Chapter 2
Background Theory

We start with a review of deterministic optimization problems in temporal networks. We then discuss three popular methodologies to model and solve generic optimization problems under uncertainty. We close with an overview of the issues that arise when these methodologies are applied to temporal networks, and we provide a survey of the relevant literature. More specific reviews of related work are provided in the Chaps. 3–6.

2.1 Temporal Networks

The literature on temporal networks is vast and has been reviewed, amongst others, in [BDM$^+$99, BEP$^+$96, BKPH05, Bru07, DH02, FL04, NSZ03, Pin08, Sch05]. Instead of giving a detailed account of all contributions, we classify some of the most popular research directions according to the three dimensions "resources", "network" and "objective". More elaborate classification schemes can be found in [BDM$^+$99, Bru07, DH02].

Resource characteristics: Optimization problems in temporal networks may assume that a resource allocation has been fixed, or they can involve the assignment of one or multiple resources. In the latter case, we can distinguish between three prevalent types of resources. *Non-renewable resources* are available in pre-specified quantities and are not replenished during the planning horizon. Typical examples of non-renewable resources are capital and man-hours. In contrast, *renewable resources* are replenished every time period, but the decision maker has to meet specified per-period consumption quotas. Examples of renewable resources are processing times on manufacturing machines and processors. In practice, many resources are *doubly constrained*, that is, they share the restrictions of non-renewable and renewable resources. Other resource characteristics include time windows during which the resources are available, as well as spatial aspects (e.g., immobile resources such as a shipyard).

W. Wiesemann, *Optimization of Temporal Networks under Uncertainty*, Advances in Computational Management Science 10,
DOI 10.1007/978-3-642-23427-9_2,

Network characteristics: Network characteristics describe the properties of the network tasks and precedences. Tasks are *preemptive* if their processing can be interrupted to execute other tasks. For example, modern operating systems use preemptive multitasking to generate the illusion of executing multiple computer applications in parallel on a single processor. If the execution of network tasks must not be interrupted, then the tasks are called *non-preemptive*. Project scheduling, circuit design and many problems in process scheduling assume that the network tasks are non-preemptive. In the introduction, we assumed that all precedences in the temporal network are of *finish-start* type, that is, an arc from node i to node j in the temporal network prescribes that task j cannot be started before task i has been completed. Alternatively, one can consider *generalized precedences* that stipulate lower and upper bounds on the time that may pass between the start and completion of any two network tasks. Other network characteristics include time windows during which the tasks must be executed (e.g., ready times and deadlines) and cash flows that arise when the tasks are processed.

Objective function: One commonly distinguishes between *regular objective functions*, which are optimized by the early start schedule (1.2), and *nonregular objective functions*, which may not be optimized by the early start schedule. Typical regular objective functions are the makespan and the lateness of the makespan beyond a given deadline. Examples of nonregular objectives are the net present value and a level resource consumption.

The methods studied in this book are primarily applicable to temporal networks with non-preemptive tasks and non-renewable resources. Chapter 3 assumes that the resource allocation is fixed and maximizes the expected net present value under generalized precedences. In Chap. 4–6 we determine assignments of non-renewable resources under finish-start precedences. Chapter 4 studies a multi-objective problem that considers discrete resources (web services) and optimizes the conditional value-at-risk of the makespan and resource costs. Chapters 5 and 6 assume continuous resources (e.g., capital or man-hours). Chapter 5 optimizes quantiles of the makespan, whereas Chap. 6 minimizes the worst-case makespan.

2.2 Optimization Under Uncertainty

In practice, most managerial decisions are taken under significant uncertainty about relevant data such as future market developments and resource availabilities. If such decision problems are formulated as optimization models, the models contain parameters whose values are uncertain. In the following, we review three popular approaches to model and solve optimization problems with uncertain parameters. In the remainder of the book, we will present applications of two of these approaches to optimization problems in temporal networks.

