're only going to be involved with *convex* functions φ, having convex epigraphs E. In that case, the condition $(z, -1) \in N_E(\bar{u}, \varphi(\bar{u}))$ corresponds to having $(z, -1) \cdot \left[(u, \alpha) - (\bar{u}, \varphi(\bar{u})) \right] \leq 0$ for all $(u, \alpha) \in E$, so

$$z \in \partial\varphi(\bar{u}) \iff \varphi(u) \geq \varphi(\bar{u}) + z \cdot [u - \bar{u}] \text{ for all } u.$$

(This characterization was in fact the original definition of "subgradient" in convex analysis. Only much later was it generalized to nonconvex functions, as indicated.)

Connection with envelope formulas: Equivalently, according to the inequality test just given, we have for a convex function φ that

$$z \in \partial\varphi(\bar{u}) \iff \left\{ \begin{array}{l} \text{there is an affine function } l, \text{ having } z \\ \text{as its (constant) gradient, such that} \\ \varphi(u) \geq l(u) \text{ for all } u, \text{ and } \varphi(\bar{u}) = l(\bar{u}). \end{array} \right.$$

Thus, whenever φ has a representation as the pointwise maximum of a collection of affine functions l, we can get subgradients of φ by looking at where these affine functions l actually "touch" φ.

Some background properties: For φ convex, the subgradient set $\partial\varphi(\bar{u})$ at a point $\bar{u}$ could be empty, or could consist of a unique z, or could contain more than one vector z. It's always a *closed, convex* set, at least, as can be seen immediately from the definition. It can be proved that $\partial\varphi(\bar{u})$ is a *nonempty, bounded* set if and only if φ is finite on a neighborhood of $\bar{u}$.

If φ is differentiable at $\bar{u}$, then $\partial\varphi(\bar{u})$ is a singleton set whose sole element z is the gradient vector $\nabla\varphi(\bar{u})$. Interestingly, the converse is true as well: if $\partial\varphi(\bar{u})$ is a singleton set, then φ must be differentiable at $\bar{u}$.

Calculus rules: A lot can be said about how to get subgradients of a convex function φ in circumstances when φ is expressed in terms of other, more elementary convex functions for which subgradients are already available. A key case for us will be that of

$$\varphi = \lambda_1\varphi_1 + \cdots + \lambda_r\varphi_r \quad \text{with } \varphi_i \text{ convex, } \lambda_i \geq 0.$$

Then we have a rule coming straight from the inequality test for subgradients:

$$\left.\begin{array}{l} z_i \in \partial\varphi_i(\bar{u}) \\ i = 1, \ldots, r \end{array}\right\} \implies \lambda_1 z_1 + \cdots + \lambda_r z_r \in \partial\varphi(\bar{u}).$$

This can be written neatly as the inclusion

$$\partial(\lambda_1\varphi_1 + \cdots + \lambda_r\varphi_r)(\bar{u}) \supset \lambda_1\partial\varphi_1(\bar{u}) + \cdots + \lambda_r\partial\varphi_r(\bar{u}),$$

where the "sum of sets" on the right refers to the sum of all possible combinations obtained by choosing vectors from those sets. As a matter of fact, "$\supset$" can often be replaced by "$=$" in this relation (as for example in the case where the convex functions φ_i are finite in a neighborhood of $\bar{u}$), but we won't go into that here.

Application of subgradients to expectations: The inclusion just explained can be applied in particular in our linear programming framework:

$$\widehat{\varphi} = \textstyle\sum_{q=1}^{\bar{q}} \pi^q \varphi^q \implies \partial\widehat{\varphi}(u_0) \supset \sum_{q=1}^{\bar{q}} \pi^q \partial\varphi^q(u_0).$$

An analogue in our general probability notation can readily be obtained too for the case of $\widehat{\varphi}(u_0) = I\!E_\omega\left\{\varphi(u_0, \omega)\right\}$, which covers not only weighted combinations of

convex functions but also "integral combinations." In denoting by $\partial\varphi(u_0,\omega)$ the set of subgradients of $\varphi(\cdot,\omega)$ at u_0 (i.e., with "∂" referring to subgradients taken only in the first argument), we get

$$z(\omega)\in\partial\varphi(u_0,\omega)\ \text{ almost surely }\quad\Longrightarrow\quad I\!E_\omega\big\{z(\omega)\big\}\in\partial\widehat{\varphi}(u_0).$$

Cutting plane approach to the projected problem: We're headed toward the most popular method for solving two-stage stochastic linear programming problems, but it will be best to look first at the main idea in the broader case of a two-stage stochastic programming problem that's not necessarily linear. The projected problem then takes the form

$$(\mathcal{P}_0)\qquad\qquad \text{minimize }\ f_0(u_0)+\widehat{\varphi}(u_0)\ \text{ over }\ u_0\in U_0$$

for a convex set U_0 and convex functions f_0 and $\widehat{\varphi}$. We can suppose that U_0 and f_0 are easy enough to handle directly, and that the difficulty just comes from $\widehat{\varphi}$. As earlier, we denote by C_0 the set of points $u_0\in U_0$ satisfying $f_0(u_0)<\infty$. We assume that $\widehat{\varphi}$ is finite on C_0, with the subgradient set $\partial\widehat{\varphi}(u_0)$ nonempty for all $u_0\in C_0$.

Successive envelope approximations: An approach will be described for generating successively better "approximations" of $(\mathcal{P}_0)$ out of subgradients of $\widehat{\varphi}$. Afterward we can see how such subgradients can be computed through the rules above.

The basic picture is that of a sequence of problems, indexed by $\nu=1,2,\ldots$ in which $\widehat{\varphi}$ is replaced by a simpler function $\widehat{\varphi}^\nu$:

$$(\mathcal{P}_0^\nu)\qquad\qquad \text{minimize }\ f_0(u_0)+\widehat{\varphi}^\nu(u_0)\ \text{ over }\ u_0\in U_0.$$

We'll have $\widehat{\varphi}^1\le\cdots\le\widehat{\varphi}^\nu\le\cdots\le\widehat{\varphi}$, so that the optimal values in these problems will form a nondecreasing sequence.

Algorithm statement: At the start, when $\nu=1$, $\widehat{\varphi}^1$ can be taken to be any convex function $\le\widehat{\varphi}$ on C_0, e.g. an affine function (see below). In the general step, where we have problem $(\mathcal{P}_0^\nu)$, we calculate

$$u_0^\nu\in\operatorname{argmin}(\mathcal{P}_0^\nu),\qquad \widehat{\varphi}^\nu(u_0^\nu),\qquad \widehat{\varphi}(u_0^\nu).$$

If $\widehat{\varphi}^\nu(u_0^\nu)=\widehat{\varphi}(u_0^\nu)$, we terminate with the claim that *the current point u_0^ν is an optimal solution $\bar{u}_0$ to $(\mathcal{P}_0)$ itself.* Otherwise $\widehat{\varphi}^\nu(u_0^\nu)<\widehat{\varphi}(u_0^\nu)$, in which case we generate an affine function l^ν by taking

$$l^\nu(u_0):=\widehat{\varphi}(u_0^\nu)+z^\nu\cdot[u_0-u_0^\nu]\ \text{ for some }\ z^\nu\in\partial\widehat{\varphi}(u_0^\nu).$$

Then we define a next approximate function $\widehat{\varphi}^{\nu+1}$ by

$$\widehat{\varphi}^{\nu+1}(u_0) := \max\{\widehat{\varphi}^{\nu}(u_0), l^{\nu}(u_0)\}$$

and proceed to the next iteration, where the problem to be solved is $(\mathcal{P}_0^{\nu+1})$.

Form of the approximate functions: On the basis of this prescription we have in each problem $(\mathcal{P}_0^{\nu})$ that

$$\widehat{\varphi}^{\nu}(u_0) = \max\{\widehat{\varphi}^{1}(u_0), l^{1}(u_0), \ldots, l^{\nu}(u_0)\}.$$

Such an expression can be handled in the minimization through the introduction of an epigraphical variable, for instance. Note that the initial function $\widehat{\varphi}^1$ could in particular be taken to be an affine function l^0 defined like l^ν but with respect to a subgradient z^0 of $\widehat{\varphi}$ at some choice of a starting point u_0^0.

Technical issues: For the algorithm to make sense, we have to know that an optimal solution u_0^ν to $(\mathcal{P}_0^\nu)$ exists. This will be true surely if C_0 is closed and bounded (as well as nonempty), and the functions f_0 and $\widehat{\varphi}^\nu$ are continuous relative to C_0. The continuity of $\widehat{\varphi}^\nu$ follows from its max formula as long as $\widehat{\varphi}^1$ is continuous.

Termination: It was claimed that if, in iteration ν, the current point u_0^ν has $\widehat{\varphi}^\nu(u_0^\nu) = \widehat{\varphi}(u_0^\nu)$, it actually solves $(\mathcal{P}_0)$ itself. Why?

In this situation the objective function $f_0 + \widehat{\varphi}$ in $(\mathcal{P}_0)$ has the same value at u_0^ν as the objective function $f_0 + \widehat{\varphi}^\nu$ in $(\mathcal{P}_0^\nu)$, namely the value $\inf(\mathcal{P}_0^\nu)$. Since u_0^ν can't give a lower value to $f_0 + \widehat{\varphi}$ than $\inf(\mathcal{P}_0)$, while on the other hand $\inf(\mathcal{P}_0^\nu) \leq \inf(\mathcal{P}_0)$, it's legitimate to conclude that u_0^ν gives the value $\inf(\mathcal{P}_0)$ to $f_0 + \widehat{\varphi}$ and thus constitutes an optimal solution to $(\mathcal{P}_0)$.

Convergence in general: If the algorithm doesn't terminate in finitely many iterations in the manner just explained, it generates an infinite sequence of points u_0^ν and values $\alpha^\nu := \inf(\mathcal{P}_0^\nu)$ with $\alpha^1 \leq \cdots \leq \alpha^\nu \leq \cdots \leq \bar{\alpha} := \inf(\mathcal{P}_0)$. It evidently makes improvements of a sort in each iteration, but what guarantee is there that α^ν converges to $\bar{\alpha}$ as $\nu \to \infty$, rather than stagnating at some value lower than $\bar{\alpha}$? Additional assumptions are needed to handle this.

Suppose for instance (along with the closedness and boundedness of C_0 and the continuity of f_0 and $\widehat{\varphi}^1$ on C_0) that the function $\widehat{\varphi}$ is finite not merely on C_0 but on some open set $O \supset C_0$ (as would be true under the complete recourse property, say). *Then (1) $\alpha^\nu \to \bar{\alpha}$, and (2) every cluster point of the sequence $\{u_0^\nu\}_{\nu=1}^\infty$ is an optimal solution $\bar{u}_0$ to $(\mathcal{P}_0)$.* (Note that the sequence $\{u_0^\nu\}_{\nu=1}^\infty$ is bounded by virtue of lying in C_0, so it does have a cluster point.)

Justification: The proof of this exactly parallels the proof of convergence of the cutting plane method for minimizing a smooth convex function over a compact convex set, but with subgradients substituted for gradients. An appeal must be made to "continuity properties" in the behavior of subgradients of $\widehat{\varphi}$, which we're not going into here, and so the details will be omitted.

"L-shaped method" in stochastic linear programming: The cutting plane algorithm for solving the projected problem can be applied in the two-stage linear-programming case we investigated above, and special features then come to the fore. The resulting algorithm is known as the "L-shaped method" for reasons that are now obscure but probably have to do with the pattern of matrices A_{00}, $A_{10}(\omega)$ and $A_{11}(\omega)$ —with no A_{01} matrix.

Linear programming in the subproblems: Let the initial approximation $\widehat{\varphi}^0$ be chosen affine (see above). Then in each iteration the subproblem to be solved is:

$$(\mathcal{P}_0^\nu) \qquad \begin{array}{l} \text{minimize} \quad c_0 \cdot u_0 + \max\{l^0(u_0), l^1(u_0), \ldots, l^{\nu-1}(u_0)\} \\ \text{subject to} \quad u_0 \geq 0, \;\; A_{00}u_0 \geq b_0. \end{array}$$

Through the introduction of an epigraphical variable to rewrite the max term, this can be converted to linear programming in one higher dimension.

Subgradient calculation: A crucial step is that of calculating, for the current vector u_0^ν, a subgradient $z^\nu \in \partial\widehat{\varphi}(u_0)$. In our discrete probability setting, we know this can be reduced to the following:

$$\text{find } z^{q\,\nu} \in \partial\varphi^q(u_0^\nu) \text{ for each } q, \text{ then take } z^\nu := \textstyle\sum_{q=1}^{\bar{q}} \pi^q z^{q\,\nu}.$$

How can we determine a vector $z^{q\,\nu} \in \partial\varphi^q(u_0^\nu)$? Our envelope formula for φ^q comes in here. We can take $z^{q\,\nu}$ to be the gradient of any affine function in this formula whose graph "touches" that of φ^q at u_0^ν. Thus, we merely need to

$$\text{calculate } v_1^{q\,\nu} \in \operatorname{argmax}\big(\mathcal{D}_1^q(u_0^\nu)\big), \text{ then set } z^{q\,\nu} = -A_{10}^{q*} v_1^{q\,\nu}.$$

An important observation is that, in carrying this out, we can always choose $v_1^{q\,\nu}$ to be an optimal solution to $\big(\mathcal{D}_1^q(u_0^\nu)\big)$ that's an extreme point of the feasible set D_1^q in this problem, i.e., $v_1^{q\,\nu} \in \operatorname{ext} D_1^q$.

Getting an extreme point: Some linear programming algorithms, most notably the "simplex method," automatically produce, as an optimal solution, an extreme point of the feasible set.

Finite termination property: Under the provision that the vectors $v_1^{q\,\nu}$ are always taken to be in $\operatorname{ext} D_1^q$, this specialized version of the cutting plane algorithm *always terminates in finitely many iterations with an optimal solution* $\bar{u}_0$ *to* $(\mathcal{P}_0)$.

Justification: Because of the way the subgradients z^ν are generated from vectors $v_1^{q\,\nu} \in \operatorname{ext} D_1^q$, with the sets $\operatorname{ext} D_1^q$ being finite, there are only finitely many possible vectors that are eligible to be a z^ν. If the algorithm were to fail to terminate and instead keep iterating indefinitely, the infinite sequence of vectors z^ν it would produce would have to contain repetitions. Every time we generate z^ν, we derive from it an affine function that has this vector as its gradient and touches the graph of $\widehat{\varphi}$ somewhere; that function is uniquely determined by z^ν, regardless of where we think of it as touching. Thus, repetitions in the sequence of vectors z^ν necessarily induce repetitions in the sequence $l^0, l^1, \ldots, l^{\nu-1}, l^\nu, \ldots$ of affine functions used in approximating $\widehat{\varphi}$. But when l^ν is obtained, it satisfies

$$l^\nu(u_0^\nu) = \widehat{\varphi}(u_0^\nu) > \widehat{\varphi}^\nu(u_0^\nu) = \max\bigl\{l^0(u_0^\nu), l^1(u_0^\nu), \ldots, l^{\nu-1}(u_0^\nu)\bigr\}$$

and therefore can't be one of the functions $l^0, l^1, \ldots, l^{\nu-1}$ encountered previously. In other words, such repetition is impossible.

Comments: The proof that the L-shaped method has finite termination confirms that it is soundly formulated, but doesn't, in itself, show that it's "good." For all we know, the finitely many iterations could take a million years! For *practical* justification, any computational method depends on the experience people have with how it behaves for real problems. In this, the L-shaped method has in fact been very successful, due in no small measure to the clever implementations that have been made of it.

These implementations have been based to some extent on making simplifying assumptions about the randomness. Especially favorable is the case where the $c_1(\omega)$ and $A_{11}(\omega)$ aren't uncertain at all and therefore don't depend on ω. Then, in our discrete probability notation, we can drop the q superscript on these elements and on the feasible set D_1^q in $(\mathcal{D}_1^q(u_0))$, which becomes just

$$D_1 := \bigl\{v_1 \geq 0 \,\big|\, A_{11}^* v_1 \leq c_1\bigr\}.$$

The only difference then between the various problems $(\mathcal{D}_1^q(u_0))$ is in the coefficient vector $b_1^q - A_{10}^q u_0$ in the linear function being minimized over D_1. Optimal solutions can always be obtained in $\operatorname{ext} D_1$.

With or without this simplification, it's clear that in each iteration of the algorithm one is faced with solving batches of small-scale linear programming subproblems which are *identical except for shifts in some coefficients.* In this situation, the advanced technology of the simplex method can be brought to bear. Having solved one of the problems, we can get the solution to another problem in the same batch by making relatively minor "adjustments." The option of solving some of these problems in parallel (on separate processors) is available as well.

Another feature of successful implementations is having a way of deleting some of the older affine functions in the expression for $\widehat{\varphi}^{\nu}$ when they no longer seem needed. This has to be done carefully, however, so that progress toward a solution isn't disrupted.

Benders decomposition: The L-shaped method, and more generally the cutting plane approach to the projected problem $(\mathcal{P}_0)$, are often cited as instances of "Benders decomposition" in stochastic programming. The original idea of Benders was that of projecting a problem in two vector variables into a projected problem in one of them, and then using a cutting plane technique to handle the projected objective function.

Multistage stochastic linear programming: These ideas can be extended to problems with more than just one recourse stage, but of course everything gets more complicated. In a three-stage model, for example, the process has the form

choose $u_0 \geq 0$ with $A_{00}u_0 \geq b_0$, paying $c_0 \cdot u_0$,

observe $\omega_1 \in \Omega_1$,

choose $u_1 \geq 0$ with $A_{10}(\omega_1)u_0 + A_{11}(\omega_1)u_1 \geq b_1(\omega_1)$, paying $c_1(\omega_1) \cdot u_1$,

observe $\omega_2 \in \Omega_2$,

choose $u_2 \geq 0$ with $A_{20}(\omega_1, \omega_2)u_0 + A_{21}(\omega_1, \omega_2)u_1 + A_{22}(\omega_1, \omega_2)u_2 \geq b_2(\omega_1, \omega_2)$, paying $c_2(\omega_1, \omega_2) \cdot u_2$.

The corresponding optimization problem concerns policies $u(\cdot)$ comprised of an initial decision u_0, first recourse $u_1(\omega)$ and second recourse $u_2(\omega_1, \omega_2)$.

Projected costs: Costs in the second recourse stage can be projected back to the first recourse stage and from there back to the initial stage. Once more, linear programming duality can be put to good use in representing these projected costs for purposes of computation.

Nested Benders decomposition: Cutting plane methods in such a multistage setting can be developed in a "nested" manner, based on repeated projection.

4. EXTENDED LINEAR-QUADRATIC MODELS

The virtues of stochastic linear programming are offset to some degree by the shortcomings of linear modeling in optimization. The most serious shortcoming is seen in the insistence on linear objective functions, which are unable to accommodate the penalty expressions that ought to be a vital part of problem formulation. An awkward inconsistency is met also in the fact that the projected problems one gets through consideration of projected costs are no longer linear programming problems. In these respects, the context of linear programming is unpleasantly fragile for stochastic programming.

A painless remedy for these difficulties is available and will now be laid out. We'll see that the classical duality theory of linear programming problems can readily be broadened to cover "extended linear-quadratic" programming problems, which even go beyond quadratic programming problems. Among other things, this generalization will provide powerful tools for handling penalty expressions.

Linear versus quadratic programming in optimization: Linear programming models are very popular because of the high state of development of linear programming methodology. Linear programming problems don't fit into optimization theory in the same way that solving linear equations, say, fits with the theory of solving nonlinear equations, however. For instance, we don't typically approximate an optimization problem in conventional format with smooth objective function f_0 and constraint functions f_i by "linearizing" all these functions locally around a given point. Instead, the appropriate approximation is based on linearizing only the constraint functions and replacing the objective function by a *quadratic* function, moreover one derived not just from f_0 but the Lagrangian function by way of a vector of Lagrange multipliers. This is seen in nonlinear programming techniques like "sequential quadratic programming," which essentially generalize the optimization version of Newton's method to the case of constraints.

Two-stage stochastic quadratic programming: It's natural therefore to work, at least as a start in the right direction, toward an adaptation to stochastic *quadratic programming* of the ideas we've been considering for stochastic *linear* programming. In the two-stage case, it's clear that the process should then be viewed as follows:

$$\begin{aligned}
&\text{choose } u_0 \geq 0 \text{ with } A_{00}u_0 \geq b_0, \text{ paying } c_0 \cdot u_0 + \tfrac{1}{2}u_0 \cdot C_0 u_0, \\
&\text{observe } \omega \in \Omega, \\
&\text{choose } u_1 \geq 0 \text{ with } A_{10}(\omega)u_0 + A_{11}(\omega)u_1 \geq b_1(\omega), \\
&\qquad\qquad \text{paying } c_1(\omega) \cdot u_1 + \tfrac{1}{2}u_1 \cdot C_1(\omega)u_1,
\end{aligned}$$

where the matrices $C_0 \in \mathbb{R}^{n_0 \times n_0}$ and $C_1(\omega) \in \mathbb{R}^{n_1 \times n_1}$ are symmetric and positive semidefinite.

Difficulty: A major obstacle that arises in such a model is the absence of a duality theory for quadratic programming on quite the same level of symmetry and simplicity as the one for linear programming. This is troublesome because dualization lies at the heart of the solution techniques used in stochastic linear programming.

Before progress can be made in stochastic quadratic programming, we have to look closely at this difficulty to see what can be done about it. We'll find that a broader format than just quadratic programming is needed to make dualization more understandable. An important gain will be made at the same time in the constructive use of penalty expressions.

Quadratic programming duality: Let's leave the stochastic programming scene temporarily and put some effort into basic duality theory. Consider the quadratic programming problem

$$(\mathcal{P}_{\rm qp}) \qquad \text{minimize } c\cdot u + \tfrac{1}{2}u\cdot Cu \text{ subject to } u \geq 0,\ Au \geq b$$

for $u \in \mathbb{R}^n$ and a symmetric matrix $C \in \mathbb{R}^{n\times n}$ that's positive semidefinite. In the special case where $C = 0$, $(\mathcal{P}_{\rm qp})$ becomes the canonical linear programming problem

$$(\mathcal{P}_{\rm lp}) \qquad \text{minimize } c\cdot u \text{ subject to } u \geq 0,\ Au \geq b,$$

which dualizes, as we know, to

$$(\mathcal{D}_{\rm lp}) \qquad \text{maximize } b\cdot v \text{ subject to } v \geq 0,\ A^*v \leq c.$$

Naively one might hope that in dualizing $(\mathcal{P}_{\rm qp})$ one would get another quadratic programming problem, involving an auxiliary quadratic form constructed somehow from the data in $(\mathcal{P}_{\rm qp})$, but that's not the case, at least not straightforwardly. Instead, one gets a dual problem in which a penalty expression appears. That, however, sparks interest in itself.

Derivation of the dual problem from the Lagrangian: The Lagrangian function for problem $(\mathcal{P}_{\rm qp})$ has the form

$$L(u,v) = c\cdot u + \tfrac{1}{2}u\cdot Cu + v\cdot[b - Au] \ \text{ on } \mathbb{R}^n_+ \times \mathbb{R}^m_+.$$

Out of the general theory of Lagrangians, as developed in the context of two-person zero-sum games, we know that the given ("primal") problem $(\mathcal{P}_{\rm qp})$ can be identified with

$$\text{minimize } f(u) \text{ subject to } u \geq 0, \text{ where } f(u) := \sup_{v\geq 0} L(u,v),$$

and that the corresponding dual problem comes out then abstractly as

$$\text{maximize } g(v) \text{ subject to } v \geq 0, \text{ where } g(v) := \inf_{u \geq 0} L(u, v).$$

To get further, we have to see what additional information about g can be gleaned from the particular structure we now have for L. We obtain

$$g(v) = v \cdot b + \inf_{u \geq 0} \big\{ c \cdot u + \tfrac{1}{2} u \cdot Cu - v \cdot Au \big\} = v \cdot b - \theta(A^* v - c)$$

for the function θ on $\mathbb{R}^n$ defined by the (envelope!) formula

$$\theta(z) := \sup_{u \geq 0} \big\{ z \cdot u - \tfrac{1}{2} u \cdot Cu \big\}.$$

Clearly $\theta(z)$ is convex as a function of z, because it's the pointwise supremum of a collection of affine functions of z. Whether or not we easily can bring the formula for $\theta(z)$ down to something sharper, the dual problem we obtain in association with $(\mathcal{P}_{\text{qp}})$ can be written in terms of θ as

$$(\mathcal{D}_{\text{qp}}) \qquad \text{maximize } b \cdot v - \theta(A^* v - c) \text{ subject to } v \geq 0.$$

Since θ might in principle take on the value ∞ in some regions, there could be additional constraints on v in $(\mathcal{D}_{\text{qp}})$ that have yet to be made explicit.

Linear programming subcase: As a matter of fact, in the linear programming case, where $C = 0$, problem $(\mathcal{P}_{\text{qp}})$ reduces to $(\mathcal{P}_{\text{lp}})$, and the dual problem $(\mathcal{D}_{\text{qp}})$ reduces accordingly to $(\mathcal{D}_{\text{lp}})$. The θ function is then just the indicator of the nonpositive orthant $\mathbb{R}^n_-$, so that $\theta(A^* v - c) = 0$ if $A^* v - c \leq 0$ but $\theta(A^* v - c) = \infty$ if $A^* v - c \not\leq 0$. In other words, the θ term then represents the constraint $A^* v \leq c$.

A quadratic example in more detail: Let's inspect the case where the matrix C is diagonal, say $C = \operatorname{diag}[\gamma_1, \ldots, \gamma_n]$ with $\gamma_j \geq 0$. If actually $\gamma_j = 0$ for all j, we have the linear programming case just described, where θ is the indicator of $\mathbb{R}^n_-$. This will serve as a reference point. Our aim is to understand what we get more generally when the γ_j's aren't all 0, say

$$\begin{cases} \gamma_j > 0 & \text{for } j = 1, \ldots, r, \\ \gamma_j = 0 & \text{for } j = r+1, \ldots, n. \end{cases}$$

The calculation of $\theta(z)$ can proceed then in terms of the components of $z = (z_1, \ldots, z_n)$ and $u = (u_1, \ldots, u_n)$ with

$$\theta(z_1, \ldots, z_n) = \sup_{\substack{u_j \geq 0 \\ j=1,\ldots,n}} \Big\{ \sum\nolimits_{j=1}^n \big[z_j u_j - \frac{\gamma_j}{2} u_j^2 \big] \Big\} = \sum\nolimits_{j=1}^n \theta_j(z_j)$$

for the functions

$$\theta_j(z_j) := \sup_{u_j \geq 0} \Big\{ z_j u_j - \frac{\gamma_j}{2} u_j^2 \Big\},$$

which come out in the form:

$$\begin{cases} \text{for } j = 1, \ldots, r: & \theta_j(z_j) = \begin{cases} 0 & \text{when } z_j \leq 0, \\ \frac{1}{2\gamma_j} z_j^2 & \text{when } z_j > 0, \end{cases} \\ \text{for } j = r+1, \ldots, n: & \theta_j(z_j) = \begin{cases} 0 & \text{when } z_j \leq 0, \\ \infty & \text{when } z_j > 0. \end{cases} \end{cases}$$

Therefore, in terms of having

$$z_j = \sum\nolimits_{i=1}^{m} v_i a_{ij} - c_j \text{ for } j = 1, \ldots, n \text{ when } z = A^* v - c,$$

the effect of γ_j being positive is to replace the constraint $\sum_{i=1}^m v_i a_{ij} \leq c_j$ by a penalty term that vanishes when the constraint is satisfied but grows quadratically (at a rate governed by $1/\gamma_j$) when the constraint is violated.

Penalty interpretation more broadly: Even when C isn't a diagonal matrix, the function θ appearing in $(\mathcal{D}_{\text{qp}})$ can be seen to vanish on $\mathbb{R}^n_+$ but to be positive everywhere else (with ∞ as a possibility). The expression $\theta(A^*v - c)$ acts therefore as a penalty substitute for the constraint $A^*v \leq c$ that appears when $C = 0$.