Fig. 2.1 Temporal structure of a two-stage (*left*) and a multi-stage (*right*) recourse problem. In the *left time line*, the wait-and-see decision y may depend on x and ξ. In the *right time line*, the wait-and-see decision y^t may depend on x and ξ^s, $s < t$

2.2.1 Stochastic Programming

Stochastic programming models the uncertain problem parameters as random variables with known probability distributions. One of the basic models is the *two-stage recourse problem.*

$$\inf_{x \in X} \{f(x) + \mathbb{E}[Q(x;\xi)]\}, \tag{2.1a}$$

where

$$Q(x;\xi) = \inf_{y \in Y(x,\xi)} \{q(y;x,\xi)\}. \tag{2.1b}$$

In this problem, the parameter vector ξ is assumed to be uncertain. The decision maker needs to take a *here-and-now decision* $x \in X$ before the value of ξ is known, while the *wait-and-see decision* $y \in Y(x,\xi)$ can be selected under full knowledge of ξ. Conceptually, we can assume that x is chosen at the beginning of time period 1, ξ is revealed during time period 1, and y is selected at the beginning of time period 2 (after ξ is known), see Fig. 2.1, left. The goal is to minimize the sum of deterministic first-stage costs $f(x)$ and expected second-stage costs $\mathbb{E}[Q(x;\xi)]$, where the expectation is taken with respect to ξ. Note that for any value of x and ξ, the second-stage problem $Q(x;\xi)$ is deterministic. If there is a finite set of values $\xi^1, \xi^2, \ldots$ such that $\xi \in \{\xi^1, \xi^2, \ldots\}$ with probability one, then (2.1) can be formulated as an explicit optimization problem. Otherwise, (2.1) can be approximated by a surrogate model that replaces the probability distribution of ξ with a finite-valued approximation. In either case, the resulting optimization model has the structure of a *scenario fan* whose branches represent the possible realizations of ξ, see Fig. 2.2, left. The decision maker's information set (i.e., the set of scenarios that may be realized) is shown in Fig. 2.2, right. At the beginning of the first time period, the decision maker is unaware of the realized scenario ξ^k. The information set therefore contains all scenarios. In the second time period, the decision maker knows the realized scenario ξ^k. The information set has therefore shrunk to one of the singleton sets $\{\xi_1\}, \ldots, \{\xi_5\}$.

The use of the expected value in (2.1a) reflects the assumption that the decision maker is *risk-neutral*. In many applications of temporal networks, such as project management and production scheduling, this assumption may not hold. Instead,

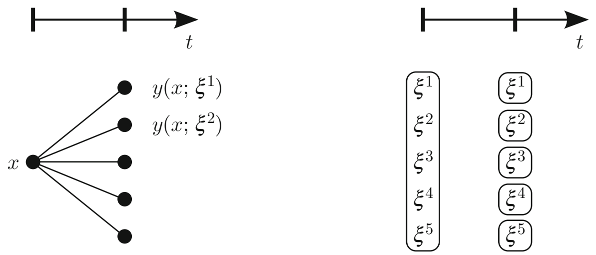

Fig. 2.2 Scenario representation of two-stage recourse problems. The *left chart* shows that for each realization (scenario) ξ^k of the random vector ξ, a separate recourse decision $y(x;\xi^k)$ can be selected. The *right chart* visualizes the acquisition of information over time

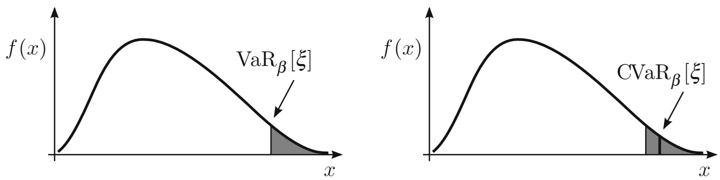

Fig. 2.3 Value-at-risk (*left*) and conditional value-at-risk (*right*) of a continuous random variable ξ with probability density function $f(x)$