Quadratic programming with an abstract constraint: Quadratic programming, like linear programming, can be formulated in terms of a general linear system of constraints, not necessarily in the canonical form given in $(\mathcal{P}_{\text{qp}})$. In particular, such a system can involve an abstract constraint $u \in U$ with U polyhedral. A prime example would be a box U, bounded or unbounded, but there are other useful candidates as well. What do we get then?

In replacing the condition $u \geq 0$ in $(\mathcal{P}_{\text{qp}})$ by $u \in U$, the form of the dual problem $(\mathcal{D}_{\text{qp}})$ comes out the same, except for a shift in the formula for the θ function: the maximization over $u \geq 0$ in obtaining $\theta(z)$ simply turns into maximization over $u \in U$. There are lots of possibilities to explore in trying to understand the expressions one could get in this way for $\theta(z)$. The main thing, however, is that we are naturally led to consider functions θ that depend on two objects: a polyhedral set U and a positive semidefinite matrix C.

Piecewise linear-quadratic penalty expressions: It's worth formalizing this notion. For any choice of a nonempty polyhedral set $U \subset \mathbb{R}^n$ and a symmetric, positive semidefinite matrix $C \in \mathbb{R}^{n \times n}$, we denote by θ_{UC} the function on $\mathbb{R}^n$ defined by

$$\theta_{UC}(z) := \sup_{u \in U} \big\{ z \cdot u - \tfrac{1}{2} u \cdot Cu \big\}.$$

This notation carries over to other circumstances. For instance, we'll soon have use also for the function θ_{VB} on $\mathbb{R}^m$ defined by

$$\theta_{VB}(w) := \sup_{v \in V} \{ w \cdot v - \tfrac{1}{2} v \cdot Bv \}$$

for any choice of a nonempty polyhedral set $V \subset \mathbb{R}^m$ and a symmetric, positive semidefinite matrix $B \in \mathbb{R}^{m \times m}$. Note that these are envelope formulas. We set aside for now the question of how θ_{UC} looks for various U and C, but some general facts can be stated (without proof, since the machinery is beyond us).

Penalty aspects: As long as $0 \in U$, the function θ_{UC} will be nonnegative everywhere, with $\theta_{UC}(0) = 0$. There can be other points besides 0 where θ_{UC} vanishes; the set of all such points is inevitably a certain *polyhedral cone* K_{UC}. The expression $\theta_{UC}(A^*v - c)$ can thus be viewed as a penalty substitute for the condition $A^*v - c \in K_{UC}$, which moreover (through the polyhedral property of K_{UC}) can be viewed as standing for a "linear" constraint system on v.

If $0 \notin U$, θ_{UC} isn't a penalty function in that mode, but may serve a useful purpose anyway. Negative values of θ_{UC} can be still be called penalties if we wish (with negative penalties as rewards).

Linear-quadratic aspects: Another property of importance is that θ_{UC} is always a *convex* function that's *piecewise linear-quadratic*. The latter means by definition that the effective domain $\{ z \,|\, \theta_{UC}(z) < \infty \}$ of θ_{UC} is a polyhedral set representable as a union of finitely many polyhedral subsets on which $\theta_{UC}(z)$ is given by a polynomial function of degree at most 2 in the components z_j of z. (As a special case there might be no "pieces": θ_{UC} could be given by a single such formula over its entire effective domain, which could even be all of $\mathbb{R}^n$.)

Advantages of the envelope formulas: One can contemplate many functions that, in some context or another, might be appropriate as a penalty function θ having a formula in pieces. Descriptions could be quite complicated, however, if one is obliged to write down just which all the pieces are and how they articulate with respect to each other. A powerful advantage of the functions θ_{UC} is that they can be specified very simply, without any need to list the pieces. It's only necessary to specify a matrix C and a polyhedral set U, which is particularly easy when U is a box and at worst requires writing down a system of linear constraints.

Experience counts: With experience one soon gets used to working with θ functions in this class and knowing what U and C to invoke for a given purpose.

Extended linear-quadratic programming: If piecewise linear-quadratic penalty functions have a natural role in dualizing quadratic programming problems, and are attractive as tools in optimization models more generally, why not incorporate them into the problem format from the start? By a problem of *extended linear-quadratic programming* (in "primal canonical format"), we'll mean an optimization problem of the type

$$(\mathcal{P}_{\text{elqp}}) \qquad \text{minimize} \;\; c\cdot u + \tfrac{1}{2}u\cdot Cu + \theta_{VB}(b - Au) \;\; \text{over} \;\; u \in U,$$

where the sets $U \subset \mathbb{R}^n$ and $V \subset \mathbb{R}^m$ are nonempty and polyhedral, and the matrices $C \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times m}$ are symmetric and positive semidefinite.

Feasible set: The set of points $u \in U$ such that $\theta_{VB}(Au - b) < \infty$ is *polyhedral* by virtue of the fact that U and the effective domain $\{w \,|\, \theta_{VB}(w) < \infty\}$ of θ_{VB} are polyhedral. Thus, problem $(\mathcal{P}_{\text{elqp}})$ is "linearly constrained" in principle.

Objective properties: The function being minimized over the feasible set is convex and piecewise linear-quadratic.

Quadratic programming and linear programming cases: Problem $(\mathcal{P}_{\text{elqp}})$ reduces to $(\mathcal{P}_{\text{qp}})$ when $U = \mathbb{R}^n_+$, $V = \mathbb{R}^m_+$, and B is the zero matrix; then $\theta_{VB}(b - Au) = 0$ if $b - Au \leq 0$ but $\theta_{VB}(b - Au) = \infty$ if $b - Au \not\leq 0$. When $C = 0$ in addition, $(\mathcal{P}_{\text{elqp}})$ reduces of course to $(\mathcal{P}_{\text{lp}})$.

Lagrangian function: Although $(\mathcal{P}_{\text{elqp}})$ doesn't have constraints in the conventional format, and therefore doesn't fit into the framework where Lagrange multipliers have traditionally been introduced, it makes sense to speak of it anyway as having, as its *Lagrangian*, the function

$$L(u, v) := c\cdot u + \tfrac{1}{2}u\cdot Cu + b\cdot v - \tfrac{1}{2}v\cdot Bv - v\cdot Au \quad \text{on} \quad U \times V.$$

The reason is that $(\mathcal{P}_{\text{elqp}})$ consists of minimizing over U the function $f(u) = \sup_{v\in V} L(u, v)$. Indeed, this supremum comes out as $c\cdot u + \frac{1}{2}u\cdot Cu + \theta_{VB}(b - Au)$ by the definition of θ_{VB}.

Dual problem: This leads us to introduce, as dual to $(\mathcal{P}_{\text{elqp}})$, the problem of maximizing over V the function $g(v) = \inf_{u\in U} L(u, v)$. It comes out as

$$(\mathcal{D}_{\text{elqp}}) \qquad \text{maximize} \;\; b\cdot v - \tfrac{1}{2}v\cdot Bv - \theta_{UC}(A^*v - c) \;\; \text{over} \;\; v \in V,$$

and thus is again a problem of extended linear-quadratic programming (but now in "dual canonical format").

Note: In the special cases corresponding to quadratic programming and linear programming, this turns into the dual problems previously considered.

General test for ELQP: Whether an optimization problem is a problem of extended linear-quadratic programming doesn't, of course, depend on the notation in which it's written. The real criterion that emerges is this: it should be the primal (or dual) problem associated with a Lagrangian triple consisting of a convex-concave linear-quadratic function on a product of two polyhedral sets.

Duality facts: The fundamental theorem about duality in extended linear-quadratic programming exactly follows the lines of the familiar one in linear programming:

> *The optimal values in* $(\mathcal{P}_{\text{elqp}})$ *and* $(\mathcal{D}_{\text{elqp}})$ *are equal unless* $\inf(\mathcal{P}_{\text{elqp}}) = \infty$ *and* $\sup(\mathcal{D}_{\text{elqp}}) = -\infty$*, i.e., neither of the problems has a feasible solution. Moreover, as long as the common optimal value in these problems is finite, optimal solutions* $\bar{u}$ *and* $\bar{v}$ *exist for* $(\mathcal{P}_{\text{elqp}})$ *and* $(\mathcal{D}_{\text{elqp}})$*.*

Observation: These facts apply in particular to the quadratic programming problem $(\mathcal{P}_{\text{qp}})$ and its dual $(\mathcal{D}_{\text{qp}})$, even though the latter typically isn't another quadratic programming problem. Without having passed to the broader context of extended linear-quadratic programming, we wouldn't have achieved a clear and satisfying view of this duality.

Proof background: The stated facts can be derived from the theory of quadratic programming by a technique of rewriting any problem of extended linear-quadratic programming as one of conventional quadratic programming in extra variables (cf. below). Quadratic programming problems, as convex programming problems with linear constraints, don't need a constraint qualification in order to have their optimal solutions characterized in terms of saddle points of the Lagrangian. By working with such saddle points, one can connect up with the game-theoretic framework from which $(\mathcal{P}_{\text{qp}})$ and $(\mathcal{D}_{\text{qp}})$ have been derived. A property that needs to be established separately, as a stepping stone, is the existence of an optimal solution to any quadratic programming problem with finite optimal value.

Saddle point characterization of optimality: Because $(\mathcal{P}_{\text{qp}})$ and $(\mathcal{D}_{\text{qp}})$ correspond to the "game" for L, U, V, the stated duality facts mean the following: $\bar{u}$ and $\bar{v}$ are optimal solutions to $(\mathcal{P}_{\text{qp}})$ and $(\mathcal{D}_{\text{qp}})$ if and only if the pair $(\bar{u}, \bar{v})$ is a saddle point of $L(u, v)$ over $U \times V$, i.e.,

$$\bar{u} \in \operatorname*{argmin}_{u \in U} L(u, \bar{v}), \qquad \bar{v} \in \operatorname*{argmax}_{v \in V} L(\bar{u}, v).$$

Generalized Kuhn-Tucker conditions: The saddle point condition just provided translates, through the convexity of $L(u,v)$ in u and its concavity in v, to an alternative characterization of optimality in terms of normal vectors: $\bar{u}$ and $\bar{v}$ are optimal solutions to $(\mathcal{P}_{\rm qp})$ and $(\mathcal{D}_{\rm qp})$, respectively, if and only if

$$-\nabla_u L(\bar{u},\bar{v}) \in N_U(\bar{u}), \qquad \nabla_v L(\bar{u},\bar{v}) \in N_V(\bar{v}).$$

Here of course $\nabla_u L(\bar{u},\bar{v}) = c + C\bar{u} - A^*\bar{v}$, whereas $\nabla_v L(\bar{u},\bar{v}) = b - B\bar{v} - A\bar{u}$.

Reduction to quadratic programming for computational purposes: The people who write optimization software don't yet think about extended linear-quadratic programming, so one can't invoke existing software directly. But there's a simple trick that can be used when necessary. It's based on the fact that

$$\theta_{VB}(w) \;=\; \inf_{\substack{z\geq 0,\ y\\ Dy+Ez=w}} \left\{ \tfrac{1}{2} y\cdot Py + e\cdot z \right\}$$

for any representation of the (nonempty, polyhedral) set V as $\left\{ v \;\middle|\; E^*v \leq e \right\}$ for some vector $e \in I\!R^r$ and matrix $E \in I\!R^{m\times r}$, and any representation of the (symmetric, positive semidefinite) matrix B as $DP^{-1}D^*$ for some matrix $D \in I\!R^{m\times s}$ and positive *definite*, symmetric matrix $P \in I\!R^{s\times s}$. (Such representations, by no means unique, are readily available always, and P can even be taken to be an identity matrix, although that might not be the most expedient in a given situation.)

This is established by looking at the minimization problem on the right side as a quadratic programming problem in (y,z) and applying to it the dualization already described. The equation follows then as a special case of the cited duality facts.

From this expression, one can see that, for the sake of utilizing quadratic programming software, it's always possible to re-express a problem $(\mathcal{P}_{\rm elqp})$ in primal canonical format as one of ordinary quadratic programming,

$$\begin{aligned} &\text{minimize } \; c\cdot u + \tfrac{1}{2} u\cdot Cu + \tfrac{1}{2} y\cdot Py + e\cdot z \\ &\text{over all } \; (u,y,z) \in U \times I\!R^s \times I\!R^r_+ \; \text{ satisfying } \; Au + Dy + Ez = b. \end{aligned}$$

Recovering the multiplier: The Lagrange multiplier vector for the constraint equation $Au+Dy+Ez=b$, as generated in such a software application, will be *an optimal solution $\bar{v}$ to the dual problem* $(\mathcal{D}_{\rm elqp})$.

This can be seen by deriving the dual to the quadratic programming problem just described and verifying that it turns out to be $(\mathcal{D}_{\rm elqp})$. We know from convex programming in general that the Lagrange multiplier vector obtained from the optimality conditions solves the dual problem associated with the Lagrangian.

Two-stage stochastic ELQP: We're now in position to generalize stochastic linear programming to stochastic *extended linear-quadratic* programming. The two-stage model comes out as follows:

choose $u_0 \in U_0$, paying

$$c_0 \cdot u_0 + \tfrac{1}{2} u_0 \cdot C_0 u_0 + \theta_{V_0 B_0}\big(b_0 - A_{00} u_0\big),$$

observe $\omega \in \Omega$,

choose $u_1 \in U_1(\omega)$, paying

$$c_1(\omega) \cdot u_1 + \tfrac{1}{2} u_1 \cdot C_1(\omega) u_1 + \theta_{V_1(\omega) B_1(\omega)}\big(b_1(\omega) - A_{10}(\omega) u_0 - A_{11}(\omega) u_1\big).$$

Here the matrices C_0, $C_1(\omega)$, B_0 and $B_1(\omega)$ are symmetric and positive semidefinite, and the sets U_0, $U_1(\omega)$, V_0 and $V_1(\omega)$ are polyhedral (and nonempty).

Recourse problem and projected problem: The corresponding projected problem in the initial stage is

$$(\mathcal{P}_0) \qquad \begin{aligned} &\text{minimize over } u_0 \in U_0 \text{ the expression:} \\ &c_0 \cdot u_0 + \tfrac{1}{2} u_0 \cdot C_0 u_0 + \theta_{V_0 B_0}\big(b_0 - A_{00} u_0\big) + \widehat{\varphi}(u_0), \end{aligned}$$

where $\widehat{\varphi}(u_0) := I\!E_\omega\big\{\varphi(u_0, \omega)\big\}$ for the optimal value $\varphi(u_0, \omega)$ in the recourse problem

$$(\mathcal{P}_1(u_0, \omega)) \qquad \begin{aligned} &\text{minimize over } u_1 \in U_1(\omega) \text{ the expression:} \\ &c_1(\omega) \cdot u_1 + \tfrac{1}{2} u_1 \cdot C_1(\omega) u_1 \\ &\qquad + \theta_{V_1(\omega) B_1(\omega)}\big(b_1(\omega) - A_{10}(\omega) u_0 - A_{11}(\omega) u_1\big). \end{aligned}$$

We want to understand how the projected cost function $\widehat{\varphi}$ can be handled in these circumstances, different from the ones in the LP model.

Relatively complete recourse: It will be assumed that for each $u_0 \in U_0$ satisfying $\theta_{V_0 B_0}\big(b_0 - A_{00} u_0\big) < \infty$ there exists, almost surely with respect to $\omega \in \Omega$, some $u_1 \in U_1(\omega)$ satisfying $\theta_{V_1(\omega) B_1(\omega)}\big(b_1(\omega) - A_{10}(\omega) u_0 - A_{11}(\omega) u_1\big) < \infty$. This means, as we know, that the condition $\widehat{\varphi}(u_0) < \infty$ imposes no constraint in $(\mathcal{P}_0)$ beyond what was already present in the initial-stage data.

Dualization: In our pattern of taking $\Omega = \big\{\omega^q \,\big|\, q = 1, \ldots, \bar{q}\big\}$ and denoting by π^q the probability of ω^q, we can write $\varphi^q(u_0)$ in place of $\varphi(u_0, \omega)$ and get $\widehat{\varphi} = \sum_{q=1}^{\bar{q}} \pi^q \varphi^q$, where

$$\varphi^q(u_0) = \inf_{u_1^q \in U_1^q} \Big\{ c_1^q \cdot u_1^q + \tfrac{1}{2} u_1^q \cdot C_1^q u_1^q + \theta_{V_1^q B_1^q}\big(b_1^q - A_{10}^q u_0 - A_{11}^q u_1^q\big) \Big\}.$$

The important thing is that the recourse problem for which $\varphi^q(u_0)$ is the optimal value is an ELQP problem and therefore can immediately be dualized. As long as $\varphi^q(u_0) < \infty$, which we are assuming to be true when u_0 satisfies the initial-stage constraints, we have

$$\varphi^q(u_0) = \sup_{v_1^q \in V_1^q} \Big\{ [b_1^q - A_{10}^q u_0] \cdot v_1^q - \tfrac{1}{2} v_1^q \cdot B_1^q v_1^q - \theta_{U_1^q C_1^q}\big(A_{11}^{q*} v_1^q - c_1^q\big) \Big\}.$$

Moreover, the maximization involves u_0 only in the coefficient of the linear term in the objective function.

Parametric properties: The expression $\varphi^q(u_0)$ is convex and *piecewise linear-quadratic* as a function of u_0. In consequence, $\widehat{\varphi}$ too is convex and piecewise linear-quadratic.

Argument: This follows from our earlier observation about how the common optimal value in a pair of problems $(\mathcal{P}_{\rm elqp})$ and $(\mathcal{D}_{\rm elqp})$ behaves as a function $\rho(b)$ of the b vector in those problems. We saw that it was a convex, piecewise linear-quadratic function. The meaning for the present context is that the optimal value

$$\rho^q(b) = \sup_{v_1^q \in V_1^q} \Big\{ b \cdot v_1^q - \tfrac{1}{2} v_1^q \cdot B_1^q v_1^q - \theta_{U_1^q C_1^q}\big(A_{11}^{q*} v_1^q - c_1^q\big) \Big\}$$

is convex and piecewise linear-quadratic as a function of the coefficient vector b. Here we are putting $b_1^q - A_{10}^q u_0$ in place of b to get $\varphi^q(u_0) = \rho^q(b_1^q - A_{10}^q u_0)$. Since the mapping $u_0 \mapsto b_1^q - A_{10}^q u_0$ is affine, we conclude that φ^q likewise is convex and piecewise linear-quadratic.

It's elementary from the definition of "piecewise linear-quadratic" that when a weighted sum of such functions is taken, as in passing from the φ^q's to $\widehat{\varphi}$, the result is another piecewise linear-quadratic function.

Cutting plane approach to finding a solution: Just as in two-stage stochastic linear programming, we can rely on the discretization and dualization process to build up approximations $\widehat{\varphi}^\nu$ of $\widehat{\varphi}$ in which $\widehat{\varphi}^\nu$ is the pointwise max of a finite collection of affine functions generated from subgradients of $\widehat{\varphi}$. That way, a sequence of solutions u_0^ν to approximate problems $(\mathcal{P}_0^\nu)$ can be obtained, with the hope of getting convergence to an optimal solution $\bar{u}_0$ to $(\mathcal{P}_0)$.

Subgradient generation: For this purpose one simply needs to recognize from the dual formula for $\varphi^q(u_0)$, writing it as

$$\varphi^q(u_0) = \sup_{v_1^q \in V_1^q} \Big\{ [-A_{10}^{q*} v_1^q] \cdot u_0 + b_1^q \cdot v_1^q - \tfrac{1}{2} v_1^q \cdot B_1^q v_1^q - \theta_{U_1^q C_1^q}\big(A_{11}^{q*} v_1^q - c_1^q\big) \Big\},$$

that a subgradient of φ^q at u_0 can be calculated by solving the ELQP subproblem in this maximization. Specifically,

$$v_1^q \in \operatorname*{argmax}_{v_1^q \in V_1^q} \Big\{ [-A_{10}^{q*} v_1^q] \cdot u_0 + b_1^q \cdot v_1^q - \tfrac{1}{2} v_1^q \cdot B_1^q v_1^q - \theta_{U_1^q C_1^q}\big(A_{11}^{q*} v_1^q - c_1^q\big) \Big\}$$
$$\Longrightarrow \quad -A_{10}^{q*} v_1^q \in \partial \varphi^q(u_0).$$

In calculating such a vector v_1^q for each q, we generate the subgradient

$$-\sum\nolimits_{q=1}^{\bar{q}} \pi^q A_{10}^{q*} v_1^q \;\in\; \partial \widehat{\varphi}(u_0).$$

Differences with the linear-programming case: In contrast to what we had in stochastic linear programming, there is no guarantee that such a system of approximation will produce an optimal solution $\bar{u}_0$ in finitely many steps. The reason is that in solving the subproblems to get the vectors v_1^q *we can't count on them being extreme points of the set of feasible solutions.* The feasible set is indeed polyhedral still, but because the maximization isn't just of a linear function, it might not be attained at any extreme point and indeed could be attained even in the interior. Moreover, the location of that point could be influenced by u_0. Thus, there aren't just finitely many candidates for the optimal v_1^q, independently of u_0.

Still, it's possible to develop convergence results in which the infinite sequence $\{u_0^q\}_{\nu=1}^{\infty}$ sequence is bounded and all of its cluster points are optimal solutions to $(\mathcal{P}_0)$. Actually, though, there's a much better approach to solving $(\mathcal{P}_0)$, which takes advantage of the piecewise linear-quadratic structure rather than perceiving it as an obstacle.

Extended envelope generation method: The solution technique to be described next resembles the cutting plane approach in many respects, but it develops approximations $\widehat{\varphi}^\nu$ of $\widehat{\varphi}$ that aren't just of the piecewise linear type expressed by the maximum of a finite collection of affine functions, but instead are *piecewise linear-quadratic.* Such approximations obviously can follow $\widehat{\varphi}$ more closely and thus act more robustly in producing an approximate solution to $(\mathcal{P}_0)$.

Simplified dual format: We continue in the context of discretization and dualization of the stochastic ELQP format, but introduce now a convenient special form for the dual subproblems, arguing that this is possible without any loss of generality. So far, we have observed that

$$\varphi^q(u_0) = \sup_{v_1^q \in V_1^q} \Big\{ [b_1^q - A_{10}^q u_0] \cdot v_1^q - \tfrac{1}{2} v_1^q \cdot B_1^q v_1^q - \theta_{U_1^q C_1^q}\big(A_{11}^{q*} v_1^q - c_1^q\big) \Big\}.$$

We know, on the other hand, that the θ term in this maximization can be re-expressed in the form

$$\theta_{U_1^q C_1^q}(A_{11}^{q*} v_1^q - c_1^q) = \inf_{\substack{z^q \geq 0,\ y^q \\ D^q y^q + E^q z^q = A_{11}^{q*} v_1^q - c_1^q}} \Big\{ \tfrac{1}{2} y^q \cdot P^q y^q + e^q \cdot z^q \Big\}.$$

In substituting this into the maximization, we get

$$\varphi^q(u_0) = \sup_{\substack{(v_1^q, y^q, z^q) \in V_1^q \times \boldsymbol{R}^{s^q} \times \boldsymbol{R}_+^{r^q} \\ A_{11}^{q*} v_1^q - D^q y^q - E^q z^q = c_1^q}} \Big\{ [b_1^q - A_{10}^q u_0] \cdot v_1^q - \tfrac{1}{2} v_1^q \cdot B_1^q v_1^q - \tfrac{1}{2} y^q \cdot P^q y^q - e^q \cdot z^q \Big\}$$

By introducing $\tilde{v}^q := (v_1^q, y^q, z^q)$, $\tilde{V}^q := V_1^q \times I\!R^{s^q} \times I\!R_+^{r^q}$, $\tilde{B}^q = \mathrm{diag}[B^q, P^q, 0]$, and setting up the vector $\tilde{b}^q$ and matrix $\tilde{A}^q$ so that $[\tilde{b}^q - \tilde{A}^q u_0] = [b_1^q - A_{10}^q u_0, 0, -e^q]$, we obtain

$$\varphi^q(u_0) = \sup_{\tilde{v}^q \in \tilde{V}^q} \Big\{ [\tilde{b}^q - \tilde{A}^q u_0] \cdot \tilde{v}^q - \tfrac{1}{2} \tilde{v}^q \cdot \tilde{B}^q \tilde{v}^q \Big\} = \theta_{\tilde{V}^q \tilde{B}^q}(\tilde{b}^q - \tilde{A}^q u_0).$$

This is the formula we'll stick to in what follows. It gives us

$$\widehat{\varphi}(u_0) = \sum\nolimits_{q=1}^{\bar{q}} \pi^q \theta_{\tilde{V}^q \tilde{B}^q}(\tilde{b}^q - \tilde{A}^q u_0)$$

and provides an insightful and notationally more convenient way of getting at the approximations of $\widehat{\varphi}$ that we'll make use of.

Reinterpretation of the projected problem: According to this expression we actually have the formula

$$\widehat{\varphi}(u_0) = \theta_{\tilde{V}\tilde{B}}(\tilde{b} - \tilde{A} u_0)$$

for a certain choice of a set $\tilde{V}$, matrices $\tilde{B}$ and $\tilde{A}$, and vector $\tilde{b}$. Indeed we introduce $\tilde{v}$ as the supervector $(\tilde{v}^1, \ldots, \tilde{v}^{\bar{q}})$ and then take $\tilde{V} = \tilde{V}^1 \times \cdots \times \tilde{V}^{\bar{q}}$, $\tilde{B} = \mathrm{diag}[\pi^1 \tilde{B}^1, \ldots, \pi^{\bar{q}} \tilde{B}^{\bar{q}}]$, $\tilde{A} = [\tilde{A}^1, \ldots, \tilde{A}^{\bar{q}}]$, and $\tilde{b} = [\pi^1 \tilde{b}^1, \ldots, \pi^{\bar{q}} \tilde{b}^{\bar{q}}]$. In this manner, $(\mathcal{P}_0)$ can be identified with the problem

$$(\tilde{\mathcal{P}}_0) \qquad \begin{array}{l} \text{minimize over } u_0 \in U_0 \text{ the expression} \\ c \cdot u_0 + \tfrac{1}{2} u_0 \cdot C u_0 + \theta_{V_0 B_0}(b_0 - A_{00} u_0) + \theta_{\tilde{V}\tilde{B}}(\tilde{b} - \tilde{A} u_0). \end{array}$$

This problem is of course one of extended linear-quadratic programming, in which we could combine the two θ terms to a θ_{VB} by forming $V_0 \times \tilde{V}$ and $\mathrm{diag}[B_0, \tilde{B}]$.

Approximation scheme: The basic idea for approximating $(\mathcal{P}_0)$, construed as $(\tilde{\mathcal{P}}_0)$, is to replace the polyhedral set $\tilde{V}$ by a smaller set $\tilde{V}^\nu$. The corresponding term $\widehat{\varphi}^\nu(u_0) := \theta_{\tilde{V}^\nu \tilde{B}}(\tilde{b} - \tilde{A}u_0)$ then provides a substitute for $\widehat{\varphi}(u_0)$, moreover with $\widehat{\varphi}^\nu(u_0) \leq \widehat{\varphi}(u_0)$. In solving the associated problem $(\tilde{\mathcal{P}}_0^\nu)$ we'll get a point u_0^ν, and it's through a sequence of such points that we aim to come closer and closer to an optimal initial stage decision $\bar{u}_0$. A key factor, though, is that in substituting $\tilde{V}^\nu$ for $\tilde{V}$ we'll also pass to a parameterization of the dual elements in terms of special coefficients.

Envelope generation algorithm: Keeping matters simple, let's suppose that by the start of iteration ν we have generated a sequence of vectors $\tilde{v}^0, \tilde{v}^1, \ldots, \tilde{v}^\nu$ in $\tilde{V}$. The subset $\tilde{V}^\nu$ of $\tilde{V}$ that we assign to this list consists of all the vectors $\tilde{v}$ representable in the form

$$\tilde{v} = \lambda_0 \tilde{v}^0 + \lambda_1 \tilde{v}^1 + \cdots + \lambda_\nu \tilde{v}^\nu \quad \text{with } \lambda_i \geq 0 \text{ and } \lambda_0 + \lambda_1 + \cdots + \lambda_\nu = 1.$$

The way the variable coefficients λ_i can be utilized directly will be explained under "reparameterization," but the updating plan for u_0^ν and $\tilde{V}^\nu$ is as follows. We solve the approximate problem $(\tilde{\mathcal{P}}_0^\nu)$ involving $\theta_{\tilde{V}^\nu, \tilde{B}}$ to get $u_0^{\nu+1}$, and we next solve the subproblem giving the value $\theta_{\tilde{V}, \tilde{B}}(\tilde{b} - \tilde{A}u_0^{\nu+1})$, namely

$$\text{maximize } [\tilde{b} - \tilde{A}u_0^{\nu+1}] \cdot \tilde{v} - \tfrac{1}{2}\tilde{v} \cdot \tilde{B}\tilde{v} \text{ over } \tilde{v} \in \tilde{V},$$

in order to obtain $\tilde{v}^{\nu+1}$. Then by adding $\tilde{v}^{\nu+1}$ to the list $\tilde{v}^0, \tilde{v}^1, \ldots, \tilde{v}^\nu$ we get the set $\tilde{V}^{\nu+1}$ (in which an additional variable $\lambda_{\nu+1}$ enters the representation as the coefficient for $\tilde{v}^{\nu+1}$). This set, lying between $\tilde{V}^\nu$ and $\tilde{V}$, yields an approximation $\widehat{\varphi}^{\nu+1}$ to $\widehat{\varphi}$ with $\widehat{\varphi}^\nu \leq \widehat{\varphi}^{\nu+1} \leq \widehat{\varphi}$. Moreover the new function $\widehat{\varphi}^{\nu+1}$ will agree with $\widehat{\varphi}$ at the point $u_0^{\nu+1}$ in particular. Thus, when this procedure is followed from the start, we will have on reaching iteration $\nu + 1$ that $\widehat{\varphi}^{\nu+1}$ also agrees with $\widehat{\varphi}$ also at all the previous points u_0^0 (initialization) and $u_0^1, \ldots, u_0^\nu$.

Decomposition of the maximization subproblems: The indicated maximization problem with respect to $\tilde{v} \in \tilde{V}$, while seemingly of the same high dimension as $\tilde{v}$, is actually easy. This is because it decomposes immediately into a separate subproblem in each component $\tilde{v}^q$ of $\tilde{v}$ according to the product form of $\tilde{V}$ and the diagonal form of $\tilde{B}$.

Reparameterization of the minimization subproblems: The essential step of solving the approximate problem

$$(\tilde{\mathcal{P}}_0^\nu) \qquad \begin{array}{l} \text{minimize over } u_0 \in U_0 \text{ the expression} \\ c_0 \cdot u_0 + \tfrac{1}{2} u_0 \cdot C u_0 + \theta_{V_0 B_0}(b_0 - A_{00} u_0) + \theta_{\tilde{V}^\nu \tilde{B}}(\tilde{b} - \tilde{A} u_0). \end{array}$$

to obtain $u_0^{\nu+1}$ may seem discouragingly complicated. How can this be carried out when $\tilde{V}^\nu$ is embedded within $\tilde{V}$ in a space of probably very high dimension, and furthermore when a special representation of the elements of $\tilde{V}^\nu$ is called for? In fact this is the glory of the method. The maximization can be dealt with in terms of the coefficients λ_i themselves and therefore in a space of relatively low dimension.

To see how this comes about, we only have to see what happens when the weighted sum that represents $\tilde{v}$ is substituted into the defining formula

$$\theta_{\tilde{V}^\nu \tilde{B}}(\tilde{b} - \tilde{A}u_0) = \sup_{\tilde{v} \in \tilde{V}^\nu} \Big\{ [\tilde{b} - \tilde{A}u_0] \cdot \tilde{v} - \tfrac{1}{2}\tilde{v} \cdot \tilde{B}\tilde{v} \Big\}.$$

The expression being maximized in this formula is

$$[\tilde{b} - \tilde{A}u_0] \cdot \left[\sum\nolimits_{i=0}^{\nu} \lambda_i \tilde{v}^i \right] - \tfrac{1}{2} \left[\sum\nolimits_{i=0}^{\nu} \lambda_i \tilde{v}^i \right] \cdot \tilde{B} \left[\sum\nolimits_{i=0}^{\nu} \lambda_i \tilde{v}^i \right],$$

which is clearly a quadratic function of $\lambda := (\lambda_0, \lambda_1, \ldots, \lambda_\nu)$. It takes the form

$$[\tilde{b}^\nu - \tilde{A}^\nu u_0] \cdot \lambda - \tfrac{1}{2}\lambda \cdot \Gamma^\nu \lambda$$

for the vector $\tilde{b}^\nu = [\tilde{b} \cdot \tilde{v}^0, \ldots, \tilde{b} \cdot \tilde{v}^\nu]$, the matrix $\tilde{A}^\nu = [\tilde{A}^* \tilde{v}^0, \ldots, \tilde{A}^* \tilde{v}^\nu]$, and the positive semidefinite matrix Γ^ν with entries $\gamma_{ij} = v^i \cdot \tilde{B} v^j$ for $i, j = 0, 1, \ldots, \nu$. This expression has to be maximized over the polyhedral set Λ^ν consisting of all $\lambda = (\lambda_0, \ldots, \lambda_\nu)$ such that $\lambda_i \geq 0$ and $\sum_{i=0}^{\nu} \lambda_i = 1$. What we get here is yet another θ expression: the maximum value is by definition

$$\theta_{\Lambda^\nu \Gamma^\nu}(\tilde{b}^\nu - \tilde{A}^\nu u_0).$$

Low-dimensional ELQP substitute: It follows that the optimization subproblem we have to solve in order to get $u_0^{\nu+1}$ is:

> minimize over $u_0 \in U_0$ the expression
> $c_0 \cdot u_0 + \frac{1}{2} u_0 \cdot C u_0 + \theta_{V_0 B_0}(b_0 - A_{00} u_0) + \theta_{\Lambda^\nu \Gamma^\nu}(\tilde{b}^\nu - \tilde{A}^\nu u_0)$.

This is again a problem in extended linear-quadratic programming! Its dimension is relatively low, so it can be solved directly.

Refinements: In the description just given, the number of points $\tilde{v}_i$ grows by 1 in each iteration, and with it the number of components in the vector $\lambda \in \Lambda^\nu$. As with the supporting hyperplane approach, however, techniques can be developed for discarding points that no longer seem needed, and in this way the dimensionality of the subproblems can be held in check.

5. STATES AND CONTROLS

Problems of optimization under uncertainty typically concern "systems" that undergo changes over time. While our main goal is the exploration of how best to respond to such changes when they are tied in part to the ongoing realization of random variables, there is a substantial body of mathematics, already well appreciated for its many applications, in which the behavior of the system is *deterministic*. We'll go over some of this deterministic theory now for the background in handling dynamical structure that it provides. Then we'll look at its stochastic counterpart.

States and dynamics: One of the lessons learned long ago in mathematical modeling is that many systems can be described in terms of "states" which evolve over time according to certain laws of "dynamics" from a given initial state. A classical model is that of a system with states $x \in \mathbb{R}^d$ governed in continuous time by an ordinary differential equation:

$$\dot{x}(t) = F(t, x(t)) \quad \text{from} \quad x(0) = a,$$

where $\dot{x}(t) = (dx/dt)(t)$. The specification of an initial state a uniquely determines a trajectory of future states $x(t)$ for $t > 0$ (under appropriate assumptions on F).

Control of a system: To model the way that the evolution of states might be influenced in accordance with some goal, one can think of choices being available that affect the dynamics as time goes on. In the framework of an ordinary differential equation, the mapping F may involve parameters that can be represented by a vector $u \in \mathbb{R}^n$, say, and a value $u(t)$ for this parameter vector can be specified at each time t. The dynamical description then becomes

$$\dot{x}(t) = F(t, x(t), u(t)) \quad \text{from} \quad x(0) = a \quad \text{for a choice of} \quad u(\cdot).$$

The control function $u(\cdot)$ yields an expression $G(t, x) = F(t, x, u(t))$, and the differential equation $\dot{x}(t) = G(t, x(t))$ then yields a corresponding state trajectory $x(\cdot)$.

Optimization: A time interval $[0, T]$ can be fixed, and costs and constraints on $x(t)$ and $u(t)$ for $t \in [0, T]$ can be introduced. In this way one gets an optimization problem with respect to the choice of $u(\cdot)$ over some time interval.

Discretization: Problems in continuous time can be approximated by ones in discrete time so as to make them finite-dimensional and more amenable to computation.

Multistage deterministic control: Differential equations and their continuous-time control merely stand as an analogy for what we'll now be occupied with. In the setup about to be described, states and controls have an abstract character, which turns out to be surprisingly useful. Talk of "time" is supplanted by talk of "stages," so as to allow for models in which time, discrete or continuous, can march on between opportunities for intervening in the dynamics. There's no need even to think of the stages as coming from "time." For many applications they may have immediate meaning in a decision process, as was seen in stochastic programming.

Stages: $k = 0, 1, \ldots, N$. The number N gives the model's *horizon*.

Note: If N weren't specified, so that the stages go on and on, one would have an "infinite-horizon" model, but we won't really be treating that case.

States: Elements x_k of the "state spaces" $\mathcal{X}_k$, which are certain nonempty sets for $k = 0, 1, \ldots, N$. Note that the space in question can change from stage to stage.

Controls: Elements u_k of the "control spaces" $\mathcal{U}_k$, which are certain nonempty sets for $k = 0, 1, \ldots, N$. Again, this space can change from stage to stage.

Dynamics: $x_k = F_k(x_{k-1}, u_{k-1})$ for $k = 1, \ldots, N$. The initial state is supposed to be given: $x_0 = a$, a particular element of $\mathcal{X}_0$.

Tacit supposition: $F_k(x_{k-1}, u_{k-1})$ is a well defined element of $\mathcal{X}_k$ for every choice of $x_{k-1} \in \mathcal{X}_{k-1}$ and $u_{k-1} \in \mathcal{U}_{k-1}$.

Costs: $f_k(x_k, u_k)$ for $k = 0, 1, \ldots, N$, with values that may be ∞ as a means of incorporating constraints. A control u_k isn't regarded as feasible in association with the state x_k unless $f_k(x_k, u_k) < \infty$. A state x_k isn't regarded as feasible unless there's at least one such control u_k.

Stage-by-stage description: Controls are viewed as generating states and incurring costs by the following process:

starting from $x_0 = a$,
choose u_0, pay $f_0(x_0, u_0)$, get $x_1 = F_1(x_0, u_0)$,
choose u_1, pay $f_1(x_1, u_1)$, get $x_2 = F_2(x_1, u_1)$,
$\vdots$
choose u_{N-1}, pay $f_{N-1}(x_{N-1}, u_{N-1})$, get $x_N = F_N(x_{N-1}, u_{N-1})$,
choose u_N, pay $f_N(x_N, u_N)$.

Role of the final control: The final control u_N only effects costs; there's no further evolution of the state.

Twin forms of the deterministic optimal control problem: The optimization problem we want to work with has the general, but *preliminary* statement

$$(\mathcal{P}_{\mathrm{d}}) \qquad \begin{aligned} &\text{minimize } \sum\nolimits_{k=0}^{N} f_k(x_k, u_k) \text{ subject to} \\ &x_k = F_k(x_{k-1}, u_{k-1}) \text{ for } k = 1, \dots, N, \text{ with } x_0 = a \end{aligned}$$

(where "d = deterministic"). This statement is preliminary because of ambiguity in what we're actually minimizing over. We don't have a rigorously posed optimization problem until that's resolved. In fact, there are *two* good ways to get an unambiguous problem statement, and we have to be clear about distinguishing them.

Basic form ($\underline{\mathcal{P}}_{\mathrm{d}}$)*:* In one interpretation, seemingly the most natural, only the controls u_k are "decision elements." The states are regarded as determined from them by the dynamics, so that x_k is viewed for $k > 0$ as standing merely for a certain expression in the controls $u_0, \dots, u_{k-1}$. In this mode we have, starting from $x_0 = a$, that

$$\begin{aligned} x_1 &= F_1(a, u_0) =: \chi_1(u_0), \\ x_2 &= F_2\big(\chi_1(u_0), u_1\big) =: \chi_2(u_0, u_1), \\ &\vdots \\ x_k &= F_k\big(\chi_{k-1}(u_0, u_1, \dots, u_{k-2}), u_{k-1}\big) =: \chi_k(u_0, u_1, \dots, u_{k-1}), \\ &\vdots \\ x_N &= F_N\big(\chi_{N-1}(u_0, u_1, \dots, u_{N-2}), u_{N-1}\big) := \chi_N(u_0, u_1, \dots, u_{N-1}). \end{aligned}$$

The term $f_k(x_k, u_k)$ is accordingly just shorthand for $f_k(\chi_k(u_0, u_1, \dots, u_{k-1}), u_k)$ when $k > 0$, whereas $f_0(x_0, u_0) = f_0(a, u_0)$. The aim is to minimize the sum of these terms over the space $\mathcal{U}_0 \times \mathcal{U}_1 \times \cdots \times \mathcal{U}_N$.

Full form ($\overline{\mathcal{P}}_{\mathrm{d}}$)*:* In the other interpretation, which for some mathematical purposes is more advantageous, the states are decision elements along with the controls, and *the dynamical equations are regarded as constraints.* In this case the sum of terms $f_k(x_k, u_k)$ is viewed as a function of all these elements, and the minimization takes place over the space $\mathcal{X}_0 \times \mathcal{X}_1 \times \cdots \times \mathcal{X}_N \times \mathcal{U}_0 \times \mathcal{U}_1 \times \cdots \times \mathcal{U}_N$.

Status of the initial state: The double view of what we are optimizing over is reflected in the notation in the initial stage, which refers to a general x_0 when

it seems we could just replace x_0 by a. In the full form of the problem, the equation $x_0 = a$ is a constraint on x_0 as a variable.

Generality of the model: The stage-by-stage pattern we've described is convenient in many ways, but to what extent is such model structure restrictive? Could it be necessary to abandon it eventually in order to achieve greater generality in the optimization of deterministic decision processes over time? The answer is no. In explaining this, we begin with two minor observations.

Problems with no final control: In some situations it's convenient to drop the final control vector u_N from the model. The cost term $f_N(x_N, u_N)$ then becomes $f_N(x_N)$, and there's no space $\mathcal{U}_N$. Everything else stays the same.

Problems with no initial state: In other situations it's convenient to drop the initial state vector x_0. Then there is no space $\mathcal{X}_0$ or data vector a. The cost term $f_0(x_0, u_0)$ simplifies to just $f_0(u_0)$, and the state first seen is $x_1 = F_0(u_0)$.

Coverage of problems with no apparent 'states' at all: To see how a structure of states and controls can always be deemed to be present in principle, even if not visible on the surface, consider the extreme case of a problem

$$\text{minimize } f(u_0, u_1, \ldots, u_N) \text{ over } u_k \in \mathbb{R}^{n_k} \text{ for } k = 0, 1, \ldots, N,$$

where f possibly incorporates constraints through ∞ values. We can think of this as a multistage problem in which u_k is the control vector in stage k, and the "history vector" $(u_0, u_1, \ldots, u_{k-1})$ is the state vector x_k in stage k. (This is an example where there is no initial state x_0.)

The action begins with $x_1 = u_0$ and continues with $x_k = (x_{k-1}, u_{k-1})$ for $k = 2, \ldots, N$; thus, x_k is obtained by adjoining u_{k-1} to the end of x_{k-1} to make an even longer vector—here we have an instance where the dimension of the state space definitely isn't the same from stage to stage, but rather increases. Finally, we take $f_0(u_0) \equiv 0$ and $f_k(x_k, u_k) \equiv 0$ for $k = 1, \ldots, N-1$, but $f_N(x_N, u_N) = f(u_0, u_1, \ldots, u_{N-1}, u_N)$, which makes sense because $x_N = (u_0, u_1, \ldots, u_{N-1})$. Note that this dynamical system is actually linear, as a special case of the following.

Linear dynamics: As an important example, the state spaces $\mathcal{X}_k$ to $\mathbb{R}^{d_k}$, the control spaces $\mathcal{U}_k$ to $\mathbb{R}^{n_k}$, and the dynamics taken to be

$$x_k = A_k x_{k-1} + B_k u_{k-1} + b_k \text{ for } k = 1, \ldots, N, \text{ with } x_0 = a.$$

Here $A_k \in \mathbb{R}^{d_k \times d_{k-1}}$, $B_k \in \mathbb{R}^{d_k \times n_{k-1}}$, and $b_k \in \mathbb{R}^{d_k}$. (One speaks of "linear" dynamics even though $F_k(x_{k-1}, u_{k-1})$ is merely affine in x_{k-1} and u_{k-1} when $b_k \neq 0$.)

Interpretation in basic form: In the basic form $(\underline{\mathcal{P}}_{\mathrm{d}})$ of an optimal control problem with such dynamics, the x_k's symbolize affine expressions in the u_k's:

$$\begin{aligned} x_0 &= a, \\ x_1 &= A_0 a + B_0 u_0 + b_0, \\ x_2 &= A_1\big[A_0 a + B_0 u_0\big] + B_1 u_1 + b_1, \\ x_3 &= A_2\big[A_1\big[A_0 a + B_0 u_0\big] + B_1 u_1 + b_1\big] + B_2 u_2 + b_2, \end{aligned}$$

and so forth. The minimization takes place over the u_k's.

Interpretation in full form: In the full form $(\overline{\mathcal{P}}_{\mathrm{d}})$, however, the minimization takes place jointly over the x_k's and the u_k's, subject to the constraints

$$\begin{aligned} x_0 - a &= 0, \\ x_1 - A_1 x_0 + B_1 u_0 - b_1 &= 0, \\ x_2 - A_2 x_1 + B_2 u_1 - b_2 &= 0, \\ x_3 - A_3 x_2 + B_3 u_2 - b_3 &= 0, \end{aligned}$$

and so forth. An interesting aspect then is the introduction of multiplier vectors $y_k \in \mathbb{R}^{d_k}$ for these equations. In optimality conditions, such vectors come out as "dual states" governed by a "dual dynamical system," as will soon be seen.

ELQP model in deterministic control: Building on a linear dynamical system as just described, suppose now that the costs are given by

$$f_k(x_k, u_k) = \delta_{U_k}(u_k) + r_k \cdot u_k + \tfrac{1}{2} u_k \cdot R_k u_k + \theta_{V_k S_k}\big(s_k - C_k x_k - D_k u_k\big) - c_k \cdot x_k$$

for nonempty polyhedral sets $U_k \subset \mathbb{R}^{n_k}$ and $V_k \subset \mathbb{R}^{m_k}$, symmetric positive semi-definite matrices $R_k \in \mathbb{R}^{n_k \times n_k}$ and $S_k \in \mathbb{R}^{m_k \times m_k}$, matrices $C_k \in \mathbb{R}^{m_k \times d_k}$ and $D_k \in \mathbb{R}^{m_k \times n_k}$, and vectors $r_k \in \mathbb{R}^{n_k}$, $s_k \in \mathbb{R}^{m_k}$ and $c_k \in \mathbb{R}^{d_k}$. Recall that

$$\theta_{V_k S_k}(w) := \sup_{v_k \in V_k} \Big\{ w \cdot v_k - \tfrac{1}{2} v_k \cdot S_k v_k \Big\}$$

and that δ_{U_k} is the indicator of U_k, i.e., δ_{U_k} vanishes on U_k but is ∞ outside of U_k. In its general statement, the optimal control problem is thus to

$$(\mathcal{P}_{\mathrm{elpqd}}) \qquad \begin{array}{ll} \min & \sum_{k=0}^{N} \Big(r_k \cdot u_k + \tfrac{1}{2} u_k \cdot R_k u_k + \theta_{V_k S_k}\big(s_k - C_k x_k - D_k u_k\big) - c_k \cdot x_k \Big) \\ \text{subject to} & \begin{cases} x_0 = a, & \\ x_k = A_k x_{k-1} + B_k u_{k-1} + b_k & \text{for } k = 1, \ldots, N, \\ u_k \in U_k & \text{for } k = 0, 1, \ldots, N, \end{cases} \end{array}$$

(where the "d" in "elqpd" refers to this being a *deterministic* control model).

LP example: A linear programming version, more or less in canonical format, can be obtained by specializing U_k and V_k to $\mathbb{R}^{n_k}_+$ and $\mathbb{R}^{m_k}_+$ while taking both $R_k = 0$ and $S_k = 0$. Then we have

$$f_k(x_k, u_k) = \begin{cases} r_k \cdot u_k - c_k \cdot x_k & \text{if } u_k \geq 0 \text{ and } C_k x_k + D_k u_k \geq q_k, \\ \infty & \text{otherwise.} \end{cases}$$

Tracking example: Another good case to keep in mind is the one in which $V_k = \mathbb{R}^{m_k}$, S_k is nonsingular, $D_k = 0$ and $c_k = 0$. Then the expression being minimized has the form

$$\sum\nolimits_{k=0}^{N} \Big(\delta_{U_k}(u_k) + r_k \cdot u_k + \tfrac{1}{2} u_k \cdot R_k u_k + \tfrac{1}{2} |q_k - C_k x_k|_k^2 \Big), \quad \text{with } |z|_k^2 := z \cdot S_k^{-1} z.$$

In this case we're choosing controls $u_k \in U_k$, costing $r_k \cdot u_k + \frac{1}{2} u_k \cdot R_k u_k$, with the aim of making the vector sequence $\{C_k x_k\}_{k=0}^N$ track (i.e., follow) a given sequence $\{q_k\}_{k=0}^N$. The norms $|\cdot|_k$ are used to quantify the tracking errors, and the objective function balances these errors against the control costs. In particular one could have $S_k = \varepsilon I$, so that $\frac{1}{2}|\cdot|^2 = \frac{1}{2\varepsilon}|\cdot|^2$. Then the smaller the value of ε the more weight would be given to insisting on the closeness of $C_k x_k$ to q_k.

Note that in the still more special case with $U_k = \mathbb{R}^{n_k}$, $r_k = 0$ and $q_k = 0$ for all k, the total cost is

$$\sum\nolimits_{k=0}^{N} \Big(\tfrac{1}{2} x_k \cdot Q_k x_k + \tfrac{1}{2} u_k \cdot R_k u_k \Big), \quad \text{where } Q_k := C_k^* S_k^{-1} C_k.$$

This is a standard setup in optimal control. Any symmetric, positive semidefinite matrix Q_k can of course be represented (in more than one way) as $C_k^* S_k^{-1} C_k$ for some choice of C_k and positive definite S_k (e.g., $S_k = I$, as is always possible).

Full Lagrangian: In its full form ($\overline{\mathcal{P}}_{\text{elqpd}}$), the optimal control problem ($\mathcal{P}_{\text{elqpd}}$) can be identified as the primal problem associated with a certain expression $\overline{L}(x, u; y, v)$ in $(x, u) \in [\mathbb{R}^d \times U]$ and $(y, v) \in [\mathbb{R}^d \times V]$, where

$$\mathbb{R}^d = \mathbb{R}^{d_0} \times \cdots \times \mathbb{R}^{d_N}, \qquad U := U_0 \times \cdots \times U_N, \qquad V := V_0 \times \cdots \times V_N$$

along with the notation

$$\begin{aligned} x = (x_0, \ldots, x_N) \in \mathbb{R}^d, \qquad & u = (u_0, \ldots, u_N) \in U, \\ y = (y_0, \ldots, y_N) \in \mathbb{R}^d, \qquad & v = (v_0, \ldots, v_N) \in V. \end{aligned}$$

The function $\overline{L}$ in question is defined on $[\mathbb{R}^d \times U] \times [\mathbb{R}^d \times V]$ by

$$\begin{aligned}\overline{L}(x,u;y,v) = \sum_{k=0}^{N}\Big(r_k\cdot u_k + \tfrac{1}{2}u_k\cdot R_k u_k - c_k\cdot x_k\Big) \\ + \sum_{k=0}^{N}\Big(v_k\cdot[s_k - C_k x_k - D_k u_k] - \tfrac{1}{2}v_k\cdot S_k v_k\Big) \\ + y_0\cdot[x_0 - a] + \sum_{k=1}^{N} y_k\cdot[x_k - A_k x_{k-1} - B_k u_{k-1} - b_k].\end{aligned}$$

Indeed, $(\overline{\mathcal{P}}_{\rm elqpd})$ consists of minimizing $\overline{f}(x,u)$ over $\mathbb{R}^d \times U$, where

$$\overline{f}(x,u) := \sup_{(y,v)\in \mathbb{R}^d\times V} \overline{L}(x,u;y,v).$$

Thus $\overline{L}$, on $[\mathbb{R}^d \times U] \times [\mathbb{R}^d \times V]$, serves as a Lagrangian for $(\overline{\mathcal{P}}_{\rm elqpd})$.

ELQP status: Since the sets $[\mathbb{R}^d \times U]$ and $[\mathbb{R}^d \times V]$ are polyhedral, while $\overline{L}(x,u;y,v)$ is a linear-quadratic expression that is convex in (x,u) and concave in (y,v), we conclude that $(\overline{\mathcal{P}}_{\rm elqpd})$ is a problem of extended linear-quadratic programming.

Methodological implications: In principle we may therefore contemplate solving for an optimal solution $(\bar{x},\bar{u})$ by a way of any numerical method in extended linear-quadratic programming that's capable of coping with the dimensionality that might be faced. It's valuable also, though, to explore the special structure of the problem that might emerge from its optimality conditions, since that could lay a foundation for more specialized solution techniques.

Optimality in the ELQP model: On the basis of our full Lagrangian representation, we can conclude from the theory of extended linear-quadratic programming that:

> *A pair $(\bar{x},\bar{u})$ is optimal for $(\overline{\mathcal{P}}_{\rm elqpd})$ if and only if there is a pair $(\bar{y},\bar{v})$ such that $(\bar{x},\bar{u};\bar{y},\bar{v})$ furnishes a saddle point of $\overline{L}$ on $[\mathbb{R}^d \times U] \times [\mathbb{R}^d \times V]$. Furthermore, $(\bar{y},\bar{v})$ fills this role if and only if it solves the associated dual problem.*

Let's see what can be developed out these facts in conjunction with the highly specialized Lagrangian structure.

Re-expression of the full Lagrangian: The Lagrangian $\overline{L}$ can fruitfully be rewritten as the special sum

$$\begin{aligned}\overline{L}(x,u;y,v) = \sum\nolimits_{k=0}^{N} l_k(u_k,v_k) + \bar{l}(x,u;y,v), \quad \text{with} \\ l_k(u_k,v_k) := r_k\cdot u_k + \tfrac{1}{2}u_k\cdot R_k u_k + v_k\cdot s_k - \tfrac{1}{2}v_k\cdot S_k v_k - v_k\cdot D_k u_k,\end{aligned}$$

where the term $\bar{l}(x,u;y,v)$ embodies everything about the dynamical connections between the various stages and can be expressed either as

$$\begin{aligned}\bar{l}(x,u;y,v) = {} & y_0\cdot[x_0-a]+\sum\nolimits_{k=1}^{N} y_k\cdot[x_k-A_kx_{k-1}-B_ku_{k-1}-b_k]\\ & -\sum\nolimits_{k=0}^{N} v_k\cdot[C_kx_k]-\sum\nolimits_{k=0}^{N} c_k\cdot x_k.\end{aligned}$$

or equivalently in terms of the transpose matrices as

$$\begin{aligned}\bar{l}(x,u;y,v) = {} & \sum\nolimits_{k=0}^{N-1} x_k\cdot[y_k-A_{k+1}^*y_{k+1}-C_k^*v_k-c_k]+x_N\cdot[y_N-C_N^*v_N-c_N]\\ & -\sum\nolimits_{k=0}^{N-1} u_k\cdot[B_{k+1}^*y_{k+1}]-\sum\nolimits_{k=1}^{N} b_k\cdot y_k-a\cdot y_0.\end{aligned}$$

Here $\bar{l}(x,u;y,v)$ is affine in (x,u) for fixed (y,v), and also in (y,v) for fixed (x,u).