the decision maker is *risk-averse* and prefers solutions that do not just perform well "on average", but that also perform satisfactory "in most cases". The most widely used approach to obtain risk-averse solutions is to minimize the variance of $Q(x;\xi)$, which traces back to the seminal paper [Mar52] on financial portfolio selection. However, minimizing the variance of $Q(x;\xi)$ penalizes both the excess and the shortfall of $Q(x;\xi)$ with respect to its expected value $\mathbb{E}[Q(x;\xi)]$. This may not be appropriate for optimization problems in temporal networks. If the goal is to minimize the makespan, for example, a decision maker only wants to penalize upward deviations from the expected value (i.e., delays), whereas downward deviations are indeed desirable. A decision maker may therefore prefer to optimize a one-sided quantile-based risk measure such as the value-at-risk (VaR):

$$\mathrm{VaR}_\beta\,[Q(x;\xi)] = \inf\{\alpha \in \mathbb{R} : \mathbb{P}\,(Q(x;\xi) > \alpha) \leq 1 - \beta\}.$$

For ease of exposition, we assume for the remainder of this section that the random parameters ξ follow a continuous probability distribution. For $\beta \in [0,1]$, we can then interpret the β-VaR of $Q(x;\xi)$ as the β-quantile of $Q(x;\xi)$. For high values of β (e.g., $\beta \geq 0.9$), minimizing the β-VaR of $Q(x;\xi)$ favors solutions that perform well in most cases. The VaR of a random variable ξ is illustrated in Fig. 2.3, left.

In recent years, VaR has come under criticism due to its nonconvexity, which makes the resulting optimization models difficult to solve. Moreover, the nonconvexity implies that VaR is not sub-additive and hence not a coherent risk measure in

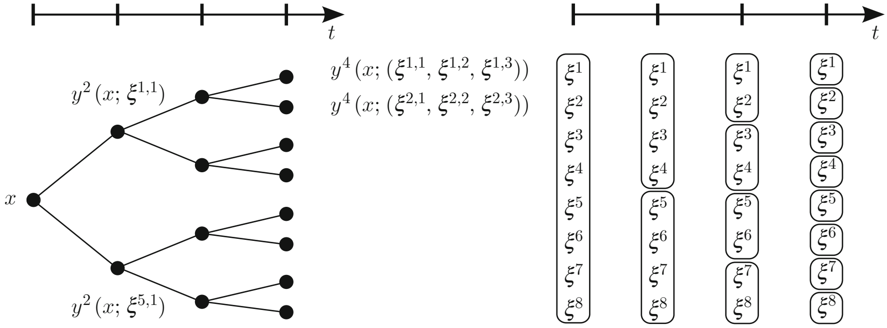

Fig. 2.4 Scenario representation of multi-stage recourse problems. In analogy to the scenario fan in Fig. 2.2, left, the *left chart* visualizes the scenario tree associated with a multi-stage recourse problem, whereas the *right chart* shows the acquisition of information over time

the sense of [ADEH99]. Both drawbacks are rectified by the conditional value-at-risk (CVaR), which is defined as follows:

$$\mathrm{CVaR}_\beta\,[Q(x;\xi)] = \mathbb{E}\left(Q(x;\xi) \mid Q(x;\xi) \geq \mathrm{VaR}_\beta\,[Q(x;\xi)]\right).$$

The β-CVaR of $Q(x;\xi)$ represents the expected value of $Q(x;\xi)$ under the assumption that $Q(x;\xi)$ exceeds its β-VaR, that is, under the assumption that $Q(x;\xi)$ is among the $(1-\beta)\cdot 100\%$ "worst" outcomes. By definition, the β-CVaR exceeds the β-VaR for any $\beta \in [0,1]$. The CVaR of a random variable ξ is illustrated in Fig. 2.3, right. It has been shown in [RU00] that the β-CVaR is equivalent to

$$\inf_{\alpha\in\mathbb{R}} \left\{ \alpha + \frac{1}{1-\beta}\mathbb{E}\,[Q(x;\xi)-\alpha]^+ \right\},$$

where $[x]^+ = \max\{x,0\}$. Hence, the techniques presented for recourse problems with expected value objective functions are directly applicable to optimization problems involving CVaR.