Dualization: The two equivalent ways of expressing $\bar{l}(x,u;y,v)$ make it easy to determine the dual problem associated with this Lagrangian setup. On general grounds, the dual problem consists of maximizing $\overline{g}(y,v)$ over $\mathbb{R}^d\times V$, where

$$\overline{g}(y,v) := \inf_{(x,u)\in\mathbb{R}^d\times U} \overline{L}(x,u;y,v).$$

From the second formula for $\bar{l}(x,u;y,v)$ we can readily see that the problem thereby obtained is the *full* form ($\overline{\mathcal{D}}_{\mathrm{elqpd}}$) of the following *dual* problem

$$(\mathcal{D}_{\mathrm{elqpd}})\qquad \begin{array}{l}\max\ \sum\nolimits_{k=0}^{N}\Big(v_k\cdot s_k-\frac{1}{2}v_k\cdot S_kv_k\Big)-\sum\nolimits_{k=1}^{N} b_ky_k-a\cdot y_0\\ \qquad -\sum\nolimits_{k=0}^{N-1}\theta_{U_kR_k}\big(B_{k+1}^*y_{k+1}+D_k^*v_k-r_k\big)-\theta_{U_NR_N}\big(D_N^*v_N-r_N\big)\\ \text{subject to}\ \begin{cases} y_N = C_N^*v_N+c_N, & \\ y_k = A_{k+1}^*y_{k+1}+C_k^*v_k+c_k & \text{for } k=0,\dots,N-1,\\ v_k\in V_k & \text{for } k=0,\dots,N-1,N,\end{cases}\end{array}$$

i.e., the version of this problem in which the minimization takes place over the y_k's and v_k's jointly, and the equations for y_k's are treated as constraints. This optimization problem too, of course, is a problem of extended linear-quadratic programming.

Dual dynamics: The vectors y_k are *dual states*, whereas the vectors v_k are *dual controls*. They obey a dynamical system that goes *backward* from stage N to stage 0. From the choice of v_N one gets $y_N = C_N^*v_N+c_N$. Stage by stage thereafter, one gets y_k from y_{k+1} and v_k by $y_k = A_{k+1}^*y_{k+1}+C_k^*v_k+c_k$.

Optimality stage by stage: The saddle point condition on $(\bar{x},\bar{u})$ and $(\bar{y},\bar{v})$, which we know to be equivalent to $(\bar{x},\bar{u})$ solving $(\overline{\mathcal{P}}_{\mathrm{elqpd}})$ and $(\bar{y},\bar{v})$ solving $(\overline{\mathcal{D}}_{\mathrm{elqpd}})$, boils down now to the following:

$$\text{(a)}\quad \begin{cases} \bar{x}_k = A_k\bar{x}_{k-1} + B_k\bar{u}_{k-1} + b_k \quad \text{for } k = 1,\dots,N, \\ \bar{x}_0 = a, \end{cases}$$

$$\text{(b)}\quad \begin{cases} \bar{y}_k = A^*_{k+1}\bar{y}_{k+1} + C^*_k\bar{v}_k + c_k \quad \text{for } k = 0,\dots,N-1, \\ \bar{y}_N = C^*_N\bar{v}_N + c_N, \end{cases}$$

$$\text{(c)}\quad \bar{u}_k \in \operatorname*{argmin}_{u_k\in U_k}\Big\{l_k(u_k,\bar{v}_k) - u_k\cdot[B_{k+1}\bar{y}_{k+1}]\Big\} \text{ for } k = 0,\dots,N-1,$$

$$\bar{u}_N \in \operatorname*{argmin}_{u_N\in U_N}\Big\{l_N(u_N,\bar{v}_N)\Big\},$$

$$\text{(d)}\quad \bar{v}_k \in \operatorname*{argmax}_{v_k\in V_k}\Big\{l_k(\bar{u}_k,v_k) - v_k\cdot[C_k\bar{x}_k]\Big\} \text{ for } k = 0,1,\dots,N.$$

Interestingly, this consists just of the primal dynamics, the dual dynamics, and a condition on $\bar{u}_k$ and $\bar{v}_k$ in stage k which only depends on the data embodied in l_k, U_k and V_k, and the state vector images $C_k\bar{x}_k$ and $B^*_{k+1}\bar{y}_{k+1}$.

Interpretation through stage-based ELQP subproblems: According to this, optimal primal and dual controls $\bar{u}_k$ and $\bar{v}_k$ are characterized by the following property:

> *Let $\bar{x}$ and $\bar{y}$ be the primal and dual trajectories generated from $\bar{u}$ and $\bar{v}$ by the dynamics. Then in each stage k we have an extended linear-quadratic programming problem $(\widetilde{\mathcal{P}}_k)$, depending on $C_k\bar{x}_k$ and $B^*_{k+1}\bar{y}_{k+1}$, such that $\bar{u}_k$ solves $(\widetilde{\mathcal{P}}_k)$ and $\bar{v}_k$ solves the corresponding dual problem $(\widetilde{\mathcal{D}}_k)$.*

In fact the primal and dual subproblems are:

$$(\widetilde{\mathcal{P}}_k)\qquad \min_{u_k\in U_k}: \quad [r_k - B^*_{k+1}\bar{y}_{k+1}]\cdot u_k + \tfrac{1}{2}u_k\cdot R_k u_k + \theta_{V_kS_k}\big(s_k - C_k\bar{x}_k - D_k u_k\big)$$

$$(\widetilde{\mathcal{D}}_k)\qquad \max_{v_k\in V_k}: \quad v_k\cdot[s_k - C_k\bar{x}_k] - \tfrac{1}{2}v_k\cdot S_k v_k - \theta_{U_kR_k}\big(B^*_{k+1}\bar{y}_{k+1} + D^*_k v_k - r_k\big),$$

except that the $B^*_{k+1}\bar{y}_{k+1}$ terms drop out when $k = N$. These subproblems isolate the optimization in stage k controls to stage k itself, accomplishing this by supplementing the stage k data by means of primal information generated forward from the past through the primal states and dual information generated backward from the future through the dual states.

Explanation: Conditions (c) and (d) above correspond to $(\bar{u}_k,\bar{v}_k)$ being a saddle point on $U_k\times V_k$ of the function

$$\widetilde{l}_k(u_k,v_k) := l_k(u_k,v_k) - u_k\cdot B^*_{k+1}\bar{y}_{k+1} - v_k\cdot C_k\bar{x}_k.$$

This means that $\bar{u}_k$ and $\bar{v}_k$ solve the primal and dual problems of extended linear-quadratic programming that are associated with the triple $\widetilde{l}_k$, U_k, V_k, and these problems are $(\widetilde{\mathcal{P}}_k)$ and $(\widetilde{\mathcal{D}}_k)$.

Duality for control problems in their basic form: We've seen that the problems $(\mathcal{P}_{\rm elqpd})$ and $(\mathcal{D}_{\rm elqpd})$, when interpreted in their full forms $(\overline{\mathcal{P}}_{\rm elqpd})$ and $(\overline{\mathcal{D}}_{\rm elqpd})$, are dual to each other with respect to the *full* Lagrangian $\overline{L}$ on $[\mathbb{R}^d \times U] \times [\mathbb{R}^d \times V]$. But they can also be seen as dual to each other when interpreted in their basic forms $(\underline{\mathcal{P}}_{\rm elqpd})$ and $(\underline{\mathcal{D}}_{\rm elqpd})$, in which only the control vectors are decision elements.

For this purpose we merely have to introduce the *basic* Lagrangian $\underline{L}$ on $U \times V$ by taking

$$\underline{L}(u,v) = \sum\nolimits_{k=0}^{N} l_k(u_k, v_k) + \underline{l}(u,v)$$

with $l_k(u_k, v_k)$ the same as above, and with $\underline{l}(u,v)$ defined to be the value obtained from $\bar{l}(x,u;y,v)$ when x is viewed as standing for the associate expression in u, generated from the primal dynamics, while y is viewed as standing for the associate expression in v, generated from the dual dynamics. It's clear from the analysis we've already made that $(\underline{\mathcal{P}}_{\rm elqpd})$ can be identified with the problem of minimizing

$$\underline{f}(u) := \sup_{v \in V} \underline{L}(u,v) \text{ for } u \in U,$$

whereas $(\underline{\mathcal{D}}_{\rm elqpd})$ can be identified with the problem of maximizing

$$\underline{g}(v) := \inf_{u \in U} \underline{L}(u,v) \text{ for } v \in V.$$

This confirms the claimed relationship of "basic" duality—again in the ELQP framework. Of course, the saddle point condition for optimality that's obtained by looking at the primal and dual control problems in this basic form is equivalent to the one obtained by looking at them in their full form.

ELQP status revisited: We see that $(\mathcal{P}_{\rm elqpd})$ comes out as a problem of extended linear-quadratic programming regardless of whether it's interpreted in its basic form or in its full form. Moreover the same holds for the dual problem $(\mathcal{D}_{\rm elqpd})$.

Multistage stochastic control: Let's now extend the abstract deterministic model of multistage control to allow for *stochastic* costs and dynamics. As before, in stages $k = 0, 1, \ldots, N$ there are states x_k belonging to state spaces $\mathcal{X}_k$ and controls u_k belonging to control spaces $\mathcal{U}_k$, along with costs $f_k(x_k, u_k)$. In addition, however, there are uncertain elements ω_k in spaces Ω_k, and these can influence the dynamics:

$$x_k = F_k(x_{k-1}, u_{k-1}, \omega_k) \text{ for } k = 1, \ldots, N, \text{ with } x_0 = a.$$

The stochastic aspects of these elements are allowed, in general, to evolve with the states and controls.

Probability distributions: The probability distribution for ω_k in Ω_k can depend on x_{k-1} and u_{k-1} and is denoted by $\pi_k(x_{k-1}, u_{k-1})$. Thus, in our discrete probability framework the probability of a particular element $\omega_k \in \Omega_k$ is a number

$$\pi_k(x_{k-1}, u_{k-1}; \omega_k) \geq 0, \quad \text{with} \sum_{\omega_k \in \Omega_k} \pi_k(x_{k-1}, u_{k-1}; \omega_k) = 1.$$

In "continuous probability," the distribution $\pi_k(x_{k-1}, u_{k-1})$ would correspond to a density function $\rho_k(x_{k-1}, u_{k-1}, \omega_k)$ of $\omega_k \in \Omega_k$, and so forth.

Stage-by-stage description: It's crucial to understand the order of things—what comes before what, and what can depend on what. The process being modeled has features like those in multistage deterministic control, but also resembles multistage stochastic programming, except for a much broader treatment of probabilities:

start from $x_0 = a$,

choose u_0, pay $f_0(x_0, u_0)$,

observe ω_1 from $\pi_1(x_0, u_0)$, get $x_1 = F_1(x_0, u_0, \omega_1)$

choose u_1, pay $f_1(x_1, u_1)$,

observe ω_2 from $\pi_2(x_1, u_1)$, get $x_2 = F_2(x_1, u_1, \omega_2)$,

choose u_2, pay $f_2(x_2, u_2)$,

$\vdots$

observe ω_k from $\pi_k(x_{k-1}, u_{k-1})$, get $x_k = F_k(x_{k-1}, u_{k-1}, \omega_k)$,

choose u_k, pay $f_k(x_k, u_k)$,

$\vdots$

observe ω_N from $\pi_N(x_{N-1}, u_{N-1})$, get $x_N = F_N(x_{N-1}, u_{N-1}, \omega_N)$,

choose u_N, pay $f_N(x_N, u_N)$.

As we now know well, this pattern doesn't by itself give us an optimization problem but simply furnishes the raw materials out of which an optimization problem might be formulated. To arrive at such a problem, we'll need to work with control *policies* of some kind.

Certainty versus uncertainty in the states: A key idea in the process description is knowing the state x_k at the time of choosing u_k, and thus knowing everything

about the function f_k then as well, including any implicit constraints that stem from ∞ values of f_k. The successor state x_{k+1}, on the other hand, isn't known at that time except as a random element whose distribution depends on that of ω_{k+1}, and hence parametrically on x_k and u_k, through the equation $x_{k+1} = F_{k+1}(x_k, u_k, \omega_{k+1})$ and the distribution $\pi_{k+1}(x_k, u_k)$ for ω_{k+1}.

Cost modeling: One might imagine situations in which, instead of $f_k(x_k, u_k)$, the cost term for stage k has the form $f_k(x_k, u_k, \omega_{k+1})$. There's really no need for that, however. The underlying question is whether a subsequent decision might alter this cost. If yes, the cost ought to be regarded as paid at some later stage. If no, one can simply replace it by its expectation relative to the distribution $\pi_{k+1}(x_k, u_k)$ for ω_{k+1}. That would convert it back to the case of a value that's known on the basis of knowing x_k and u_k.

Status of observations: In speaking of the "observation" of ω_k at the start of stage k, we are effectively taking the position that, from this stage on, ω_k has a known value. It's from knowing ω_k along with x_{k-1} and u_{k-1} that we know x_k. For now, this is indeed how we wish to look at things, but other approaches are interesting as well. In particular, we could have the view that it isn't ω_k we observe, but just the resultant state x_k itself. We'll get into this variant and others in the next chapter.

Fundamental policies: For the rest of this chapter, we put ourselves in the perspective of stage 0 and consider that when a stage $k > 0$ is reached we'll have a record available of the observations $\omega_1, \ldots, \omega_k$. We allow the decision to be made in this stage to respond to these observations and thus model it by a function

$$u_k(\cdot) : (\omega_1, \ldots, \omega_k) \mapsto u_k(\omega_1, \ldots, \omega_k),$$

as in basic stochastic programming. A mapping

$$u(\cdot) : \omega \mapsto \big(u_0, u_1(\omega_1), u_2(\omega_1, \omega_2), \ldots, u_N(\omega_1, \omega_2, \ldots, \omega_N)\big)$$

will be called a *fundamental* policy. The problem we'll next formulate will concern optimization over such policies, which are also termed the *nonanticipative* mappings from the space $\Omega = \Omega_1 \times \cdots \times \Omega_N$ to the space $\mathcal{U}_0 \times \mathcal{U}_1 \times \cdots \times \mathcal{U}_N$.

Feedback alternative: We'll later be looking instead at schemes in which we regard x_k as the sole item of information to which we can respond in choosing u_k in stage k. Policies then will be *feedback* policies, whose components have the form $u_k(\cdot) : x_k \mapsto u_k(x_k)$ as mappings from $\mathcal{X}_k$ to $\mathcal{U}_k$.

Consequences of executing a fundamental policy: Having selected a (fundamental) policy $u(\cdot)$, the following is what we get in the future—as seen from the standpoint of the present, with the random elements as yet unresolved.

Evolution of states: The states come out as functions of the observations that precede them. From the policy $u(\cdot) = (u_0, u_1(\cdot), \ldots, u_N(\cdot))$ we obtain a *trajectory* $x(\cdot) = (x_0, x_1(\cdot), \ldots, x_N(\cdot))$, which is likewise a nonanticipative mapping, by setting

$$\begin{aligned} x_0 &= a, \\ x_1(\omega_1) &= F_1(x_0, u_0, \omega_1), \\ x_2(\omega_1, \omega_2) &= F_2(x_1(\omega_1), u_1(\omega_1), \omega_2), \\ &\vdots \\ x_N(\omega_1, \ldots, \omega_N) &= F_N(x_{N-1}(\omega_1, \ldots, \omega_{N-1}), u_{N-1}(\omega_1, \ldots, \omega_{N-1}), \omega_N). \end{aligned}$$

Evolution of probabilities: Once stage N is reached, a probability distribution π will have been constructed over $\Omega = \Omega_1 \times \cdots \times \Omega_N$ for $\omega = (\omega_1, \ldots, \omega_N)$. Such a distribution isn't known prior to that, because of its possible dependence on the policy $u(\cdot)$ and the trajectory $x(\cdot)$. It emerges $u(\cdot)$ and $x(\cdot)$ only as follows. On entering stage 1, we have

$$\pi^1 := \pi_1(x_0, u_0) : \text{ the marginal distribution of } \omega_1,$$

i.e., the distribution that will turn out to be the projection on Ω_1 of the eventual distribution π on Ω. Then through $x_1(\cdot)$ and $u_1(\cdot)$ we get

$$\pi_2\big(x_1(\omega_1), u_1(\omega_1)\big) : \text{ the conditional distribution of } \omega_2 \text{ given } \omega_1.$$

The marginal distribution π^1 of ω_1 and this conditional distribution for ω_2, given ω_1, combine to produce

$$\pi^2 := \text{ the marginal distribution of } (\omega_1, \omega_2)$$

for the start of stage 2. Similarly, after entering a stage $k-1$ with

$$\pi^{k-1} := \text{ the marginal distribution of } (\omega_1, \ldots, \omega_{k-1}),$$

we proceed by way of $x_{k-1}(\omega_1, \ldots, \omega_{k-1})$ and $u_{k-1}(\omega_1, \ldots, \omega_{k-1})$ to

$$\pi_k\big(x_{k-1}(\omega_1, \ldots, \omega_{k-1}), u_{k-1}(\omega_1, \ldots, \omega_{k-1})\big) : \\ \text{the conditional distribution of } \omega_k \text{ given } (\omega_1, \ldots, \omega_{k-1}).$$

The marginal distribution π^{k-1} of $(\omega_1, \ldots, \omega_{k-1})$ and this conditional distribution for ω_k, given $(\omega_1, \ldots, \omega_{k-1})$, combine then to produce

$$\pi^k := \text{ the marginal distribution of } (\omega_1, \ldots, \omega_{k-1}, \omega_k).$$

When finally $k = N$, we have the entire distribution $\pi^N = \pi$ for ω on Ω.

Mathematical technicalities: In general, the development of these conditional and marginal distributions is complicated to justify. It's an important topic in probability theory, requiring careful assumptions. There's no trouble, though, in the setting of discrete probability.

Evolution of costs: From the policy $u(\cdot)$ and trajectory $x(\cdot)$ we also obtain expressions for the costs paid in the various stages as functions of the uncertain elements:

$$\begin{aligned} c_0 &:= f_0(x_0, u_0), \\ c_1(\omega_1) &:= f_1\big(x_1(\omega_1), u_1(\omega_1))\big), \\ c_2(\omega_1, \omega_2) &:= f_2\big(x_2(\omega_1, \omega_2), u_2(\omega_1, \omega_2)\big), \\ &\vdots \\ c_N(\omega_1, \ldots, \omega_N) &:= f_N\big(x_N(\omega_1, \ldots, \omega_N), u_N(\omega_1, \ldots, \omega_N)\big). \end{aligned}$$

The corresponding total cost

$$c_0 + c_1(\omega_1) + c_2(\omega_1, \omega_2) + \cdots + c_N(\omega_1, \ldots, \omega_N)$$

is a random variable from the perspective of stage 0, but its expectation can be taken relative to the probability distribution π that comes out of $u(\cdot)$ and $x(\cdot)$, and this expected value can be regarded as known in stage 0.

Twin forms of the stochastic optimal control problem: The optimization problem now has the general statement

$$(\mathcal{P}_\mathrm{s}) \qquad \begin{array}{l} \text{minimize } \mathbb{E}\Big\{ \sum_{k=0}^{N} f_k(x_k(\omega_1, \ldots, \omega_k), u_k(\omega_1, \ldots, \omega_k)) \Big\} \text{ subject to} \\ \begin{cases} x_k(\omega_1, \ldots, \omega_k) = F_k(x_{k-1}(\omega_1, \ldots, \omega_{k-1}, u_{k-1}(\omega_1, \ldots, \omega_{k-1}), \omega_k) \\ \text{for } k = 1, \ldots, N, \quad \text{with } x_0 = a \end{cases} \end{array}$$

(where "s=stochastic"). The expectation is taken over the probability distribution for ω that's generated in parallel with the states and controls. But as in the deterministic case, this problem statement is merely *preliminary*. There are two ways of interpreting it which have to be distinguished.

Basic form ($\underline{\mathcal{P}}_{\rm s}$)*:* In the basic version, each $x_k(\omega_1,\ldots,\omega_k)$ is regarded as standing for a certain expression in a, u_0, $u_1(\omega_1),\ldots,u_{k-1}(\omega_1,\ldots,\omega_{k-1})$ as developed from the dynamics. The minimization therefore takes place over the space of all fundamental policies $u(\cdot)$.

Full form ($\overline{\mathcal{P}}_{\rm s}$)*:* In the full version, the minimization takes place not only with respect to policies $u(\cdot)$ but also with respect to trajectories $x(\cdot) = (x_0, x_1(\cdot),\ldots,x(\cdot))$ (nonanticipative). The dynamical equations are regarded then as constraints on x_0, $x_1(\omega_1),\ldots,x_N(\omega_1,\ldots,\omega_N)$ as well as u_0, $u_1(\omega_1),\ldots,u_N(\omega_1,\ldots,\omega_N)$.

Convexity: The road is now wide open to the discussion of linear dynamics, extended linear-quadratic cost expressions, and so forth. Essentially, everything we looked at in the deterministic case would have an analog in this stochastic case, if it weren't for one big hang-up:

> *In allowing the probability distributions for ω to be influenced by the states and controls, there is a very serious danger of convexity being lost.*

When that happens, the optimization problem is prone to becoming intractable numerically, at least in this formulation.

How basic stochastic programming fits in: Stochastic programming, as developed so far, corresponds to the special case of stochastic optimal control in which

(a) $x_k = (\omega_1,\ldots,\omega_k)$ for $k > 0$ (no initial state),

(b) $F_k(x_{k-1}, u_{k-1}, \omega_k) = (x_{k-1}, \omega_k)$ for $k > 1$ (whereas $F_1(\omega_1) = \omega_1$),

(c) $\pi_k(x_{k-1}, u_{k-1})$ is independent of u_{k-1},

(d) $f_k(x_k, u_k)$ is convex in u_k.

The effect of (c) in combination with (a) and (b) is the freeing up of the probability structure from being influenced by the decisions u_k that are selected. This independence is essential also in obtaining, through (d), the convexity of the problem.

Advanced stochastic programming: Convexity can in fact be maintained under weaker assumptions. The key is to divide the state into two parts: $x_k = (x'_k, x''_k)$, where $x''_k = (\omega_1,\ldots,\omega_k)$ as above, but x'_k is subject to a linear system of dynamics in the controls u_k, with matrices that can depend on $(\omega_1,\ldots,\omega_k)$. The distributions $\pi_k(x_{k-1}, u_{k-1}) = \pi_k(x'_{k-1}, x''_{k-1}, u_{k-1})$ are taken to be independent of x'_{k-1} and u_{k-1}, and the costs $f_k(x_k, u_k) = f_k(x'_k, x''_k, u_k)$ to be convex with respect to (x'_k, u_k). We'll develop this advanced model—in better notation—after examining the alternative approach to stochastic control through feedback policies.

6. DYNAMIC PROGRAMMING

Dating back to the 1950's but following a track different from that of much of the rest of optimization, is a theory of multistage decision problems under uncertainty known as 'dynamic programming'. This theory focuses on states and controls in a context of stochastic dynamics as inspired originally by engineering applications, but it makes little use of Lagrange multipliers, convexity, duality and other such problem characteristics in linear and nonlinear programming. Its centerpiece is a "principle of optimality" giving a recursive relation among cost-to-go functions. The methodology doesn't lead to numerical solutions to practical problems except in rather special cases, yet dynamic programming has many interesting things to offer nonetheless. We'll survey its foundations here with an eye toward trying later to combine some of the best features with those coming from other approaches.

Multistage stochastic control from a different viewpoint: Dynamic programming grows out of a stage-by-stage *process* of control under uncertainty that's just like the one we've already investigated. In stages $k = 0, 1, \ldots, N$ there are states $x_k \in \mathcal{X}_k$, controls $u_k \in \mathcal{U}_k$, and uncertain elements $\omega_k \in \Omega_k$, governed by dynamics $x_k = F_k(x_{k-1}, u_{k-1}, \omega_k)$ and probability distributions $\pi_k(x_{k-1}, u_{k-1})$. Costs $f_k(x_k, u_k)$ are accumulated and foreseen through their probabilistic expectations. There's a significant shift in attitude, however, in how optimization should be carried out and what it should be based on.

Parametric treatment of the initial state: There isn't just a single problem for a single initial state a. Instead, an entire family of problems is made the focus, namely one for each possible initial state x_0. The optimal values and solutions to these problems are studied in their parametric dependence on x_0.

States as sole repositories of current information: The attitude is adopted that, when stage k is reached, the state x_k contains all the information that's available about the past of the system. The choice of the control u_k can respond only to x_k. There's no memory of anything else, except to the extent that it has been built into x_k. This holds even for the realized values of $\omega_1, \ldots, \omega_k$.

Modeling implications: From one angle, this is a harmless assumption which could be seen merely as clarifying what a "state" ought to mean. It simply requires us to pay close attention to how we set up the states in a particular application. From another angle, however, this has the potential to be a serious restriction because of its effect on watershed properties like convexity.

Example of states as cumulative observations: The state could, as a special case, contain a record of all past observations. For instance, we could start with x_0 as furnishing some kind of background information and then, after ω_1 has been observed, take $x_1 = (x_0, \omega_1)$. After ω_2 has been observed we could take $x_2 = (x_0, \omega_1, \omega_2)$, and so forth. This would correspond to $\mathcal{X}_k = \mathcal{X}_0 \times \Omega_1 \times \cdots \times \Omega_k$ and

$$F_k(x_{k-1}, \omega_k) = (x_{k-1}, \omega_k) \text{ for } k = 1, \ldots, N,$$

with no controls being involved in the dynamics at all. Many variations on this theme are possible. There could be additional aspects to the state beyond such a record of observations. Or, observations could be subject to a "sunset law": we might have $x_k = (\omega_{k-2}, \omega_{k-1}, \omega_k)$ (when $k \geq 3$) to capture the idea that decisions ought not to be based on more than the three latest observations, say. Again x_k would be determined then from x_{k-1} and ω_k without intervention of u_{k-1}.

Feedback policies: Because the choice of the control u_k in stage k can't be based on any information about the past beyond what's in x_k, the concept of a policy has to be different from the one we worked with until now. We can't take for granted that the record of observations $\omega_1, \ldots, \omega_k$ is part of x_k (although it might be, as just noted). Anyway, that wouldn't fit with the desire now for parametric analysis, since fundamental policies only make sense with respect to a fixed initial state a. Instead we have to concentrate on *feedback* policies

$$u(\cdot) = \big(u_0(\cdot), u_1(\cdot), \ldots, u_N(\cdot)\big) \text{ with component functions } u_k(\cdot) : \mathcal{X}_k \to \mathcal{U}_k.$$

This means we work with $u_k(x_k)$ rather than $u_k(\omega_1, \ldots, \omega_k)$ and moreover with $u_0(x_0)$, whereas previously we had only a single choice of the initial control.

Feedback: This term refers generally to schemes in which a process of generating states from controls is supplemented by one of getting controls from states.

Evolution of the states and controls: In the execution of a feedback policy $u(\cdot)$, the dynamical system reduces to one in which x_k, as an uncertain element of $\mathcal{X}_k$ with a specific probabilistic behavior, can be known from x_{k-1}:

$$x_k = F_k(x_{k-1}, u_{k-1}(x_{k-1}), \omega_k), \text{ where}$$
$$\omega_k \text{ has distribution } \pi_k(x_{k-1}, u_{k-1}(x_{k-1})).$$

Proceeding stage by stage, starting from x_0 as a parameter, we get a conditional distribution for x_1 given x_0, then one for x_2 given x_0 and x_1, and so forth.

In this manner we build up, from the standpoint of stage 0, a distribution of $(x_1, x_2, \ldots, x_N)$ as a random variable in $\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_N$ with respect to any initial $x_0 \in \mathcal{X}_0$. Since controls are uniquely prescribed from the states, we can think of this as likewise building up a distribution of $(u_1, u_2, \ldots, u_N)$ as a random variable in $\mathcal{U}_1 \times \mathcal{U}_2 \times \cdots \times \mathcal{U}_N$.

Evolution of the underlying probabilities: The distributions for $(x_1, x_2, \ldots, x_N)$ and $(u_1, u_2, \ldots, u_N)$ correspond of course to one for $\omega = (\omega_1, \omega_2, \ldots, \omega_N)$ that emerges at the same time on the probability space Ω.

Evolution of costs: The cost terms can be viewed similarly. From the distribution generated for x_k we get a distribution for $c_k = f_k(x_k, u_k(x_k))$. The expected values of these cost terms for $k = 0, 1, \ldots, N$ therefore make sense in the light of any initial state x_0 and feedback policy $u(\cdot)$.

Technical considerations: In a general probability framework, it would be necessary to impose further conditions on the mappings $u_k(\cdot) : \mathcal{X}_k \to \mathcal{U}_k$, such as "measurability," in order to be sure that the expectations are well defined. Questions of the "integrability" of the cost expressions as functions on Ω would also need attention. By concentrating on the case of discrete probability, we are able to avoid such ramifications.

Expected cost functional: On the basis of this probabilistic evolution, we can associate with any initial state x_0 and feedback policy $u(\cdot) = (u_0(\cdot), u_1(\cdot), \ldots, u_N(\cdot))$ an expected cost value

$$\begin{aligned} J_0\big[x_0, u(\cdot)\big] &= J_0\big[x_0, u_0(\cdot), u_1(\cdot), \ldots, u_N(\cdot)\big] \\ &:= \mathbb{E}\Big\{ f_0\big(x_0, u_0(x_0)\big) + f_1\big(x_1, u_1(x_1)\big) + \cdots + f_N\big(x_N, u_N(x_N)\big) \Big\}, \end{aligned}$$

where the expectation is taken with respect to the probability distribution on the cost terms that's induced by x_0 and $u(\cdot)$, as described.

Feasibility: A feedback policy $u(\cdot)$ is deemed to be *feasible* if $J_0\big[x_0, u(\cdot)\big] < \infty$ for every $x_0 \in \mathcal{X}_0$. This necessitates having

$$\begin{aligned} u_k(x_k) \in C_k(x_k) \text{ for } k = 0, 1, \ldots, N, \text{ where} \\ C_k(x_k) := \big\{ u_k \in \mathcal{U}_k \,\big|\, f_k(x_k, u_k) < \infty \big\}, \end{aligned}$$

and it's equivalent to that property in our framework of discrete probability. Here $C_k(x_k)$ is the *feasible control set* in stage k. Obviously there can't be any policy $u(\cdot)$ unless $C_k(x_k) \neq \emptyset$ almost surely with respect to the probability

distribution on x_k that comes out of the preceding states and policy mappings $u_0(\cdot), \ldots, u_{k-1}(\cdot)$, for every initial x_0.

General problem of dynamic programming: Given a multistage stochastic control model as above, find a feedback policy $\bar{u}(\cdot) = (\bar{u}_0(\cdot), \bar{u}_1(\cdot), \ldots, \bar{u}_N(\cdot))$ such that, with respect to every other feedback policy $u(\cdot) = (u_0(\cdot), u_1(\cdot), \ldots, u_N(\cdot))$, one has

$$J_0\big(x_0, \bar{u}_0(\cdot), \ldots, \bar{u}_N(\cdot)\big) \;\leq\; J_0\big(x_0, u_0(\cdot), \ldots, u_N(\cdot)\big) \;\text{ for all } x_0.$$

Such a policy $\bar{u}(\cdot)$ is called *optimal*—in the sense of dynamic programming.

Interpretation as a form of parametric optimization: This description of the general problem underscores one of the most distinguishing features of dynamic programming. The subject isn't concerned with solving just a *single* optimization problem, as would correspond to the specification of a single initial state $x_0 = a$ as earlier, but rather with solving systematically an entire *family* of optimization problems parameterized by x_0. A optimal policy $\bar{u}(\cdot)$ of the kind being sought isn't an optimal solution to one optimization problem, but a parametric description of optimal solutions to many such problems.

Rationale: The focus on parametric optimization is dictated in part by a recursive rule for constructing optimal policies which will soon be explained. Equally important in the concept, however, is the idea of feedback itself as making sense in every stage of control. Much of the motivation derives also from interest in studying what happens to the mappings $\bar{u}_0(\cdot), \ldots, \bar{u}_k(\cdot), \ldots$ as the horizon N in the model is allowed to go toward ∞.

Connection with basic stochastic programming: Stochastic programming, as developed so far, fits closely with the case of dynamic programming in which

(a) $x_k = (x_0, \omega_1, \ldots, \omega_k)$ for $k > 0$,

(b) $F_k(x_{k-1}, u_{k-1}, \omega_k) = (x_{k-1}, \omega_k)$ for $k > 0$,

(c) $\pi_k(x_{k-1}, u_{k-1})$ is independent of u_{k-1},

(d) $f_k(x_k, u_k)$ is convex in u_k.

Feedback policies are comprised then of mappings $u_k(\cdot) : (x_0, \omega_1, \ldots, \omega_k) \to x_k$ and are almost the same as the fundamental policies we looked at previously, except that instead of a fixed initial state a there's a parameter x_0.

Any optimal policy $\bar{u}(\cdot)$ in the dynamic programming sense will, in particular in this setup, yield an optimal policy in the stochastic programming sense simply through specialization of x_0 to a. A bigger question, however, is whether the higher

level of effort required to produce an optimal policy in the dynamic programming sense is well spent if only one such specialization is called for.

Residual minimization in later stages: Parametric optimization in multistage stochastic control has so far been viewed only from stage 0, but an important virtue of the dynamic programming setup is that it exhibits essentially the same structure from any stage forward.

Let's try this out by shifting our perspective to that of stage 1. We imagine ourselves at a state $x_1 \in \mathcal{X}_1$. How we got there doesn't matter; all that counts is what happens from now on. Although mappings $u_0(\cdot) : \mathcal{X}_0 \to \mathcal{U}_0$ for stage 0 are no longer relevant, any choice of mappings $u_k(\cdot) : \mathcal{X}_k \to \mathcal{U}_k$ for $k = 1, \ldots, N$ results in a probability distribution for the cost just as before and thereby has an associated expected cost $J_1\big[x_1, u_1(\cdot), \ldots, u_N(\cdot)\big] := \mathbb{E}\big\{f_1\big(x_1, u_1(x_1)\big) + \cdots + f_N\big(x_N, u_N(x_N)\big)\big\}$. The situation is just what we faced earlier, but with one stage fewer to go until termination, and with x_1 as the parameter instead of x_0.

The same goes for subsequent stages as well. Having arrived reached stage k, we have an $N-(k-1)$-stage *residual problem* of dynamic programming, parameterized by x_k, in which the expected cost functional is

$$J_k\big[x_k, u_k(\cdot), \ldots, u_N(\cdot)\big] := \mathbb{E}\Big\{f_k\big(x_k, u_k(x_k)\big) + \cdots + f_N\big(x_N, u_N(x_N)\big)\Big\},$$

the expectation being taken relative to the probability distribution for these cost terms that's generated by $(u_k(\cdot), \ldots, u_N(\cdot))$ from x_k. We seek a residual feedback policy $(\bar{u}_k(\cdot), \ldots, \bar{u}_N(\cdot))$ such that, with respect to every other residual feedback policy $(u_k(\cdot), \ldots, u_N(\cdot))$, one has

$$J_k\big[x_k, \bar{u}_k(\cdot), \ldots, \bar{u}_N(\cdot)\big] \;\leq\; J_k\big[x_k, u_k(\cdot), \ldots, u_N(\cdot)\big] \quad \text{for all } x_k.$$

Such a feedback policy is said to be *optimal from stage k on*.

Cost-to-go functions: The optimal value in the control problem faced if, on arriving in stage k, the state is x_k, is called the (expected) *cost-to-go* associated with x_k. As a function ψ_k of $x_k \in \mathcal{X}_k$, it's given by the formula

$$\psi_k(x_k) = \inf_{u_k(\cdot), \ldots, u_N(\cdot)} J_k\big[x_k, u_k(\cdot), \ldots, u_N(\cdot)\big].$$

The key to dynamic programming is the exploration of the relationships between the successive functions $\psi_0, \psi_1, \ldots, \psi_N$. This can be carried out by analyzing the structure of the expectations that are involved.

Some rules for expectations: To proceed, we'll need ways of manipulating expectations of random variables. For convenience in stating general rules about this, we can think of the random variables as functions of $\omega \in \Omega$ with the distribution of ω fixed, although the applications we'll make will be in more diverse situations.

Several rules are obvious from the fact that an expectation is in principle an integral with respect to a probability measure. For instance, for random variables $\theta(\omega)$ and $\theta'(\omega)$ (either vector or scalar) one has

$$\mathbb{E}_\omega\big\{\theta(\omega) + \theta'(\omega)\big\} = \mathbb{E}_\omega\big\{\theta(\omega)\big\} + \mathbb{E}_\omega\big\{\theta'(\omega)\big\}, \qquad \mathbb{E}_\omega\big\{\lambda\theta(\omega)\big\} = \lambda\,\mathbb{E}_\omega\big\{\theta(\omega)\big\}.$$

As an extension of the rule for scalar multiplication, a vector inner product $v \cdot \theta(\omega)$ satisfies

$$\mathbb{E}_\omega\big\{v \cdot \theta(\omega)\big\} = v \cdot \mathbb{E}_\omega\big\{\theta(\omega)\big\}.$$

Also, because "probabilities add up to 1" it's always true for any constant c that

$$\mathbb{E}_\omega\big\{c\big\} = c, \qquad \mathbb{E}_\omega\big\{\theta(\omega) + c\big\} = \mathbb{E}_\omega\big\{\theta(\omega)\big\} + c.$$

Conditional expectations (i.e., expectations with respect to conditional probability distributions) follow the same principles but give rise to the following additional rule in case of $\omega = (\omega_1, \omega_2) \in \Omega = \Omega_1 \times \Omega_2$:

$$\mathop{\mathbb{E}}_{\omega_1,\omega_2}\big\{\theta(\omega_1, \omega_2)\big\} = \mathop{\mathbb{E}}_{\omega_1}\Big\{ \mathop{\mathbb{E}}_{\omega_2|\omega_1}\big\{\theta(\omega_1, \omega_2)\big\}\Big\}.$$

Here ω_1 is a parameter in the inner expectation, which is taken with respect to the *conditional* probability distribution for ω_2 given the value of ω_1, so that this inner expectation designates a value $\widehat{\theta}(\omega_1)$. The outer expectation is then that of $\widehat{\theta}(\omega_1)$ as a function of ω_1 with respect to the *marginal* distribution for ω_1 induced by the joint distribution for (ω_1, ω_2).

Conditional and marginal distributions: Roughly speaking, the marginal distribution of ω_1 induced by a joint distribution of (ω_1, ω_2) is the one in which the probability of ω_1 belonging to a subset A of X_1 is identified with the probability of (ω_1, ω_2) belonging to the subset $A \times \Omega_2$ of $\Omega_1 \times \Omega_2$. The representation of any joint probability distribution for (ω_1, ω_2) in terms of a marginal distribution of ω_1 along with a conditional distribution of ω_2 for each ω_1, so as to justify the rule just given, is an important part of probability theory and is fraught with technical complications of a "measurability" nature.

Example of applying the expectation rules: If $\theta(\omega_1, \omega_2) = \theta_1(\omega_1) + \theta_2(\omega_1, \omega_2)$, then

$$\mathop{I\!E}_{\omega_1,\omega_2}\big\{\theta(\omega_1,\omega_2)\big\} = \mathop{I\!E}_{\omega_1}\Big\{\theta_1(\omega_1) + \mathop{I\!E}_{\omega_2|\omega_1}\big\{\theta_2(\omega_1,\omega_2)\big\}\Big\},$$

where the outer expectation is taken with respect to the marginal distribution of ω_1 and the inner expectation with respect to the conditional distributions of ω_2 corresponding to the various values of ω_1.

Some rules for minimization: Similar rules will be needed for coping with the minimization of complicated expressions of a variable z. Obviously

$$\begin{aligned}\inf_z\big\{\lambda\varphi(z)\big\} &= \lambda\inf_z\big\{\varphi(z)\big\} \text{ for } \lambda > 0,\\ \inf_z\big\{\varphi(z) + c\big\} &= \inf_z\big\{\varphi(z)\big\} + c \text{ for any constant } c.\end{aligned}$$

In general one only has $\inf_z\big\{\varphi(z)+\varphi'(z)\big\} \geq \inf_z\big\{\varphi(z)\big\}+\inf_z\big\{\varphi'(z)\big\}$, but equality does hold for a separable function $\varphi(z_1, z_2) = \varphi(z_1) + \varphi(z_2)$:

$$\inf_{z_1,z_2}\big\{\varphi(z_1) + \varphi(z_2)\big\} = \inf_{z_1}\big\{\varphi_(z_1)\big\} + \inf_{z_2}\big\{\varphi_2(z_2)\big\},$$

and similarly for more than two arguments. On the other hand, one always has

$$\inf_{z_1,z_2}\big\{\varphi(z_1,z_2)\big\} = \inf_{z_1}\Big\{\inf_{z_2}\big\{\varphi(z_1,z_2)\big\}\Big\},$$

where the inner minimization produces a function of z_1 to which the outer minimization is then addressed.

Example of applying the minimization rules: If $\varphi(z_1, z_2) = \varphi_1(z_1) + \varphi_2(z_1, z_2)$, then

$$\inf_{z_1,z_2}\big\{\varphi_1(z_1) + \varphi_2(z_1,z_2)\big\} = \inf_{z_1}\Big\{\varphi_1(z_1) + \inf_{z_2}\big\{\varphi_2(z_1,z_2)\big\}\Big\}.$$

This is seen from the fact that, for fixed z_1, we have $\inf_{z_2}\big\{\varphi_1(z_1)+\varphi_2(z_1,z_2)\big\} = \varphi_1(z_1) + \inf_{z_2}\big\{\varphi_2(z_1,z_2)\big\}$ because the term $\varphi_1(z_1)$ merely has the role of a constant when we minimize with respect to z_2.

One-stage models: To grasp the point of dynamic programming, we have to look first at the simplest case, from which we'll learn a lesson we can apply repeatedly. What if there is only one stage to the model? Then the initial stage is also the last, and no uncertainty enters the picture. The sole cost-to-go function is ψ_0, which by definition has the formula

$$\psi_0(x_0) = \inf_{u_0(\cdot)} f_0(x_0, u_0),$$

with the minimization carried out over all mappings $u_0(\cdot) : \mathcal{X}_0 \to \mathcal{U}_0$. It's obvious in this elementary situation the mapping $u_0(\cdot)$ affects ψ_0 at a particular point x_0 only through its value $u_0(x_0)$ at that point. Equally well we have the formula

$$\psi_0(x_0) = \inf_{u_0} f_0(x_0, u_0),$$

in which $\psi_0(x_0)$ appears as the optimal value in a minimization problem over $\mathcal{U}_0$ instead of over a space of mappings. The corresponding optimal solution set

$$\Psi_0(x_0) = \operatorname*{argmin}_{u_0} f_0(x_0, u_0)$$

also plays a role, because dynamic programming requires us to look for a mapping $\bar{u}_0(\cdot) : \mathcal{X}_0 \to \mathcal{U}_0$ such that, with respect to every other $u_0(\cdot) : \mathcal{X}_0 \to \mathcal{U}_0$, one has

$$f_0\big(x_0, \bar{u}_0(x_0)\big) \leq f_0\big(x_0, u_0(x_0)\big) \quad \text{for all } x_0.$$

Clearly $\bar{u}_0(\cdot)$ enjoys this property if and only if

$$\bar{u}_0(x_0) \in \Psi_0(x_0) \text{ for all } x_0.$$

In other words, an optimal policy in the dynamic programming sense can be put together by selecting for each $x_0 \in \mathcal{X}_0$ some element of $\Psi_0(x_0)$ as the value of $\bar{u}_0(\cdot)$.

Note: For this to be possible the set $\Psi_0(x_0)$ must of course be nonempty for each $x_0 \in \mathcal{X}_0$. Conditions on f_0 and the spaces $\mathcal{X}_0$ and $\mathcal{U}_0$ might have to be imposed to make sure that's the case, but regardless, the function ψ_0 is well defined (as long as infinite values are admitted).

Two-stage models: Continuing with this scrutiny of special cases, suppose that $N = 1$, so that we have cost-to-go functions ψ_0 and ψ_1. These are defined by

$$\begin{aligned}
\psi_0(x_0) &= \inf_{u_0(\cdot), u_1(\cdot)} \Big\{ \mathop{I\!E}_{\omega_1 | x_0, u_0(x_0)} \Big\{ f_0\big(x_0, u_0(x_0)\big) + f_1\big(x_1, u_1(x_1)\big) \Big\} \Big\}, \\
\psi_1(x_1) &= \inf_{u_1(\cdot)} f_1\big(x_1, u_1(x_1)\big),
\end{aligned}$$

with the minimization taken over mappings $u_0(\cdot) : \mathcal{X}_0 \to \mathcal{U}_0$ and $u_1(\cdot) : \mathcal{X}_1 \to \mathcal{U}_1$. The stage 1 formula has x_1 standing merely for an element of $\mathcal{X}_1$, but the stage 0 formula has x_1 standing for $F_1\big(x_0, u_0(x_0), \omega_1\big)$, i.e., for a random variable based on x_0, $u_0(\cdot)$, and the probability distribution $\pi_1\big(x_0, u_0(x_0)\big)$ for ω_1. (We could write $x_1(\omega_1)$ in this case but in the interest of notational simplicity have chosen simply to

write x_1 again. The notation "$\omega_1 \mid x_0, u_0(x_0)$" under the expectation sign emphasizes that the expected value generally depends on x_0 and $u_0(x_0)$ through the dependence of the distribution $\pi_1\big(x_0, u_0(x_0)\big)$ on these elements.)

The stage 1 formula refers to a residual one-stage problem. Our analysis of such problems tells us it can be written equally well as

$$\psi_1(x_1) = \inf_{u_1} f_1(x_1, u_1).$$

The stage 0 formula, on the other hand, can be manipulated by the listed rules of expectation and minimization to obtain

$$\begin{aligned}
\psi_0(x_0) &= \inf_{u_0(\cdot), u_1(\cdot)} \Big\{ f_0\big(x_0, u_0(x_0)\big) + \mathop{I\!E}_{\omega_1 | x_0, u_0(x_0)} \Big\{ f_1\big(x_1, u_1(x_1)\big) \Big\} \Big\} \\
&= \inf_{u_0(\cdot)} \Big\{ f_0\big(x_0, u_0(x_0)\big) + \inf_{u_1(\cdot)} \Big\{ \mathop{I\!E}_{\omega_1 | x_0, u_0(x_0)} \Big\{ f_1\big(x_1, u_1(x_1)\big) \Big\} \Big\} \Big\} \\
&= \inf_{u_0(\cdot)} \Big\{ f_0\big(x_0, u_0(x_0)\big) + \mathop{I\!E}_{\omega_1 | x_0, u_0(x_0)} \Big\{ \inf_{u_1(\cdot)} f_1\big(x_1, u_1(x_1)\big) \Big\} \Big\} \\
&= \inf_{u_0(\cdot)} \Big\{ f_0\big(x_0, u_0(x_0)\big) + \mathop{I\!E}_{\omega_1 | x_0, u_0(x_0)} \Big\{ \psi_1\big(x_1\big) \Big\} \Big\} \\
&= \inf_{u_0(\cdot)} \Big\{ f_0\big(x_0, u_0(x_0)\big) + \mathop{I\!E}_{\omega_1 | x_0, u_0(x_0)} \Big\{ \psi_1\big(F_1(x_0, u_0(x_0), \omega_1)\big) \Big\} \Big\}.
\end{aligned}$$

This expression can be simplified conceptually by introducing the function

$$\widehat{\psi}_0(x_0, u_0) := \mathop{I\!E}_{\omega_1 | x_0, u_0} \Big\{ \psi_1\big(F_1(x_0, u_0, \omega_1)\big) \Big\},$$

where the distribution of ω_1 is given by $\pi_1(x_0, u_0)$. It comes out as

$$\psi_0(x_0) = \inf_{u_0(\cdot)} \Big\{ f_0\big(x_0, u_0(x_0)\big) + \widehat{\psi}_0\big(x_0, u_0(x_0)\big) \Big\}$$

and therefore through our one-stage analysis as

$$\psi_0(x_0) = \inf_{u_0} \Big\{ f_0(x_0, u_0) + \widehat{\psi}_0(x_0, u_0) \Big\},$$

where the minimization for each x_0 is carried out over $\mathcal{U}_0$ instead of over mappings from $\mathcal{X}_0$ to $\mathcal{U}_0$. The one-stage analysis further reveals now how to get an optimal policy $\big(\bar{u}_0(\cdot), \bar{u}_1(\cdot)\big)$: just take

$$\begin{aligned}
\bar{u}_0(x_0) \in \Psi_0(x_0) &= \operatorname*{argmin}_{u_0} \Big\{ f_0(x_0, u_0) + \widehat{\psi}_0(x_0, u_0) \Big\} \text{ for each } x_0, \\
\bar{u}_1(x_1) \in \Psi_1(x_1) &= \operatorname*{argmin}_{u_1} f_1(x_1, u_1) \text{ for each } x_1.
\end{aligned}$$

Summary: The picture emerges that we could determine ψ_1 and $\bar{u}_1(\cdot)$ by minimizing $f_1(x_1, u_1)$ in u_1 for each x_1, and after that, by forming $\widehat{\psi}_0(x_0, u_0)$, we could determine ψ_0 and $\bar{u}_0(\cdot)$ by minimizing $f_0(x_0, u_0) + \widehat{\psi}_0(x_0, u_0)$ in u_0 for each x_0. That would produce an optimal policy $(\bar{u}_0(\cdot), \bar{u}_1(\cdot))$.

Principle of optimality in dynamic programming: *In the general multistage framework, the cost-to-go functions ψ_k for $k < N$ satisfy the recursive relations*

$$\psi_k(x_k) = \inf_{u_k} \Big\{ f_k(x_k, u_k) + \widehat{\psi}_k(x_k, u_k) \Big\} \text{ where}$$
$$\widehat{\psi}_k(x_k, u_k) := \mathop{\mathbb{E}}_{\omega_{k+1}|x_k,u_k} \Big\{ \psi_{k+1}\big(F_{k+1}(x_k, u_k, \omega_{k+1})\big) \Big\},$$

from which they can be generated backward from stage to stage, starting with

$$\psi_N(x_N) = \inf_{u_N} f_N(x_N, u_N).$$

In terms of the corresponding argmin sets $\Psi_k(x_k)$ for $k = 0, 1, \ldots, N$, an optimal policy $\bar{u}(\cdot) = \big(\bar{u}_0(\cdot), \bar{u}_1(\cdot), \ldots, \bar{u}_N(\cdot)\big)$ can be obtained from this procedure by taking

$$\bar{u}_k(x_k) \in \Psi_k(x_k) \text{ for each } x_0.$$

Proof. Our preceding analysis of one-stage models directly supports the assertions about ψ_N and $\bar{u}_N(\cdot)$. For any $k < N$, we can argue now much as we did for two-stage models, although the notation is somewhat more complicated. By the definition of the cost-to-go functions, we have

$$\psi_k(x_k) = \inf_{u_k(\cdot), u_{k+1}(\cdot), \ldots, u_N(\cdot)} \Big\{ \mathop{\mathbb{E}}_{\omega_k, \omega_{k+1}, \ldots, \omega_N} \Big\{ f_k\big(x_k, u_k(x_k)\big) + f_{k+1}\big(x_{k+1}, u_{k+1}(x_{k+1})\big) + \cdots + f_N\big(x_N, u_N(x_N)\big) \Big\} \Big\},$$
$$\psi_{k+1}(x_{k+1}) = \inf_{u_{k+1}(\cdot), \ldots, u_N(\cdot)} \Big\{ \mathop{\mathbb{E}}_{\omega_{k+1}, \ldots, \omega_N} \Big\{ f_{k+1}\big(x_{k+1}, u_{k+1}(x_{k+1})\big) + \cdots + f_N\big(x_N, u_N(x_N)\big) \Big\} \Big\},$$

where in the first formula $x_{k+1}, x_{k+2}, \ldots, x_N$ stand for random variables generated by the dynamics starting from x_k, but in the second formula only $x_{k+2}, \ldots, x_N$ have this character, and they are generated by the dynamics starting from x_{k+1}

as a parameter ranging over $\mathcal{X}_{k+1}$. We can apply the rules of expectation and minimization to the first formula to rewrite it as

$$\begin{aligned}
\psi_k(x_k) &= \inf_{u_k(\cdot)} \Big\{ \mathop{\mathbb{E}}_{\omega_k} \Big\{ f_k\big(x_k, u_k(x_k)\big) + \\
&\qquad \inf_{u_{k+1}(\cdot),\dots,u_N(\cdot)} \Big\{ \mathop{\mathbb{E}}_{\omega_{k+1},\dots,\omega_N|\omega_k} \Big\{ f_{k+1}\big(x_{k+1}, u_{k+1}(x_{k+1})\big) \\
&\qquad\qquad + \cdots + f_N\big(x_N, u_N(x_N)\big) \Big\} \Big\} \Big\} \Big\} \\
&= \inf_{u_k(\cdot)} \Big\{ \mathop{\mathbb{E}}_{\omega_k} \Big\{ f_k\big(x_k, u_k(x_k)\big) + \psi_{k+1}(x_{k+1}) \Big\} \Big\} \\
&= \inf_{u_k(\cdot)} \Big\{ \mathop{\mathbb{E}}_{\omega_k} \Big\{ f_k\big(x_k, u_k(x_k)\big) + \psi_{k+1}\big(F_{k+1}(x_k, u_k(x_k), \omega_{k+1})\big) \Big\} \Big\}.
\end{aligned}$$

We recognize that the last version gives the optimal value in a two-stage model as already analyzed, and that it therefore yields the claimed expression of $\psi_k(x_k)$ as the infimum of $f_k(x_k, u_k) + \widehat{\psi}_k(x_k, u_k)$ with respect to u_k. At the same time we deduce that $\bar{u}_k(\cdot)$ is optimal if $\bar{u}_k(x_k)$ is taken for each x_k to be an element of the associated minimizing set. □

Dynamic programming algorithm: These recursive relations are the foundation for a 'algorithm' which, at least in concept, might be used to solve the general problem in dynamic programming. First we construct the function ψ_N, obtaining at the same time a mapping $\bar{u}_N(\cdot) : \mathcal{X}_N \to \mathcal{U}_N$. Next we insert this in the recursive relation in order to construct the function ψ_{N-1} and a mapping $\bar{u}_{N-1}(\cdot) : \mathcal{X}_{N-1} \to \mathcal{U}_{N-1}$. Continuing from ψ_{N-1} we get ψ_{N-2} and so forth down to ψ_0, generating additional mappings $\bar{u}_{N-2}(\cdot), \dots, \bar{u}_0(\cdot)$ along the way.

In the end we not only have all the cost-to-go functions for the problem but a particular optimal feedback policy $\bar{u}(\cdot)$ formed by $\bar{u}_0(\cdot), \bar{u}_1(\cdot), \dots, \bar{u}_N(\cdot)$ (as long as the minimizing sets that are encountered are all nonempty).

Technical troubles: To apply the recursive formula for ψ_k in terms of ψ_{k+1}, it's necessary to solve *for each* $x_k \in \mathcal{X}_k$ an optimization problem in which the expression

$$f_k(x_k, u_k) + \mathop{\mathbb{E}}_{\omega_{k+1}|x_k,u_k} \Big\{ \psi_{k+1}\big(F_k(x_k, u_k, \omega_{k+1})\big) \Big\}$$

is minimized over all $u_k \in \mathcal{U}_k$. What might this involve? This problem could in some cases be discrete, with $\mathcal{U}_k$ a finite set and therefore only finitely many choices of u_k to compare. Then, *as long as the number of choices isn't too large*, the minimization can be carried out by "brute force."

Let's think instead, though, about cases involving continuous variables, say with $\mathcal{U}_k = I\!R^{n_k}$. In our setup we know the function f_k in stage k, and with it the implicit control set $C_k(x_k)$. For the sake of concreteness, let

$$f_k(x_k, u_k) = \begin{cases} g_k(x_k, u_k) & \text{if } u_k \in U_k,\ G_k(x_k, u_k) \leq 0, \\ \infty & \text{otherwise} \end{cases}$$

for a finite function g_k, so that $C_k(x_k)$ is given by the constraints $u_k \in U_k$ and $G_k(x_k, u_k) \leq 0$. Under the condition that $\widehat{\varphi}_k(x_k, u_k) < \infty$ whenever $u_k \in C_k(x_k)$, so that there aren't any implicit constraints induced from the future, the constraint structure faced in this subproblem can be imagined as falling into the conventional patterns in finite-dimensional optimization and not posing any unusual difficulties: we have to minimize the finite expression $g_k(x_k, u_k) + \widehat{\psi}_k(x_k, u_k)$ over all $u_k \in U_k$ with $G_k(x_k, u_k) \leq 0$.

In contrast to $g_k(x_k, u_k)$, $G_k(x_k, u_k)$ and U_k, however, the values of $\widehat{\psi}_k(x_k, u_k)$ aren't known directly but have to be developed from a formula dependent on ψ_{k+1}, which itself had to be constructed from other data. This raises questions about how the minimization can be carried out, since some techniques might demand for instance that we know first or second partial derivatives of $\widehat{\psi}_k(x_k, u_k)$ with respect to the components of u_k. Will such derivatives exist? Indeed, is there even any assurance that $\widehat{\psi}_k(x_k, u_k)$ will have continuity properties with respect to u_k of the kind taken for granted in elementary optimization? We already know from our look at stochastic programming that the answer can well be negative, but that convexity might come to the rescue if it can be counted on.

Troubles with the expectation: From the standpoint of mathematical rigor, we would need to ensure in the first place that the expression being integrated with respect to the probability distribution $\pi_{k+1}(x_k, u_k)$ of ω_{k+1} makes sense ("measurability" etc.), yet this is beset with the issue of what we can say about the function ψ_{k+1}. Clearly, a sound theory demands that facts be carefully developed about the extent to which crucial properties of the cost-to-go functions can be propagated backward from stage N in the presence of nice properties of the data elements f_k, F_k, and the constraints.

Troubles with global minimization: We can't truly get $\psi_k(x_k)$ without performing the minimization *globally*. Apart from "brute force" cases, that may be very hard to do. Convexity of $f_k(x_k, u_k)$ with respect to u_k could help a lot, but when can we hope for that to be available? Again the difficulty lies in the circumstance that $\widehat{\psi}_k$ depends in general not only on F_k but also on the particular probability distribution $\pi_{k+1}(x_k, u_k)$ for ω_{k+1}.

Practical considerations: In a specific problem of multistage optimal control with a fixed initial state a, the real goal is to determine the initial control element $\bar{u}_0$ that should be utilized. Everything else is subsidiary and designed as a way of achieving an adequate *present* representation of the *future* (with its various recourse opportunities). The dynamic programming approach can serve to find the desired element $\bar{u}_0$, but it requires the construction of whole functions ψ_k on the spaces $\mathcal{X}_k$, at least for $k = 1, \ldots, N$. This is a formidable undertaking, as already can be appreciated just by raising the question of how a function ψ_k might be represented. The following are some cases to think about.

Functions given by tables of values: Lacking any other handle on the situation, we can imagine representing and storing the functions simply through extensive lists of their values. After all, many functions with only an empirical basis are stored like this in computers anyway. But there are sharp limitations to such an approach. Usually it's employed for functions of one or two variables over a modest range like an interval or box. If the state x_k is a vector having 10, or 100, or 1000 "continuous" components, the idea is hopeless. Furthermore it suffers from the disadvantage that function properties of potentially great importance, like convexity for instance, don't readily come to the surface.

Functions given by specifying parameters: Much of mathematics revolves around special classes of functions: linear, quadratic, polynomial, trigonometric, etc. To select a function from such a class, it's only necessary to specify the values of certain coefficients. We might hope to take advantage of this by concentrating on cases of dynamic programming where the cost-to-go functions ψ_k stay within a particular class as we propagate them backward. The procedure could then be reduced in effect to one in which formulas are derived for propagating the needed coefficient values backward. Unfortunately, although there are interesting and important examples of this, the idea is severely restricted in its applicability. The operation of minimization, especially constrained minimization, just doesn't preserve classes of functions like those that have served so well traditionally.

Functions given by approximation: If the true functions ψ_k are hard to come by, perhaps we can develop and maintain approximations of them which at least will support *approximate* methods of solving stochastic multistage control problems. This is a great idea—and an intriguing challenge for mathematical research. Once more a break with tradition is demanded because approximation by polynomials (e.g. through "Taylor expansions") or trigonometric functions (e.g. through

"Fourier expansions") miss the boat. Even if it were possible to calculate such approximations, they couldn't be propagated through the formula for ψ_k in terms of ψ_{k+1}. Maybe, though, other kinds of approximation (e.g. ones relying on convexity and duality) might be discovered that could work effectively.

Calculation of expectation values: Just because expectations are integrals, and integration is a familiar, classical operation, that doesn't mean that expected values are easy to calculate. Even when the function being integrated is expressed by an analytic formula, an integral in three or more dimensions can be hard to compute. Roughly speaking, beyond dimension 10 numerical integration is virtually impossible, and difficulties can set in much earlier than that.

Schemes based on multidimensional grids, for instance, fail because the number of points needed in a grid in order to ensure a reasonable degree of accuracy grows exponentially. If 1000 points along the real line are needed to compute an integral well with respect to a single variable, than with 10 integration variables there could be the need for $(1000)^{10} = 10^{30}$ points making up a grid in $\mathbb{R}^{10}$. And this is just for computing the value of a single expectation, whereas what we actually have to aim at, in the case of $\widehat{\psi}_k(x_k, u_k) = \mathbb{E}_{\omega_{k+1}}\left\{\psi_{k+1}\big(F(x_k, u_k, \omega_{k+1})\big)\right\}$, is such a value as parameterized by x_k and u_k!

Simulation and sampling: One idea that has attracted a lot of attention for such reasons is the calculation of expectations approximately through sampling. Suppose for simplicity that the distribution of ω_k doesn't really depend on x_k or u_k. Instead of trying to calculate the expectation of $\psi_{k+1}\big(F_k(x_k, u_k, \omega_{k+1}\big)$ by numerical integration in ω_{k+1} for each x_k and u_k (with respect to the distribution $\pi_{k+1}(x_k, u_k)$), we can think of using random number generators and simulation in a scheme of producing a sample set of values $\omega_{k+1}^1, \omega_{k+1}^2, \ldots, \omega_{k+1}^s$ that mimics what might happen if we made s independent observations of ω_{k+1}. We could then, as a sort of approximation, replace the true expected value $\widehat{\psi}_k(x_k, u_k)$ by the simulated "empirical" expected value

$$\widehat{\psi}_k^\star(x_k, u_k) = \frac{1}{s}\sum\nolimits_{q=1}^{s} \psi_{k+1}\big(F_k(x_k, u_k, \omega_{k+1}^q).$$

Note that there's no problem with maintaining dependence on x_k and u_k.

Interesting mathematical questions then come up. In what quantifiable sense does $\widehat{\psi}_k^\star$ approximate $\widehat{\psi}_k$, and how does the number s of sample points affect the closeness of approximation? If simulation is used in each stage, the given control problem is replaced by one in which all the probability distributions are

discrete and finite, and when this altered problem is solved, an optimal initial $\bar{u}_0^\star$ is obtained which is likely to differ from an initial control $\bar{u}_0$ that would correspond to solving the given problem, if that were possible. Can we be sure that as the sample sizes in the simulation are increased, $\bar{u}_0^\star$ will converge to $\bar{u}_0$, or something like that? How big a sample might be needed in order for us to be confident about our closeness of approximation? These are subjects of ongoing research for which newly developed techniques of variational analysis are essential.

Example of linear-quadratic regulation: A beautiful illustration of dynamic programming at its best is seen in applications to stabilizing a system around a desired state, such as a state of equilibrium. Let's think of a system with states $x_k \in \mathbb{R}^d$ (identified as the state space $\mathcal{X}_k$ for all k) which, in the absence of any controls or disturbances, would be governed by

$$x_k = Ax_{k-1}$$

for a certain matrix $A \in \mathbb{R}^{d\times d}$. The origin is a stationary point for such dynamics: if $x_{k-1} = 0$, then $x_k = 0$. This could be a stable rest point, in the sense that the trajectory $x_0, x_1, \ldots, x_k, \ldots$ generated from any initial state x_0 exhibits convergence to 0. (Certain conditions on the eigenvalues of A would imply that.) The origin could be then be interpreted as representing a state of stable equilibrium. Whether this is the case or not, let's imagine that the origin represents the state we wish the system to reach and maintain.

Deterministic control: Our wishes can only be exercised through the possibility of control, however. Let's augment the dynamics now to

$$x_k = Ax_{k_1} + Bu_{k-1}$$

for a control vectors u_k in $\mathbb{R}^n$ (identified as the control space $\mathcal{U}_k$ for all k) and a certain matrix $B \in \mathbb{R}^{d\times n}$. We can then imagine that if the system is initially in a state $x_0 \neq 0$ we can perhaps steer it there by selecting a sequence of vectors $u_0, u_1, \ldots, u_k, \ldots, u_{N-1}$ with the property that the corresponding trajectory $x_0, x_1, \ldots, x_k, \ldots, x_{N-1}, x_N$ generated from the augmented dynamics has $x_N = 0$. Whether this is possible or not, regardless of the initial state x_0, depends on the matrices A and B and is an issue studied in control theory. (There's an algebraic answer which we don't need to go into here.) A less demanding criterion would be the possibility that over an *unlimited* number of time periods one could choose a control sequence such that the corresponding state sequence *converges* to 0.

Supposing some degree of control is possible, we can ask how it might be exercised optimally. It's standard to consider minimizing a sum of expressions

$$\tfrac{1}{2}x_k \cdot Qx_k + \tfrac{1}{2}u_k \cdot Ru_k =: f_k(x_k, u_k)$$

in which Q and R are symmetric matrices in $\mathbb{R}^{d\times d}$ and $\mathbb{R}^{n\times n}$, with Q positive semidefinite and R positive definite. The quadratic term in x_k gives a penalty for deviations of x_k from the desired state 0, while the quadratic term in u_k measures the "energy" expended in executing u_k. (Doing nothing, i.e., taking $u_k = 0$, costs nothing; energy is modeled as growing quadratically with the magnitude of the control effort.)

Stochastic control: Consider now the situation in which the dynamics are really

$$x_k = Ax_{k-1} + Bu_{k-1} + \omega_k = F_k(x_{k-1}, u_{k-1}, \omega_k),$$

with ω_k representing a *disturbance*—a random vector in $\mathbb{R}^d$ having expectation $E\{\omega_k\} = 0$, the distribution of ω_k being the same for every k and independent of x_k and u_k; we'll denote it simply by π. Even if the system started at $x_0 = 0$ and we left it alone (taking $u_k = 0$ for all k), the state x_k could wander farther and farther from 0 as time passes. On the other hand, no matter how much control we might try to apply, the repeated disturbances could keep us from being able to keep the state at 0. But we could still try to choose the controls so as to minimize a sum of quadratic expressions $f_k(x_k, u_k)$ of the type described.

Dynamic programming formulation: We are led to seek, for an arbitrary but fixed horizon N, a policy $\bar{u}(\cdot) = \big(\bar{u}_0(\cdot), u_1(\cdot), \ldots, \bar{u}_N(\cdot)\big)$ that, regardless of the initial state x_0, minimizes the expression

$$\mathop{\mathbb{E}}_{\omega_0,\ldots,\omega_N}\Big\{\sum\nolimits_{k=0}^{N}\big[\tfrac{1}{2}x_k \cdot Qx_k + \tfrac{1}{2}u_k(x_k) \cdot Ru_k(x_k)\big]\Big\}$$

in which $x_1, \ldots, x_N$ stand for the random variables generated from x_0 and $u(\cdot)$ by $x_k = Ax_{k-1} + Bu_{k-1}(x_{k-1}) + \omega_k$. (The distribution of x_k depends then on that of $(\omega_1, \ldots, \omega_k)$, which is just the product of k copies of the distribution π.)

Application of the dynamic programming algorithm: We start by determining the final cost-to-go function

$$\psi_N(x_N) = \inf_{u_N \in \boldsymbol{R}^n}\Big\{\tfrac{1}{2}x_N \cdot Qx_N + \tfrac{1}{2}u_N \cdot Ru_N\Big\}.$$

Trivially this definition just yields

$$\psi_N(x_N) = \tfrac{1}{2} x_N \cdot Q x_N, \qquad \bar{u}_N(x_N) \equiv 0.$$

Next we can invoke the formula for ψ_{N-1} in terms of ψ_N in the dynamic programming principle of optimality, employing this simple expression for ψ_N. In working out all the details, we find that ψ_{N-1} is again a quadratic function with a positive semidefinite matrix, but this time with a constant term added. Further iterations result in cost-to-go functions with this property as well.

To avoid excess algebraic tedium in the verification of this fact, let's establish it by a recursive argument, which will also have the advantage of leading to an algorithm for generating the matrices and constants that are involved. *The claim is that*

$$\psi_k(x_k) = \tfrac{1}{2} x_k \cdot K_k x_k + \gamma_k \text{ for } k = 0, 1, \ldots, N$$

for certain symmetric, positive semidefinite matrices $K_k \in \mathbb{R}^{d \times d}$ and constants γ_k, and hence that ψ_k is convex.

This has already been seen to be true for ψ_N with $K_N = Q$ and $\gamma_N = 0$. Calculating backwards, we suppose now that it's true for ψ_{k+1}. From the formula for ψ_k in terms of ψ_{k+1} we get

$$\begin{aligned}
\psi_k(x_k) &= \inf_{u_k \in \boldsymbol{R}^n} \Big\{ \tfrac{1}{2} x_k \cdot Q x_k + \tfrac{1}{2} u_k \cdot R u_k + \mathop{\mathbb{E}}_{\omega_{k+1}} \psi_{k+1}\big(Ax_k + Bu_k + \omega_{k+1}\big) \Big\} \\
&= \tfrac{1}{2} x_k \cdot Q x_k + \inf_{u_k \in \boldsymbol{R}^n} \Big\{ \tfrac{1}{2} u_k \cdot R u_k + \mathop{\mathbb{E}}_{\omega_{k+1}} \Big\{ \psi_{k+1}\big(Ax_k + Bu_k + \omega_{k+1}\big) \Big\} \Big\}
\end{aligned}$$

in which we have

$$\begin{aligned}
&\psi_{k+1}\big(Ax_k + Bu_k + \omega_{k+1}\big) \\
&\qquad = \tfrac{1}{2} [Ax_k + Bu_k + \omega_{k+1}] \cdot K_{k+1} [Ax_k + Bu_k + \omega_{k+1}] + \gamma_{k+1} \\
&\qquad = \tfrac{1}{2} [Ax_k + Bu_k] \cdot K_{k+1} [Ax_k + Bu_k] \\
&\qquad\quad + \omega_{k+1} \cdot K_{k+1} [Ax_k + Bu_k] + \tfrac{1}{2} \omega_{k+1} \cdot K_{k+1} \omega_{k+1} + \gamma_{k+1}.
\end{aligned}$$

In taking the expectation of this with respect to ω_{k+1} only the terms containing ω_{k+1} are affected, and moreover we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\omega_{k+1}} \Big\{ \omega_{k+1} \cdot K_{k+1} [Ax_k + Bu_k + \omega_{k+1}] \Big\} \\
&\qquad\qquad = E\{\omega_{k+1}\} \cdot K_{k+1} [Ax_k + Bu_k + \omega_{k+1}] = 0
\end{aligned}$$

because $E\{\omega_{k+1}\} = 0$. Adding the expectation of $\frac{1}{2}\omega_{k+1} \cdot K_{k+1}\omega_{k+1}$ to the constant γ_{k+1} to get a new constant γ_k, we obtain

$$\psi_k(x_k) = \tfrac{1}{2}x_k \cdot Qx_k + \gamma_k + \inf_{u_k \in \boldsymbol{R}^n} \varphi_k(x_k, u_k)$$

for the function φ_k defined by

$$\begin{aligned}\varphi_k(x_k, u_k) &:= \tfrac{1}{2}u_k \cdot Ru_k + \tfrac{1}{2}[Ax_k + Bu_k] \cdot K_{k+1}[Ax_k + Bu_k] \\ &= \tfrac{1}{2}x_k \cdot A^*K_{k+1}Ax_k + u_k \cdot B^*K_{k+1}Ax_k + \tfrac{1}{2}u_k \cdot [R + B^*K_{k+1}B]u_k\end{aligned}$$

(with * denoting transpose). Because K_{k+1} is positive semidefinite and R is positive definite, we have $\varphi_k(x_k, u_k)$ convex jointly in x_k and u_k and strictly convex in u_k, with the matrix $R + B^*K_{k+1}B$ being positive definite and in particular nonsingular. The convexity of φ_k ensures that $\inf_{u_k} \varphi_k(x_k, u_k)$ is convex in x_k and therefore by the positive semidefiniteness of Q that $\psi_k(x_k)$ is convex in x_k. The strict convexity in u_k enables us to carry out the minimization by taking the gradient of $\varphi_k(x_k, u_k)$ in u_k and setting it equal to 0. We get the equation

$$B^*K_{k+1}Ax_k + (R + B^*K_{k+1}B)u_k = 0,$$

which solves out to $u_k = -(R + B^*K_{k+1}B)^{-1}B^*K_{k+1}Ax_k$. Substituting back to see what the minimum value is, we come up with

$$\begin{aligned}\psi_k(x_k) &= \tfrac{1}{2}x_k \cdot (Q + A^*K_{k+1}A)x_k \\ &\quad - \tfrac{1}{2}(B^*K_{k+1}Ax_k) \cdot (R + B^*K_{k+1}B)^{-1}(B^*K_{k+1}Ax_k) + \gamma_k \\ &= \tfrac{1}{2}x_k \cdot \Big[(Q + A^*K_{k+1}A) - A^*K_{k+1}B(R + B^*K_{k+1}B)^{-1}B^*K_{k+1}A\Big]x_k + \gamma_k.\end{aligned}$$

Seems complicated? This is indeed a quadratic form x_k plus a constant, in which the matrix is symmetric and moreover positive semidefinite, due to ψ_k being convex. Thus, we have verified that ψ_k inherits from ψ_{k+1} the pattern claimed.

Solution: The methodology has revealed that $\psi_k(x_k) = \frac{1}{2}x_k \cdot K_k x_k + \gamma_k$ for the symmetric, positive definite matrices K_k and constants $\gamma_k \geq 0$ given recursively by

$$\begin{cases} K_k &= (Q + A^*K_{k+1}A) - A^*K_{k+1}B(R + B^*K_{k+1}B)^{-1}B^*K_{k+1}A, \\ \gamma_k &= \gamma_{k+1} + E\Big\{\frac{1}{2}\omega_{k+1} \cdot K_{k+1}\omega_{k+1}\Big\} \end{cases}$$

for $k = N - 1, \ldots\ldots, 0$, starting from

$$K_N = Q, \qquad \gamma_N = 0,$$

and that an optimal policy for stage k (any $k < N$) is given by the linear rule

$$\bar{u}_k(x_k) = L_k x_k, \qquad L_k = -(R + B^* K_{k+1} B)^{-1} B^* K_{k+1} A.$$

We wouldn't like to perform the indicated matrix computations ourselves by hand, but for a computer there's nothing to them, as long as they are numerical stable.

Dynamic programming as a source of control laws: We've been looking at dynamic programming as a possible approach to solving multistage control problems under uncertainty when an initial state a is given, but that doesn't do full justice to the subject. It's real strength emerges in circumstances in which policies take on the aura of "laws" about how the control invoked should be dictated by the current state. The results in the example of linear-quadratic regulation have such a quality, revealing in particular that "linear control laws" suffice for that application.

Models with repetitive circumstances: Dynamic programming is especially of interest in circumstances in which our attention is concentrated on the behavior as $N \to \infty$ when the data elements in the problem are the same at every stage:

$$\mathcal{X}_k = \mathcal{X}, \quad \mathcal{U}_k = \mathcal{U}, \quad f_k = f, \quad F_k = F, \quad \pi_k = \pi.$$

In such models the imposition of a finite horizon N seems particularly artificial, and there's motivation for trying to understand the "limiting case" that might arise as the horizon recedes farther and farther into the future.

Asymptotic considerations: Let's indicate the dependence of the cost-to-go functions on N by marking them with superscript N. Thus, we have $\psi_0^N, \psi_1^N, \ldots, \psi_N^N$, all on the space $\mathcal{X}$, given recursively by

$$\begin{aligned}
\psi_N^N(x) &= \inf_{u \in \mathcal{U}} f(x,u), \\
\psi_k^N(x) &= \inf_{u \in \mathcal{U}} \Big\{ f(x,u) + \mathbb{E}_\omega \Big\{ \psi_{k+1}^N\big(F(x,u,\omega)\big) \Big\} \Big\} \text{ for } k = N-1, \ldots, 1, 0,
\end{aligned}$$

where the subscripts have been dropped from x, u and ω because they are now superfluous; the expectations are taken always with respect to the same distribution π. Along with this we have optimal policy functions given by

$$\bar{u}_k^N(x) \in \Psi_k^N(x)$$

for the corresponding argmin sets. The question is what happens to these cost-to-go functions and policy functions as $N \to \infty$. One may conjecture that, as

the horizon recedes into the indefinite future, the cost-to-go picture from stage 1 should more and more resemble the cost-to-go picture from stage 0. Heuristically on this basis one is led to the following "steady-state" equation for an unknown cost-to-go function ψ:

$$\psi(x) = \inf_{u \in \mathcal{U}} \Big\{ f(x,u) + \mathbb{E}_\omega \Big\{ \psi\big(F(x,u,\omega)\big) \Big\} \Big\},$$

yielding the policy function

$$\bar{u}(x) \in \operatorname*{argmin}_{u \in \mathcal{U}} \Big\{ f(x,u) + \mathbb{E}_\omega \Big\{ \psi\big(F(x,u,\omega)\big) \Big\} \Big\}.$$

There are some technical difficulties with taking the conjectured equation for ψ at face value, because of convergence issues. Nevertheless, the asymptotic concept is important, and a major branch of dynamic programming theory is devoted to justifying it in particular cases. The benefit is a single control "law" $x \mapsto \bar{u}(x)$ for generating the control to be applied when in state x, no matter what stage the process might be in.

Linear-quadratic regulation revisited: These asymptotic considerations in the special context of the linear-quadratic regulator problem lead to the investigation of the existence of a symmetric, positive definite matrix K such that

$$K \;=\; (Q + A^*KA) - A^*KB(R + B^*KB)^{-1}B^*KA,$$

which would be associated with a steady-state linear control law

$$\bar{u}(x) = Lx, \qquad L = -(R + B^*KB)^{-1}B^*KA.$$

The special relation that K is required to satisfy is called a discrete-time *Riccati equation*. Such equations have been closely studied for the sake of this very application. It's known that under our assumptions about A, B, Q and R there is indeed a unique symmetric, positive definite matrix K providing a solution. Moreover, if the matrices K_k and L_k previously generated are denoted by K_k^N and L_k^N to indicate their dependence on the horizon N, one has

$$\lim_{N\to\infty} K_k^N \;=\; K \text{ and } \lim_{N\to\infty} L_k^N \;=\; L \text{ for all } k.$$

Deterministic equivalent: A remarkable feature of this special case is that the asymptotic law $u = Lx$ doesn't in any way involve the probability distribution π for

the disturbances! We would end up with the same matrix L no matter what this distribution is (as long as it makes ω average out to 0). In fact, it is the same law that would be obtained if we dropped the disturbances entirely and looked only at the corresponding deterministic problem! Unfortunately, like the derivation of the law itself, this feature depends on the very simple linear-quadratic structure and *the absence of any constraints*.

Adjustment of the convergence proposal: The example of linear-quadratic regulation shows that the suggested equation for an "asymptotic" cost-to-go function ψ might not correspond quite to a limit of the functions ψ_k^N as $N \to \infty$. If we let $\bar{\psi}_k = \frac{1}{2} x \cdot K_k x$, so that $\psi_k = \bar{\psi}_k + \gamma_k$ for $\gamma_k = I\!E_\omega \{ \frac{1}{2} \omega \cdot K_{k+1} \omega \}$ with $\gamma_N = 0$, we see that $\bar{\psi}_k^N(x)$ converges to $\psi(x) = \frac{1}{2} x \cdot K x$, but

$$\psi_k^N = \bar{\psi}_k^N + \sum\nolimits_{k=0}^{N} \gamma_k \quad \text{with} \quad 0 \le \gamma_k \to I\!E_\omega \Big\{ \tfrac{1}{2} \omega \cdot K \omega \Big\}.$$

Unless $I\!E_\omega \{ \frac{1}{2} \omega \cdot K \omega \} = 0$, which would exclude applications with real disturbances, we'd get $\psi_k^N(x) \to \infty$ for all x. Thus, the convergence proposal needs some degree of modification, for instance in terms of discounting future costs by some factor.

Future mathematical challenges: Little is known about the proposed asymptotic equation in general, although an ability to solve it approximately could have interesting consequences in a number of situations where the designation of a particular horizon N is troublesome.

7. ADVANCED STOCHASTIC PROGRAMMING

This section of the lecture notes never got written as an addition to the earlier sections. However, the material that was intended to go into it was published separately as an article:

> "Duality and optimality in multistage stochastic programming,"
> Annals of Operations Research 85 (1999), 1-19.

This article is appended now to these notes.

DUALITY AND OPTIMALITY IN MULTISTAGE STOCHASTIC PROGRAMMING

Abstract. A model of multistage stochastic programming over a scenario tree is developed in which the evolution of information states, as represented by the nodes of a scenario tree, is supplemented by a dynamical system of state vectors controlled by recourse decisions. A dual problem is obtained in which multipliers associated with the primal dynamics are price vectors that are propagated backward in time through a dual dynamical system involving conditional expectation. A format of Fenchel duality is employed in order to have immediate specialization not only to linear programming but extended linear-quadratic programming. The resulting optimality conditions support schemes of decomposition in which a separate optimization problem is solved at each node of the scenario tree.

1. Introduction

In all branches of optimization, duality draws on convexity. In stochastic programming, another important ingredient to duality is dynamical structure. Such structure describes the evolution of information about the random elements in a problem's environment, and to that extent is essential to the very concept of optimization under uncertainty, but it can also be developed with respect to other ways that the past and future affect the present. Concepts of "state" and "control" are useful in this. Controls correspond to decisions to be made, whereas states summarize the current primal or dual status of the system at the time of making those decisions.

A distinguishing feature of the multistage model of stochastic programming to be laid out here is that the dynamics of uncertainty, discretized as a scenario tree in which nodes represent probabilistic states of information, is supplemented by a linear dynamical system of vectors representing auxiliary aspects of state. The relations in this system can be treated as constraints to which multiplier vectors are assigned, and those vectors become dual states in a dual dynamical system.

A forerunner to this model was developed in Rockafellar and Wets [1], but with the evolution of probabilistic information described by a sequence of fields of measurable sets in a discrete probability space. The notion of a scenario tree, adopted here instead, adds concreteness without serious loss of generality. In associating the nodes of the tree with decision stages, the crucial property of nonanticipativity of decisions is made automatic. Moreover, the role of conditional expectations in the dynamics of prices is clarified. Independently, Korf [2] has found an equivalent expression for that role in the measurability context of [1], but the scenario tree approach brings it out quite naturally.

Another difference between the model in [1] and the one here is that the framework of linear or extended linear-quadratic programming has been generalized to the extended Fenchel duality format, which is more flexible and less cluttered. Simultaneously, some features of the cost structure in [1], such as ways of writing initial and terminal costs, have been simplified. The problem should therefore be easier to understand and work with.

A powerful property of the optimality conditions in [1] emerges again here. It is seen in how the dynamical systems of primal and dual state vectors lead to a decomposition in which a small-scale optimization problem, depending on those vectors, is solved at each individual node of the scenario tree in order to obtain the optimal controls. This differs both from the classical Dantzig-Wolfe type of decomposition (which, in extension to convex programming, is tied to separability of the Lagrangian in the primal argument for fixed values of the dual argument) and from Benders decomposition (which concerns cost-to-go

functions and their subgradients). It relates instead to a characterization of primal and dual optimal solution pairs as saddle points of a generalized Lagrangian function that is the sum of two "sub-Lagrangians." One of these breaks into separate expressions for each node of the scenario tree, while the other is a bi-affine representation of the dynamics of the primal and dual state vectors.

Decomposition of this third kind dates back, in deterministic rather than stochastic settings, to Rockafellar [3], [4]. In stochastic programming computations it has recently been utilized by Salinger [5]. A corresponding numerical application to the deterministic case of the model in [1] has been carried out by Eckstein and Ferris [6]. The backward-forward splitting methods applied by Chen [7] to deterministic problems would be suited to this kind of decomposition also; for related results on the general convergence of such methods, see [8].

The Lagrangian saddle point scheme in [1] relies on a "reduced" formulation of the underlying problem, in which only the controls appear as independent variables. Here, in further contrast to [1], we look at a "full" formulation in tandem with the reduced formulation. In that way additional insights are obtained about the interpretation of the primal and dual state vectors.

2. Scenario Tree and Dynamics

We begin with a *scenario tree* based on a finite set I of nodes i. One of the nodes, $i = 0$, stands for the here-and-now. Every other node $i \neq 0$ has a unique predecessor node, denoted by i_-, and a transition probability $\tau_i > 0$, which is the probability that i will be the successor to i_-. The successors to any node i, denoted generically by i_+, form the set $I_+(i) \subset I$; the transition probabilities of those nodes add to 1. Thus, i_+ can be viewed as a discrete random variable over $I_+(i)$ with distribution given by the probabilities τ_{i_+}. Nodes i with $I_+(i) = \emptyset$ are called *terminal*; they constitute the set $T \subset I$.

In tracing back from any node i through its predecessors to 0, we trace in reverse a sequence of realizations of the discrete random variables associated with the transition probabilities. It's convenient to think of i as standing for this history of realizations, and to define π_i to be the probability of the particular sequence occurring. Such probabilities are generated recursively by

$$\pi_0 = 1, \qquad \pi_i = \tau_i \pi_{i_-} \text{ for } i \neq 0. \tag{2.1}$$

In the case of a node $i \in T$, the history of realizations corresponds to a path from the root 0 of the tree all the way to one of its "leaves" and is called a *scenario*. The probabilities π_i for $i \in T$ obviously add to 1 and provide a distribution for the set of all scenarios.

In many, or most, situations it may be helpful to view the node set I as partitioned into subsets I_k designating *stages* $k = 0, 1, \ldots, N$, where $I_0 = \{0\}$, $I_N = T$, and the successor to a node in I_k belongs to I_{k+1}. Mathematically, however, there's no actual need for that, so stage notation will be left out in the interests of simplicity. Further details about the scenario tree can always be brought in when it matters.

Every information state $i \in I$ is viewed as providing the opportunity for a decision to be made. We model this as the choice of a vector $u_i \in \mathbb{R}^{n_i}$; the vector u_0 gives rise to the "here-and-now" decision, whereas the vectors u_i for $i \neq 0$ give "recourse" decisions. Optimization revolves around these elements, which will be termed the *controls* of the system, but it's important that the decision environment in state i be able to be molded by the decisions taken in the states leading up to i. The mechanism for this is provided by the introduction for each $i \in I$ of a *state* vector $x_i \in \mathbb{R}^{d_i}$ and letting the states and controls be governed by a dynamical system

$$x_0 = a \text{ (given)}, \qquad x_i = F_i(x_{i_-}, u_{i_-}) \text{ for } i \neq 0. \tag{2.2}$$

The optimization environment in information state i is modeled then by a cost expression $f_i(x_i, u_i)$ (oriented toward minimization), in which the vector x_i acts as a parameter element supplying the influence from the past. This cost is allowed to be ∞ as a means of incorporating constraints without having to appeal at once to some particular constraint structure and its burdensome notation; a vector u_i is only considered feasible relative to x_i if $f_i(x_i, u_i) < \infty$. The forms of F_i and f_i will be specialized in due course, but it's useful for now to approach the situation more generally.

In attaching F_i, f_i and x_i by subscript to the information state i, we take the position that these elements are known to the decision maker upon reaching i; thus too, the function $f_i(x_i, \cdot)$ giving the costs (and implicit constraints) in choosing u_i is known. (Any random variables that enter the description of f_i are regarded as having been built into the specification of the transition probabilities encountered on the way from 0 to i.) Observe however that although F_i is supposed to be known upon reaching i, it might not have been known at the predecessor node i_-, when u_{i_-} had to be chosen. The explanation again is that in passing from i_- to i the dynamics in (2.2) may involve the realization of some additional aspect of uncertainty. Alternatively these dynamics can be written as

$$x_{i_+} = F_{i_+}(x_i, u_i) \text{ for } i \notin T, \quad \text{with } x_0 = a, \tag{2.3}$$

in which the role of i_+ as a random variable ranging over $I_+(i)$ is more apparent.

The stochastic programming problem we consider has two formulations, fundamentally equivalent yet different, and it will be important to distinguish between them. In the *full*

formulation, the problem is

$$(\mathcal{P}_0^+) \qquad \begin{array}{l} \text{minimize } \sum_{i\in I} \pi_i f_i(x_i, u_i) \text{ over all } x_i \text{ and } u_i \\ \text{subject to the relations (2.2) as constraints.} \end{array}$$

Implicit in this, as already mentioned, are the conditions $f_i(x_i, u_i) < \infty$, without which the expression being minimized would have the value ∞. In the *reduced* formulation, the x_i's are regarded as dependent rather than independent variables:

$$(\mathcal{P}_0^-) \qquad \begin{array}{l} \text{minimize } \sum_{i\in I} \pi_i f_i(x_i, u_i) \text{ over all } u_i, \text{ where} \\ x_i \text{ stands for an expression in prior control vectors.} \end{array}$$

The expressions in question are generated recursively from (2.2).

The two problems are obviously equivalent in the sense that vectors $\bar{u}_i$ and $\bar{x}_i$ furnish an optimal solution to $(\mathcal{P}_0^+)$ if and only if the controls $\bar{u}_i$ furnish an optimal solution to $(\mathcal{P}_0^-)$ and the states $\bar{x}_i$ furnish the corresponding trajectory obtained from them by "integrating" the dynamics (2.2). Both of these formulations are useful. Problem $(\mathcal{P}_0^-)$ has the advantage of being "smaller" and conceptually leaner, but $(\mathcal{P}_0^+)$ promotes the exploration of dual state vectors y_i, which come out as multipliers for the relations in (2.2) as constraints.

Theorem 1 (basic convexity)**.** *As long as the functions f_i are convex and the mappings F_i are affine, both $(\mathcal{P}_0^+)$ and $(\mathcal{P}_0^-)$ are problems of convex optimization—where a convex function is minimized subject to constraints describing a convex feasible set.*

The proof of this fact is elementary; we record the theorem for the perspective it offers, since convexity will be needed for duality. Note that the convexity depends heavily on the transition probabilities being unaffected by the decisions that are to be made over time; we are dealing with constants τ_i instead of variable expressions $\tau_i(x_{i_-}, u_{i_-})$. Problems $(\mathcal{P}_0^+)$ and $(\mathcal{P}_0^-)$ would still make sense if such variable transition probabilities were allowed, with the π_i's then turning into expressions in prior states and controls as generated through (2.1) and (2.2). Convexity would be lost, however, and with it the prospect of being able to use duality-based decomposition methods of solution to compensate for the extremely large scale of problems in stochastic programming.

3. Duality Background

From Theorem 1 it's clear that, for our purposes, the mappings F_i should be affine, but what structure should be introduced in the cost functions f_i to bring out duality most conveniently? Because the f_i's are extended-real-valued, constraint structure is at stake as well. We want multistage models of linear programming type to be covered nicely, and also quadratic programming analogs, for instance. Even in ordinary quadratic programming, however, there is trouble over duality. Unlike the situation in linear programming, one can't dualize a quadratic programming problem and expect to get another quadratic programming problem.

The kind of Lagrangian duality that is available from conventional formulations of convex programming with equality and inequality constraints is too narrow for the task now facing us and suffers from the further drawback that such formulations tend to emphasize "hard constraints," whereas the needs of stochastic programming may often be better served by penalty expressions. The Fenchel scheme of duality comes to the rescue here. It's much more flexible, yet just as explicit in key cases. In particular, it gets around the quadratic programming difficulty by way of "extended linear-quadratic programming," which handles penalties and even box constraints with ease. The ideas behind Fenchel duality will be reviewed now as background to presenting, in the next section, more structured versions of problems $(\mathcal{P}_0^+)$ and $(\mathcal{P}_0^-)$. A fresh treatment of such duality in more detail is available now in [9, Chap. 11].

Recall that an extended-real-valued convex function φ on $\mathbb{R}^n$ is *proper* if it nowhere takes on $-\infty$ and is not the constant function ∞. The function φ^* *conjugate* to a proper convex function φ is defined by

$$\varphi^*(w) = \sup_{u \in \mathbb{R}^n} \big\{u \cdot w - \varphi(u)\big\}.$$

It's always proper convex and lsc (lower semicontinuous). As long as φ itself is lsc, the function φ^{**} conjugate to φ^* is in turn φ:

$$\varphi(w) = \sup_{w \in \mathbb{R}^n} \big\{u \cdot w - \varphi^*(w)\big\}.$$

Although it may be hard in some cases to come up with a more explicit formula for φ^* than the definition, there are cases where it's easy, and they go a long way toward making conjugate functions a practical tool in duality schemes.

The *extended* Fenchel duality scheme that will serve as our basic guide concerns a proper lsc convex function φ on $\mathbb{R}^n$, another such function ψ on $\mathbb{R}^m$, a matrix $D \in \mathbb{R}^{n \times m}$, and vectors $p \in \mathbb{R}^n$ and $q \in \mathbb{R}^m$. The primal problem has the form

$$(\mathcal{P}) \qquad \text{minimize } \Phi(u) := p \cdot u + \varphi(u) + \psi(q - Du) \text{ over } u \in \mathbb{R}^n,$$

while the dual problem is

$$(\mathcal{D}) \qquad \text{maximize } \Psi(v) := q{\cdot}v - \psi^*(v) - \varphi^*(D^*v - p) \text{ over } v \in I\!R^m,$$

where D^* is the transpose of D. Implicit constraints come out of the effective domains $\operatorname{dom}\varphi := \{u \in I\!R^n \,|\, \varphi(u) < \infty\}$ and $\operatorname{dom}\psi := \{z \in I\!R^m \,|\, \psi(z) < \infty\}$. The implicit feasible set in $(\mathcal{P})$ is the convex set consisting of the vectors u that satisfy

$$u \in \operatorname{dom}\varphi, \quad q - Du \in \operatorname{dom}\psi. \tag{3.1}$$

Similarly, the implicit feasible set in $(\mathcal{D})$ is described by

$$v \in \operatorname{dom}\psi^*, \quad D^*v - p \in \operatorname{dom}\varphi^*. \tag{3.2}$$

Examples will be considered after the main results about this pairing of problems are stated.

For problems in this format, the constraint qualifications needed to obtain duality in general are expressed in terms of the notion of the relative interior "ri" of a convex set. Such constraint qualifications turn out not to be needed for functions that are *piecewise linear-quadratic*. A proper convex function φ is said to fall into that category if $\operatorname{dom}\varphi$ is a polyhedral (convex) set on which φ is given by a linear-quadratic formula, or a union of finitely many such sets on which φ is given by such formulas. By a linear-quadratic formula we mean a polynomial function of degree at most 2; linear functions and constant functions are a special case. For instance if φ is the indicator δ_U of a polyhedral set U (i.e., has the value 0 on U and ∞ elsewhere), then in particular, φ is piecewise linear-quadratic, although the full generality of the definition isn't utilized.

An important fact is this: if a proper convex function is piecewise linear-quadratic, its conjugate function is piecewise linear-quadratic as well. Thus, if φ and ψ are piecewise linear-quadratic in $(\mathcal{P})$, the same is true of φ^* and ψ^* in $(\mathcal{D})$. We refer to this as the *piecewise linear-quadratic* case in Fenchel duality.

Theorem 2 (extended Fenchel duality)**.**

(a) *The relation* $\inf(\mathcal{P}) = \sup(\mathcal{D}) < \infty$ *is guaranteed under the primal constraint qualification that*

$$\exists\, u \text{ satisfying} \quad u \in \operatorname{ri}\operatorname{dom}\varphi, \quad q - Du \in \operatorname{ri}\operatorname{dom}\psi. \tag{3.3}$$

Then too, unless the common optimal value is $-\infty$ *(so* $(\mathcal{D})$ *has no feasible solution),* $(\mathcal{D})$ *is sure to have an optimal solution.*

(b) *The relation* $\inf(\mathcal{P}) = \sup(\mathcal{D}) > -\infty$ *is guaranteed under the dual constraint qualification that*

$$\exists\, v \text{ satisfying } \quad v \in \operatorname{ri}\operatorname{dom}\psi^*, \quad D^*v - p \in \operatorname{ri}\operatorname{dom}\varphi^*. \tag{3.4}$$

Then too, unless the common optimal value is ∞ *(so* $(\mathcal{P})$ *has no feasible solution),* $(\mathcal{P})$ *is sure to have an optimal solution.*

(c) *In the piecewise linear-quadratic case, the primal and dual constraint qualifications are superfluous and can be replaced simply by the feasibility conditions in (3.1) and (3.2), respectively. In that case, therefore,*

$$\inf(\mathcal{P}) = \sup(\mathcal{D}) \quad \textit{unless} \quad \inf(\mathcal{P}) = \infty \textit{ and } \sup(\mathcal{D}) = -\infty,$$

(i.e., unless neither $(\mathcal{P})$ *nor* $(\mathcal{D})$ *has a feasible solution). Moreover, when the common optimal value is finite, both problems have an optimal solution.*

Proof. The basic facts in (a) and (b) go back all the way to Rockafellar [10]. The piecewise linear-quadratic case, while partially covered earlier, was recently established in its full scope in Rockafellar and Wets [9; cf. 11.42]. □

Among the special cases to note here, *linear programming* duality corresponds to taking φ to be the indicator of $\mathbb{R}^n_+$ and ψ to be the indicator of $\mathbb{R}^m_-$, so that $(\mathcal{P})$ comes out as minimizing $p{\cdot}u$ subject to $u \geq 0$ and $Du \geq q$. Then φ^* and ψ^* are the indicators of $\mathbb{R}^n_-$ and $\mathbb{R}^m_+$, so that $(\mathcal{D})$ consists of maximizing $q{\cdot}v$ subject to $v \geq 0$ and $D^*v \leq p$. This is covered by part (c) of Theorem 2.

The orthants here could be replaced by other convex cones. (The function conjugate to the indicator of a cone is the indicator of the polar cone.) More interesting for stochastic programming, however, is the case where $\mathbb{R}^n$ is replaced by some *box* (a product of closed intervals, bounded or unbounded). When $\varphi = \delta_U$, φ^* is the support function σ_U of U, and for a box U that is bounded this means φ^* is piecewise linear (and easy to write down explicitly). Even more to the point is the case where ψ is such a support function σ_V of a box V, so that $\psi^* = \delta_V$. The term $\psi(q - Du)$ in $(\mathcal{P})$ corresponds then to a linear penalty expression in place of, say, a constraint like $q - Du \leq 0$. There are rich possibilities.

A handy tool in this respect is that of the function $\theta_{V,Q}$ on $\mathbb{R}^m$ defined in terms of a nonempty polyhedral set $V \subset \mathbb{R}^m$ and a symmetric positive *semi*-definite matrix $Q \in \mathbb{R}^{m\times m}$ ($Q = 0$ allowed) by

$$\theta_{V,Q}(z) = \sup_{v\in V} \left\{ z{\cdot}v - \tfrac{1}{2} v{\cdot}Qv \right\}. \tag{3.5}$$

This means that $\theta_{V,Q}$ is the convex function conjugate to $\delta_V + j_Q$, where $j_Q(v) = \frac{1}{2}v{\cdot}Qv$. Since $\delta_V + j_Q$ falls in the category of piecewise linear-quadratic functions, the same is true of $\theta_{V,Q}$. For various useful choices of V and Q it's possible to make this linear-quadratic structure of $\theta_{V,Q}$ quite explicit. Analogously, functions $\theta_{U,P}$ can be introduced on $\mathbb{R}^n$ as the piecewise linear-quadratic conjugates of functions $\delta_U + j_P$ for a polyhedral set $U \subset \mathbb{R}^n$ and symmetric positive *semi*-definite matrix $P \in \mathbb{R}^{n\times n}$.

In taking $\varphi = \delta_U + j_P$ and $\psi = \theta_{V,Q}$, so that $\varphi^* = \theta_{U,P}$ and $\psi^* = \delta_V + j_Q$, we obtain the following pair of problems from $(\mathcal{P})$ and $(\mathcal{D})$:

$$(\mathcal{P}') \qquad \text{minimize } p{\cdot}u + \tfrac{1}{2}u{\cdot}Pu + \theta_{V,Q}(q - Du) \text{ over } u \in U,$$

$$(\mathcal{D}') \qquad \text{maximize } q{\cdot}v - \tfrac{1}{2}v{\cdot}Qv - \theta_{U,P}(D^*v - p) \text{ over } v \in V.$$

This is the duality scheme of *extended linear-quadratic programming*. It too is governed by part (c) of Theorem 2. Linear programming comes out when U and V are cones while P and Q are zero matrices. As another example, conventional quadratic programming would consist of minimizing $p{\cdot}u + \frac{1}{2}u{\cdot}Pu$ subject to $u \in U$ and $Du \geq q$, where U is $\mathbb{R}^n_+$ or perhaps some other box. A problem of such type can't be dualized within that format, but in the framework of extended linear-quadratic programming the dual problem consists of maximizing $q{\cdot}v - \theta_{U,P}(D^*v - p)$ over $v \geq 0$. (The implicit constraint $D^*v - p \in \operatorname{dom}\theta_{U,P}$ combines with $v \geq 0$ to produce the implicit feasible set in this dual.)

Because stochastic programming is our subject here, it's worth mentioning that piecewise linear-quadratic functions of type $\theta_{V,Q}$ were first introduced in a stochastic programming context, in Rockafellar and Wets [11]. This was motivated by the convenience of such functions in furnishing penalty expressions in a readily dualizable form. Penalty substitutes for constraints are especially welcome when dealing with uncertainty. The format of extended linear-quadratic programming in $(\mathcal{P}')$ and $(\mathcal{D}')$ comes from [11] as well. Examples of the many special problem statements covered by it were subsequently presented in [3; §§2,3]; for stochastic programming, see also [12], where the separable case of functions $\theta_{V,Q}$ is well described.

Although Fenchel duality isn't based on Lagrange multipliers, at least of the traditional variety, a Lagrangian function plays a crucial role nonetheless. This Lagrangian in the general case of $(\mathcal{P})$ and $(\mathcal{D})$ is

$$\begin{aligned} L(u,v) &= p{\cdot}u + \varphi(u) + q{\cdot}v - \psi^*(v) - v{\cdot}Du \\ &\qquad \text{on } \ U \times V, \ \text{ where } \ U := \operatorname{dom}\varphi, \ \ V := \operatorname{dom}\psi^*. \end{aligned} \tag{3.6}$$

In the extended linear-quadratic programming format of problems $(\mathcal{P}')$ and $(\mathcal{D}')$, the generalized Lagrangian that takes on the right role is

$$L(u,v) = p{\cdot}u + \tfrac{1}{2}p{\cdot}Pu + q{\cdot}v - \tfrac{1}{2}v{\cdot}Qv - v{\cdot}Du \quad \text{ on } U \times V. \tag{3.7}$$

The Lagrangians associated with extended linear-quadratic programming are thus the functions obtained by restricting some convex-concave linear-quadratic function to a product of nonempty polyhedral sets.

Note that the Lagrangian in each case isn't a function with unspecified domain, but a triple (L, U, V). This entire triple enters the picture through the way that the objective functions in the two problems can be recovered from L, U and V by

$$\Phi(u) = \begin{cases} \sup_{v \in V} L(u,v) & \text{when } u \in U \\ \infty & \text{when } u \notin U, \end{cases} \tag{3.8}$$

$$\Psi(v) = \begin{cases} \inf_{u \in U} L(u,v) & \text{when } v \in V \\ -\infty & \text{when } v \notin V. \end{cases} \tag{3.9}$$

It also enters in saddle point characterizations of optimality, as in the next theorem. Recall that $(\bar{u}, \bar{v})$ is a *saddle point of L on $U \times V$* when

$$\begin{cases} \bar{u} \in U, \ \bar{v} \in V, \ \text{and} \\ L(u,\bar{v}) \geq L(\bar{u},\bar{v}) \geq L(\bar{u},v) \ \text{ for all } \ u \in U, \ v \in V. \end{cases} \tag{3.10}$$

Theorem 3 (Lagrangians in Fenchel duality)**.** *In the circumstances of Theorem 2 in which* $\inf(\mathcal{P}) = \sup(\mathcal{D})$*, a pair* $(\bar{u}, \bar{v})$ *is a saddle point of the Lagrangian L over $U \times V$ in (3.6) if and only if* $\bar{u}$ *is an optimal solution to* $(\mathcal{P})$ *and* $\bar{v}$ *is an optimal solution to* $(\mathcal{D})$*. This saddle point property of* $(\bar{u}, \bar{v})$ *is equivalent to the subgradient conditions*

$$D^*\bar{v} - p \in \partial\varphi(\bar{u}), \qquad q - D\bar{u} \in \partial\psi^*(\bar{v}). \tag{3.11}$$

Proof. The original saddle point characterization of optimal solutions in Fenchel duality was developed in [13], where the corresponding subgradient conditions were first given as well. More recently see also [9; Chap. 11]. □

The saddle point characterization of optimality assists in interpreting $\bar{v}$ as a "generalized multiplier vector" associated with the term $\psi(q - Du)$ in $(\mathcal{P})$. This perspective is opened further in [14].

The formulas in (3.8) and (3.9) for the objectives in $(\mathcal{P})$ and $(\mathcal{D})$ in terms of L, U and V lead to a general principle that can help us, in more complicated situations in other notation, to ascertain whether a given optimization problem fits the Fenchel format, and if

so, what the corresponding dual problem is. All we need to know is whether the function of u (say) that is being minimized in the given problem can be expressed by *the right side of* (3.8) through a function $L(u, v)$ of the type in (3.6)—and that too can be viewed very broadly: $L(u, v)$ need only be *the difference between two lsc proper convex functions of u and v separately (as restricted to their effective domains U and V), plus some expression that's bi-affine in u and v* (i.e., affine in u for fixed v as well as affine in v for fixed u). Once this has been confirmed, we can identify the dual with the problem of maximizing the function of v given by *the right side of* (3.9), and the theorems above can be applied. The point here is that we can bypass having to write down what the vectors p and q and the matrix D are in a given case in order to invoke Fenchel duality. The objective in the dual problem can be deduced straight from the right side of (3.9) without that distraction. For stochastic programming in particular, that will be the most expedient approach to dualization.

4. Stochastic Programming Duality

The stage is set now for the specialization of the general stochastic programming problems $(\mathcal{P}_0^+)$ and $(\mathcal{P}_0^-)$ to models that are able to take advantage of extended Fenchel duality. We choose the mappings F_i to be linear in the notation

$$F_i(x_{i-}, u_{i-}) = A_i x_{i-} + B_i u_{i-} \tag{4.1}$$

for matrices A_i and B_i, so that the vectors x_i have the dynamics

$$\begin{cases} x_0 = a, \\ x_i = A_i x_{i-} + B_i u_{i-} \quad \text{for } i \neq 0. \end{cases} \tag{4.2}$$

We take the cost functions f_i to have the form

$$f_i(x_i, u_i) = p_i \cdot u_i + \varphi_i(u_i) + \psi_i(q_i - C_i x_i - D_i u_i) \tag{4.3}$$

for matrices C_i and D_i, vectors p_i and q_i, and lsc proper convex functions φ_i and ψ_i on $\mathbb{R}^{n_i}$ and $\mathbb{R}^{m_i}$, respectively. As the subscripting by i indicates, all these elements are regarded as known to the decision maker once the information state i has been reached.

(Affine mappings $F_i(x_{i-}, u_{i-}) = A_i x_{i-} + B_i u_{i-} + b_i$ could be handled in place of the linear ones in (4.2), but the additional notation gets cumbersome. Anyway, no real generality would be gained, because the vector b_i could equally well be incorporated as an extra column of B_i for which the corresponding component of u_{i-} has to be 1, as secured implicitly through the specification of the effective domain of the corresponding φ_i.)

Let $u = \{u_i\}_{i\in I}$ be the "supervector" of controls u_i, and similarly let $x = \{x_i\}_{i\in I}$. We have $u \in I\!R^n := \Pi_{i\in I} I\!R^{n_i}$ and $x \in I\!R^d := \Pi_{i\in I} I\!R^{d_i}$. Let $X : I\!R^n \to I\!R^d$ be the affine mapping defined by

$$X(u) = \text{ the primal state trajectory generated from } u \text{ by the dynamics (4.2).}$$

Our stochastic programming problem in its *full* formulation is then

$$(\mathcal{P}^+) \qquad \begin{array}{l} \text{minimize } \Phi^+(x,u) := \sum_{i\in I} \pi_i \big[p_i{\cdot}u_i + \varphi_i(u_i) + \psi_i(q_i - C_i x_i - D_i u_i)\big] \\ \text{over } x \text{ and } u, \text{ subject to } x - X(u) = 0, \end{array}$$

whereas in its *reduced* formulation it is

$$(\mathcal{P}^-) \qquad \begin{array}{l} \text{minimize } \Phi^-(u) := \sum_{i\in I} \pi_i \big[p_i{\cdot}u_i + \varphi_i(u_i) + \psi_i(q_i - C_i x_i - D_i u_i)\big] \\ \text{over } u, \text{ where } x = X(u). \end{array}$$

In the full version the equations in (4.2) are taken as a system of linear constraints on the vector variables x_i and u_i. In the reduced version, though, x_i merely stands for an affine expression in the control vectors associated with the information states leading up to i. Those expressions can be generated out of (4.2) by a chain of substitutions, but it won't actually be necessary to do that in order to make effective use of $(\mathcal{P}^-)$. The implicit conditions for feasibility in both problems can anyway be written as

$$u_i \in \operatorname{dom}\varphi_i, \quad q_i - C_i x_i - D_i u_i \in \operatorname{dom}\psi_i, \quad x = X(u), \tag{4.4}$$

whichever point of view is being adopted. Obviously $(\mathcal{P}^+)$ and $(\mathcal{P}^-)$ are equivalent in the sense that $\inf(\mathcal{P}^+) = \inf(\mathcal{P}^-)$ and

$$(\bar{x},\bar{u}) \text{ solves } (\mathcal{P}^+) \iff \bar{u} \text{ solves } (\mathcal{P}^-) \text{ and } \bar{x} = X(\bar{u}). \tag{4.5}$$

Dualization will proceed first with the full primal problem $(\mathcal{P}^+)$. The full dual problem $(\mathcal{D}^+)$ obtained in this way will have a reduced form $(\mathcal{D}^-)$, which will be shown later to be dual to the reduced problem $(\mathcal{P}^-)$ with respect to a reduced Fenchel scheme. Let

$$\begin{array}{l} L_i(u_i,v_i) := p_i{\cdot}u_i + \varphi_i(u) + q_i{\cdot}v_i - \psi_i^*(v_i) - v_i{\cdot}D_i u_i \\ \qquad\qquad \text{on } U_i \times V_i := [\operatorname{dom}\varphi_i] \times [\operatorname{dom}\psi_i^*] \subset I\!R^{n_i} \times I\!R^{m_i}. \end{array} \tag{4.6}$$

Let $v = \{v_i\}_{i\in I}$ in $I\!R^m := \Pi_{i\in I} I\!R^{m_i}$ and define

$$U = \Pi_{i\in I} U_i \subset I\!R^n, \qquad V = \Pi_{i\in I} V_i \subset I\!R^m. \tag{4.7}$$

The sets U and V are convex.

In terms of $y_i \in I\!R^{d_i}$ and $y = \{y_i\}_{i\in I} \in I\!R^d$, we take as the *full Lagrangian* L^+, associated with $(\mathcal{P}^+)$, the expression

$$L^+(x,u;y,v) := \sum_{i\in I} \pi_i \big[L_i(u_i,v_i) - v_i{\cdot}C_i x_i\big] + \sum_{i\neq 0} \pi_i y_i{\cdot}\big[x_i - A_i x_{i_-} - B_i u_{i_-}\big] + y_0{\cdot}[x_0 - a]$$
$$\text{on} \quad [I\!R^d \times U] \times [I\!R^d \times V]. \tag{4.8}$$

The vectors $y_i \in I\!R^{d_i}$ are multipliers in the traditional sense for the equations in (4.2) as constraints in $(\mathcal{P}^+)$ (so that, in overview, y is a multiplier for the constraint $x - X(u) = 0$). The vectors v_i, on the other hand, will act as generalized multipliers, in the sense of Fenchel duality, for the ψ_i terms in the objective of $(\mathcal{P}^+)$.

The principle set down at the end of §3 will guide our effort at dualization. We apply this principle by thinking of $L^+(x,u;y,v)$ as $L^+(u',v')$ for $u' = (x,u) \in U' = I\!R^d \times U$ and $v' = (y,v) \in V' = I\!R^d \times V$. Calculating $\sup_{v'\in V'} L^+(u',v')$ as on the right side of (3.8), we get the function Φ^+ in $(\mathcal{P}^+)$ as restricted by the dynamics in (4.2): namely, whereas $\Phi^+(x,u) = \infty$ when $u \notin U$, we have for $u \in U$ that

$$\sup_{(y,v)\in \boldsymbol{R}^d\times V} L^+(x,u;v,y) = \begin{cases} \Phi^+(x,u) & \text{when } x = X(u), \\ \infty & \text{when } x \neq X(u). \end{cases} \tag{4.9}$$

Next we observe that L^+ has the form required for Fenchel duality: it's the difference between the lsc proper convex functions

$$\varphi^+(u') = \sum_{i\in I} \varphi_i(u_i), \qquad (\psi^+)^*(v') = \sum_{i\in I} \psi_i^*(v_i) \tag{4.10}$$

(not really depending on x and y), as restricted to their effective domains U' and V', plus an expression that's affine in u' for fixed v' and affine in v' for fixed u'. It follows by our principle that, in the Fenchel duality framework, $(\mathcal{P}^+)$ is the primal problem associated with L^+ on $U' \times V'$, and moreover that the corresponding dual problem can be obtained by developing the expression

$$\inf_{u'\in U'} L^+(u',v') = \inf_{(x,u)\in \boldsymbol{R}^d\times U} L^+(x,u;y,v) \tag{4.11}$$

on the right side of (3.9).

This calculation is facilitated by a notation for *conditional expectation* in a state i with respect to its successor states i_+. We'll set

$$E_i\{w_{i_+}\} := \sum_{i_+\in I_+(i)} \tau_{i_+} w_{i_+} \tag{4.12}$$

when a vector w_{i_+} depends on i_+.
The trick now is to rewrite the linear and bilinear terms in $L^+(x,u;y,v)$ from the (i_-,i) mode to the (i,i_+) mode, in which

$$x_{i_+} = A_{i_+}x_i + B_{i_+}u_i \text{ for } i \notin T. \tag{4.13}$$

Denoting the transposes of A_i, B_i and C_i by A_i^*, B_i^* and C_i^*, it's easy to see in this way, through (2.1), that

$$\begin{aligned}
\sum_{i\neq 0}\pi_i y_i{\cdot}A_i x_{i_-} &= \sum_{i\notin T}\left[\sum_{i_+\in I_+(i)}\pi_{i_+}y_{i_+}{\cdot}A_{i_+}x_i\right]\\
&= \sum_{i\notin T}\left[\sum_{i_+\in I_+(i)}\pi_i\tau_{i_+}x_i{\cdot}A_{i_+}^*y_{i_+}\right] = \sum_{i\notin T}\pi_i x_i{\cdot}E_i\{A_{i_+}^*y_{i_+}\},\\
\sum_{i\neq 0}\pi_i y_i{\cdot}B_i u_{i_-} &= \sum_{i\notin T}\left[\sum_{i_+\in I_+(i)}\pi_{i_+}y_{i_+}{\cdot}B_{i_+}u_i\right]\\
&= \sum_{i\notin T}\left[\sum_{i_+\in I_+(i)}\pi_i\tau_{i_+}u_i{\cdot}B_{i_+}^*y_{i_+}\right] = \sum_{i\notin T}\pi_i u_i{\cdot}E_i\{B_{i_+}^*y_{i_+}\},
\end{aligned}$$

and consequently, for potential use in the context of (4.8), that

$$\begin{aligned}
&-\sum_{i\in I}\pi_i v_i{\cdot}C_i x_i + \sum_{i\neq 0}\pi_i y_i{\cdot}\big[x_i - A_i x_{i_-} - B_i u_{i_-}\big] + y_0{\cdot}[x_0 - a]\\
&\quad= \sum_{i\notin T}\pi_i x_i{\cdot}\big[y_i - E_i\{A_{i_+}^*y_{i_+}\} - C_i^*v_i\big] + \sum_{i\in T}\pi_i x_i{\cdot}\big[y_i - C_i^*v_i\big]\\
&\qquad - \sum_{i\notin T}\pi_i u_i{\cdot}E_i\{B_{i_+}^*y_{i_+}\} - y_0{\cdot}a.
\end{aligned} \tag{4.14}$$

The vectors x_i can be perceived now as multipliers for the constraints associated with the dynamical system

$$\begin{cases} y_i = C_i^*v_i & \text{for } i\in T,\\ y_i = E_i\{A_{i_+}^*y_{i_+}\} + C_i^*v_i & \text{for } i\notin T, \end{cases} \tag{4.15}$$

in which the vectors y_i can be interpreted as *dual* states, propagating backward in time in response to the vectors v_i as *dual* controls. Let $Y : \mathbb{R}^m \to \mathbb{R}^d$ be the affine mapping defined by

$$Y(v) = \text{ the dual state trajectory generated from } v \text{ by the dynamics (4.15).}$$

In expressing $L^+(x,u;y,v)$ in terms of the right side of (4.14) in place of the left, and performing the minimization in (4.11), we now get, as the *full* dual problem,

$$(\mathcal{D}^+) \qquad \begin{array}{l} \text{maximize} \\ \Psi^+(y,v) := \sum_{i\in I} \pi_i\big[q_i{\cdot}v_i - \psi_i^*(v_i) - \varphi^*\big(E_i\{B_{i_+}^* y_{i_+}\} + D_i^* v_i - p_i\big)\big] - a{\cdot}y_0 \\ \text{over } y \text{ and } v, \text{ subject to } y - Y(v) = 0. \end{array}$$

Here the equations in (4.15) are taken as a system of linear constraints on the vector variables y_i and v_i. In analogy with the foregoing we can immediately also write down a corresponding *reduced* dual problem, namely

$$(\mathcal{D}^-) \qquad \begin{array}{l} \text{maximize} \\ \Psi^-(v) := \sum_{i\in I} \pi_i\big[q_i{\cdot}v_i - \psi_i^*(v_i) - \varphi^*\big(E_i\{B_{i_+}^* y_{i_+}\} + D_i^* v_i - p_i\big)\big] - a{\cdot}y_0 \\ \text{over } v, \text{ where } y = Y(v). \end{array}$$

The feasibility conditions can be written for both problems as

$$v_i \in \operatorname{dom}\psi_i^*, \quad E_i\{B_{i_+}^* y_{i_+}\} + D_i^* v_i - p_i \in \operatorname{dom}\varphi_i^*, \quad y = Y(v). \tag{4.16}$$

It's clear that $(\mathcal{D}^+)$ and $(\mathcal{D}^-)$ are equivalent in the sense that $\inf(\mathcal{D}^+) = \inf(\mathcal{D}^-)$ and

$$(\bar{y},\bar{v}) \text{ solves } (\mathcal{D}^+) \iff \bar{v} \text{ solves } (\mathcal{D}^-) \text{ and } \bar{y} = Y(\bar{v}). \tag{4.17}$$

Theorem 4 (Fenchel scheme in the full model)**.** *The full problems $(\mathcal{P}^+)$ and $(\mathcal{D}^+)$ are dual to each other in the extended Fenchel sense with Lagrangian L^+ on $[\mathbb{R}^d \times U] \times [\mathbb{R}^d \times V]$. In this, the piecewise linear-quadratic case is the one in which all the convex functions φ_i and ψ_i (or equivalently φ_i^* and ψ_i^*) are piecewise linear-quadratic. The primal constraint qualification in (3.3) comes out as the strict feasibility condition*

$$\exists\, u_i \in \operatorname{ri}\operatorname{dom}\varphi_i \;\; \textit{with} \;\; q_i - C_i x_i - D_i u_i \in \operatorname{ri}\operatorname{dom}\psi_i \;\; \textit{for} \;\; x = X(u),\; i \in I, \tag{4.18}$$

whereas the dual constraint qualification in (3.4) corresponds to the strict feasibility condition

$$\exists\, v_i \in \operatorname{ri}\operatorname{dom}\psi_i^* \;\; \textit{with} \;\; E_i\{B_{i_+}^* y_{i_+}\} + D_i^* v_i - p_i \in \operatorname{ri}\operatorname{dom}\varphi_i^* \;\; \textit{for} \;\; y = Y(v),\; i \in I. \tag{4.19}$$

Proof. The preceding derivation has shown that these problems fit the framework of Fenchel duality in which φ^+ and $(\psi^+)^*$ are the functions in (4.10). In terms of the vector variable $v' = (y,v)$ being dual to $z' = (w,z)$, ψ^+ itself is given by

$$\psi^+(w,z) = \begin{cases} \sum_{i\in I} \psi_i(z_i) & \text{when } w = 0, \\ \infty & \text{when } w \neq 0. \end{cases}$$

Feasibility in this problem, which in the Fenchel scheme takes the form

$$(x,u) \in \operatorname{dom}\varphi^+, \qquad M(x,u) \in \operatorname{dom}\psi^+, \tag{4.20}$$

for a certain affine transformation M, has to correspond to (4.4); it's apparent that M must be the transformation that takes (x,u) to $(M_1(x,u), M_2(x,u))$ with $M_1(x,u)$ the element $w = x - X(u)$ and $M_2(x,u)$ the element $z = \{z_i\}_{i\in I}$ with $z_i = q_i - C_i x_i - D_i u_i$. The relative interior of a product of convex sets is the product of their relative interiors, so in replacing $\operatorname{dom}\varphi^+$ and $\operatorname{dom}\psi^+$ in (4.20) by $\operatorname{ri}\operatorname{dom}\varphi^+$ and $\operatorname{ri}\operatorname{dom}\psi^+$ we simply replace the sets $\operatorname{dom}\varphi_i$ and $\operatorname{dom}\psi_i$ in (4.4) by $\operatorname{ri}\operatorname{dom}\varphi_i$ and $\operatorname{ri}\operatorname{dom}\psi_i$. This confirms that the strict feasibility conditions in (4.18) do correspond to the constraint qualification obtained for $(\mathcal{P}^+)$ through the theory in §3.
In like manner, the strict feasibility conditions in (4.19) can be seen to correspond to the dual constraint qualification (3.4) as applied to $(\mathcal{D}^+)$. □

The direct connection between the reduced problems $(\mathcal{P}^-)$ and $(\mathcal{D}^-)$ can now be brought to light. For this purpose we define

$$l(u,v) = \text{ common value of both sides of (4.14) when } x = X(u) \text{ and } y = Y(v),$$

so that

$$l(u,v) = -\sum_{i\in I} \pi_i v_i \cdot C_i x_i \quad \text{for } x = X(u) \tag{4.21}$$

but at the same time

$$l(u,v) = -\sum_{i\notin T} \pi_i u_i \cdot E_i\{B_{i_+}^* y_{i_+}\} - y_0 \cdot a \quad \text{for } y = Y(v). \tag{4.22}$$

The value $l(u,v)$ is affine in its dependence on u for fixed v, as well as affine in its dependence on v for fixed u. Next we define the *reduced* Lagrangian L^- by

$$L^-(u,v) := \sum_{i\in I} \pi_i L_i(u_i, v_i) + l(u,v) \quad \text{on} \quad U \times V, \tag{4.23}$$

where U and V are given still by (4.7).

Theorem 5 (Fenchel scheme in the reduced model)**.** *The reduced problems $(\mathcal{P}^-)$ and $(\mathcal{D}^-)$ are dual to each other in the extended Fenchel sense with Lagrangian L^- on $U \times V$. In this, the piecewise linear-quadratic case is the one in which all the convex functions φ_i and ψ_i (or equivalently φ_i^* and ψ_i^*) are piecewise linear-quadratic. Again, the primal*

constraint qualification (3.3) comes out as (4.18), whereas the dual constraint qualification (3.4) comes out as (4.19).

Proof. Once more we appeal to the principle at the end of §3. The reduced Lagrangian L^- is the difference of the lsc, proper, convex functions

$$\varphi^-(u) = \sum_{i\in I} \varphi_i(u_i), \qquad [\psi^-]^*(u) = \sum_{i\in I} \psi_i^*(u_i),$$

as restricted to the product of their effective domains, namely $U\times V$, plus terms aggregating to an expression that is affine separately in u and v. Here $[\psi^-]^*$ is conjugate to $\psi^-(u) = \sum_{i\in I}\psi_i(u_i)$. On the basis of the two ways of looking at $l(u,v)$ in (4.21) and (4.22), we calculate that the objective functions specified in $(\mathcal{P}^-)$ and $(\mathcal{D}^-)$ have the form

$$\Phi^-(u) = \sup_{v\in V} L^-(u,v), \qquad \Psi^-(v) = \inf_{u\in U} L^-(u,v),$$

so these problems are indeed the ones that correspond in the Fenchel duality format to the triple (L^-, U, V).
The justification of the constraint qualifications follows the same argument as given in the proof of Theorem 4. □

Theorems 4 and 5 combine immediately with Theorem 2 to give us the following results for stochastic programming problems in either formulation.

Theorem 6 (stochastic programming duality)**.**
(a) *The relation* $\inf(\mathcal{P}^+) = \sup(\mathcal{D}^+) < \infty$ *is guaranteed by (4.18). Then, unless the common optimal value is* $-\infty$ *(so* $(\mathcal{D}^+)$ *has no feasible solution), problem* $(\mathcal{D}^+)$ *is sure to have an optimal solution.*
(b) *The relation* $\inf(\mathcal{P}^+) = \sup(\mathcal{D}^+) > -\infty$ *is guaranteed by (4.19). Then, unless the common optimal value is* ∞ *(so* $(\mathcal{P}^+)$ *has no feasible solution), problem* $(\mathcal{P}^+)$ *is sure to have an optimal solution.*
(c) *In the piecewise linear-quadratic case, the primal and dual constraint qualifications in (4.18) and (4.19) are superfluous and can be replaced simply by the feasibility conditions in (4.4) and (4.16), respectively. In that case,*

$$\inf(\mathcal{P}^+) = \sup(\mathcal{D}^+) \quad \textit{unless} \quad \inf(\mathcal{P}^+) = \infty \ \textit{and} \ \sup(\mathcal{D}^+) = -\infty,$$

(i.e., unless neither $(\mathcal{P}^+)$ *nor* $(\mathcal{D}^+)$ *has a feasible solution). Moreover, when the common optimal value is finite, both problems have an optimal solution.*

(d) *All these results hold equally with the full problems* $(\mathcal{P}^+)$ *and* $(\mathcal{D}^+)$ *replaced by the reduced problems* $(\mathcal{P}^-)$ *and* $(\mathcal{D}^-)$.

Examples of multistage stochastic programming problems that are covered by the results in Theorem 6 are easily generated from the examples in §3. Stochastic *linear* programming is obtained by choosing

$$\varphi_i(u_i) = \begin{cases} 0 & \text{if } u_i \in I\!R^{n_i}_+, \\ \infty & \text{if } u_i \notin I\!R^{n_i}_+, \end{cases} \qquad \psi_i(z_i) = \begin{cases} 0 & \text{if } z_i \in I\!R^{m_i}_-, \\ \infty & \text{if } z_i \notin I\!R^{m_i}_-, \end{cases}$$

so that

$$\varphi_i^*(w_i) = \begin{cases} 0 & \text{if } w_i \in I\!R^{n_i}_-, \\ \infty & \text{if } w_i \notin I\!R^{n_i}_-, \end{cases} \qquad \psi_i^*(v_i) = \begin{cases} 0 & \text{if } v_i \in I\!R^{m_i}_+, \\ \infty & \text{if } v_i \notin I\!R^{m_i}_+. \end{cases}$$

This model belongs to the piecewise linear-quadratic case, where the theorems are at their best. A much broader model in that category is obtained by choosing

$$\varphi_i(u_i) = \begin{cases} \frac{1}{2} u_i \cdot P_i u_i & \text{if } u_i \in U_i, \\ \infty & \text{if } u_i \notin U_i, \end{cases} \qquad \psi_i(z_i) = \theta_{V_i,Q_i}(z_i),$$

for nonempty polyhedral sets U_i and V_i and symmetric, positive *semi*definite matrices P_i and Q_i, with θ_{V_i,Q_i} defined as in (3.5). In dualizing one then has

$$\psi_i^*(v_i) = \begin{cases} \frac{1}{2} v_i \cdot Q_i v_i & \text{if } v_i \in V_i, \\ \infty & \text{if } v_i \notin V_i, \end{cases} \qquad \varphi_i^*(w_i) = \theta_{U_i,P_i}(w_i).$$

This is stochastic *piecewise linear-quadratic* programming.

5. Optimality Conditions and Decomposition

The duality in Theorem 6 has interesting implications for optimality conditions in multistage stochastic programming and how such conditions might be employed in computation.

Theorem 7 (saddle points in stochastic programming)**.**
(a) *In the circumstances of Theorem 6 where* $\inf(\mathcal{P}^+) = \sup(\mathcal{D}^+)$, *one has that*

$$\left.\begin{array}{r} (\bar{x},\bar{u};\bar{y},\bar{v}) \text{ is a saddle point} \\ \text{of } L^+ \text{ on } [I\!R^d \times U] \times [I\!R^d \times V] \end{array}\right\} \iff \begin{cases} (\bar{x},\bar{u}) \text{ solves } (\mathcal{P}^+), \\ (\bar{y},\bar{v}) \text{ solves } (\mathcal{D}^+). \end{cases}$$

(b) *In the same circumstances, where equally* $\inf(\mathcal{P}^-) = \sup(\mathcal{D}^-)$, *one has that*

$$\left.\begin{array}{r} (\bar{u},\bar{v}) \text{ is a saddle point} \\ \text{of } L^- \text{ on } U \times V \end{array}\right\} \iff \begin{cases} \bar{u} \text{ solves } (\mathcal{P}^-), \\ \bar{v} \text{ solves } (\mathcal{D}^-). \end{cases}$$

(c) *The two saddle point conditions are both equivalent to the following subgradient properties being satisfied:*

$$\left.\begin{aligned} E_i\{B^*_{i_+}\bar{y}_{i_+}\} + D^*_i\bar{v}_i - p_i \;&\in\; \partial\varphi_i(\bar{u}_i) \\ q_i - C_i\bar{x}_i - D_i\bar{u}_i \;&\in\; \partial\psi^*_i(\bar{v}_i) \end{aligned}\right\} \text{for } i \in I, \text{ with } \bar{x} = X(\bar{u}),\ \bar{y} = Y(\bar{v}). \tag{5.1}$$

Proof. This comes from Theorem 3 as applied by way of Theorems 4 and 5. The conditions in (5.1) fall directly out of the saddle point condition for L^+, namely

$$\begin{aligned} (\bar{x},\bar{u}) \in &\mathbb{R}^d \times U,\ \ (\bar{y},\bar{v}) \in \mathbb{R}^d \times V,\ \text{and} \\ &L^+(x,u;\bar{y},\bar{v}) \geq L^+(\bar{x},\bar{u};\bar{y},\bar{v}) \geq L^+(\bar{x},\bar{u};y,v) \\ &\qquad \text{for all}\ \ (x,u) \in \mathbb{R}^d \times U,\ \ (y,v) \in \mathbb{R}^d \times V. \end{aligned}$$

The maximization of $L^+(\bar{x},\bar{u};y,v)$ in (y,v) can be seen from the formula for L^+ in (4.8) to come down to separate maximizations in the components y_i and v_i. This yields the second set of subgradient relations along with $\bar{x} = X(\bar{u})$. Likewise, by substituting the alternative expression in (4.14) into the formula (4.8), one sees that the minimization of $L^+(x,u;\bar{y},\bar{v})$ in (x,u) corresponds to separate minimizations in x_i and u_i, which furnish the second set of subgradient conditions along with $\bar{y} = Y(\bar{v})$. The fact that (5.1) also describes saddle points $(\bar{u},\bar{u})$ of L^- is evident from (4.6) and (4.17). □

To put a good face on the subgradient conditions in (5.1), we introduce now—in terms of any $\bar{x}$ and $\bar{y}$ in $\mathbb{R}^d$, acting as parameter elements—the vectors

$$\bar{p}_i := p_i - E_i\{B^*_{i_+}\bar{y}_{i_+}\}, \quad \bar{q}_i := q_i - C_i\bar{x}_i, \tag{5.2}$$

and an associated family of subproblems, one for each information state $i \in I$:

$$(\bar{\mathcal{P}}_i) \qquad \text{minimize}\ \ \bar{p}_i{\cdot}u_i + \varphi_i(u_i) + \psi_i(\bar{q}_i - D_iu_i)\ \ \text{in}\ \ u_i.$$

The dual problem paired with $(\bar{\mathcal{P}}_i)$ in the extended Fenchel format is

$$(\bar{\mathcal{D}}_i) \qquad \text{maximize}\ \ \bar{q}_i{\cdot}v_i - \psi^*_i(v_i) - \varphi^*_i(D^*_iv_i - \bar{q}_i)\ \ \text{in}\ \ v_i,$$

and the corresponding Lagrangian is

$$\bar{L}_i(u_i,v_i) := \bar{p}_i{\cdot}u_i + \varphi_i(u_i) + \bar{q}_i{\cdot}v_i - \psi_i(v_i) - v_i{\cdot}D_iu_i\ \ \text{on}\ \ U_i \times V_i. \tag{5.3}$$

The facts in Theorems 2 and 3 are available for these problems.

Theorem 8 (problem decomposition by information states)**.** *The optimality conditions in Theorem 7 have the following interpretation:*

$$\begin{cases} \bar{x} = X(\bar{u}),\ \ \bar{y} = Y(\bar{v}),\ \ \text{and for each } i \in I \\ (\bar{u}_i, \bar{v}_i)\ \ \text{is a saddle point of } \bar{L}_i \text{ on } U_i \times V_i. \end{cases} \tag{5.4}$$

Thus, in the piecewise linear-quadratic case in particular, $\bar{u} = \{\bar{u}_i\}$ solves the reduced problem $(\mathcal{P}^-)$ if and only if there exists $\bar{v} = \{\bar{v}_i\}_{i \in I}$ such that, when $\bar{x} = \{\bar{x}_i\}_{i \in I}$ and $\bar{y} = \{\bar{y}_i\}_{i \in I}$ are taken as the trajectories of primal and dual states generated from these controls by the dynamics in (4.2) and (4.15), it turns out that, for every $i \in I$,

$$\begin{cases} \bar{u}_i \ \text{is an optimal solution to } (\bar{\mathcal{P}}_i), \\ \bar{v}_i \ \text{is an optimal solution to } (\bar{\mathcal{D}}_i). \end{cases}$$

Proof. All we need to observe is that the subgradient conditions in (5.1) are the ones furnished by Theorem 3, as specialized to the problems $(\bar{\mathcal{P}}_i)$ and $(\bar{\mathcal{D}}_i)$. In the piecewise linear-quadratic case, saddle points correspond always to pairs of optimal solutions, as we know from Theorem 2. □

In practical terms, this decomposition result is especially suited to algorithms that in some way alternate between, on the one hand, integrating linear dynamical systems to get states from controls and, on the other hand, solving collections of problems $(\bar{\mathcal{P}}_i)$, perhaps in parallel for various information states $i \in I$. Backward-forward splitting algorithms have just this character; cf. [8]. Other splitting methods can likewise exploit the special Lagrangian structure in (4.23) without relying necessarily on repeated integration of the dynamics; cf. [5] and [6]. It usually wouldn't be required, of course, to solve the corresponding dual problems $(\bar{\mathcal{D}}_i)$ directly, since almost any method for solving $(\bar{\mathcal{P}}_i)$ to get $\bar{u}_i$ would automatically produce $\bar{v}_i$ as an associated multiplier vector.

References

1. R. T. Rockafellar and R. J-B Wets, "Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time," SIAM J. Control Opt. 28 (1990), 810–822.
2. L. Korf, "Approximation and solution schemes for dynamic stochastic optimization problems," Ph.D. dissertation, Dept. of Mathematics, University of California at Davis, 1998.
3. R. T. Rockafellar, "Linear-quadratic programming and optimal control," SIAM J. Control and Opt. 25 (1987), 781–814.

4. R. T. Rockafellar, "Multistage convex programming and discrete-time optimal control," Control and Cybernetics 17 (1988), 225–246.

5. D. H. Salinger, *A Splitting Algorithm for Multistage Stochastic Programming with Application to Hydropower Scheduling*, Ph.D. dissertation, Dept. of Applied Math., University of Washington, 1997.

6. J. Eckstein and M. C. Ferris, "Operator splitting methods for monotone affine variational inequalities with a parallel application to optimal control," INFORMS J. on Computing 10 (1998), No. 2.

7. G. H.-G. Chen, *Forward-Backward Splitting Techniques: Theory and Applications*, PhD dissertation, Dept. of Applied Math., University of Washington, 1994.

8. G. H.-G. Chen and R. T. Rockafellar, "Convergence rates in forward-backward splitting," SIAM J. Optim. 7 (1997), 421–444.

9. R. T. Rockafellar and R. J-B Wets, *Variational Analysis*, Grundlehren der Mathematischen Wissenschaften 317, Springer-Verlag, 1997.

10. R. T. Rockafellar, "Duality theorems for convex functions," Bull. Amer. Math. Soc. 70 (1964), 189-192.

11. R. T. Rockafellar and R. J-B Wets, "A Lagrangian finite generation technique for solving linear-quadratic problems in stochastic programming," Math. Programming Studies 28 (1986), 63–93.

12. R. T. Rockafellar and R. J-B Wets, "Linear-quadratic problems with stochastic penalties: the finite generation algorithm," in *Stochastic Optimization*, V. I. Arkin, A. Shiraev and R. J-B Wets (eds.), Springer-Verlag Lecture Notes in Control and Information Sciences No. 81, 1987, 545-560.

13. R. T. Rockafellar, "Duality and stability for extremum problems involving convex functions," Pacific J. Math. 21 (1967), 167-187.

14. R. T. Rockafellar, "Lagrange multipliers and optimality," SIAM Review 35 (1993), 183–238.