So far, all of the models assumed a two-stage structure (decision – realization of uncertainty – decision). In a *multi-stage recourse problem*, the parameter vector ξ can be subdivided into vectors $\xi^1,\ldots,\xi^T$ such that $\xi = (\xi^1,\ldots,\xi^T)$ and ξ^t is revealed during time period $t = 1,\ldots,T$. The decision maker can take a recourse decision y^t at the beginning of every time period $t = 2,\ldots,T+1$, and y^t may depend on the values of $\xi^1,\ldots,\xi^{t-1}$, see Fig. 2.1, right. Note that y^t may *not* depend on the values of ξ^s, $s \geq t$, since this information is not available at the time the recourse decision y^t is taken. This causality requirement is called *non-anticipativity*. If the probability distribution of ξ has finitely many values, then the optimization model associated with a multi-stage recourse problem has the structure of a *scenario tree*, see Fig. 2.4. In the left chart of that figure, $\xi^{k,t}$ denotes the tth subvector of the scenario $\xi^k = (\xi^{k,1},\ldots,\xi^{k,T})$. Each path from the root node to a leaf node constitutes one scenario. Two scenarios ξ^k and ξ^l are undistinguishable at the beginning of period t if $\xi^{k,s} = \xi^{l,s}$ for all $s < t$. In this case, ξ^k and ξ^l are contained in the same information set at time t, and non-anticipativity stipulates that $y^t(x; (\xi^{k,1}, \ldots, \xi^{k,t-1})) = y^t(x; (\xi^{l,1}, \ldots, \xi^{l,t-1}))$. For example, non-anticipativity requires that $y^2(x; \xi^{k,1}) = y^2(x; \xi^{l,1})$ for $k, l \in \{1, \ldots, 4\}$ and $y^3(x; (\xi^{5,1}, \xi^{5,2})) = y^3(x; (\xi^{6,1}, \xi^{6,2}))$. In analogy to two-stage recourse problems, a multi-stage recourse problem can be approximated by a surrogate model that replaces the probability distribution of ξ with a finite-valued approximation if ξ can attain infinitely many values. While convex two-stage recourse problems can be approximated efficiently, multi-stage problems "generically are computationally intractable already when medium-accuracy solutions are sought" [SN05]. Multi-stage recourse problems therefore constitute very difficult optimization problems.

Apart from recourse problems, stochastic programming studies problems with *chance constraints*. The basic two-stage chance constrained problem can be formulated as follows:

$$\inf_{x \in X} \{ f(x) \;:\; \mathbb{P}\left(Q(x; \xi) \leq 0\right) \geq 1 - \epsilon \}, \tag{2.2}$$

where Q is defined in (2.1b). The temporal structure of problem (2.2) is the same as for two-stage recourse problems, see Fig. 2.1, left. The goal is to find a here-and-now decision x such that with a probability of at least $1 - \epsilon$, there is a wait-and-see decision $y(x; \xi) \in Y(x, \xi)$ that satisfies $q(y(x; \xi); x, \xi) \leq 0$. Note that problem (2.2) is equivalent to the following problem with a VaR constraint:

$$\inf_{x \in X} \{ f(x) \;:\; \text{VaR}_{1-\epsilon}\left[Q(x, \xi)\right] \leq 0 \}. \tag{2.2'}$$

It is therefore not surprising that chance constrained problems inherit the computational difficulties of VaR optimization problems. Indeed, even if the second-stage problem Q is a linear program, the feasible region of (2.2) is typically nonconvex and disconnected [Pré95]. Moreover, calculating the left-hand side of the constraint in (2.2) requires the evaluation of a multi-dimensional integral, which itself constitutes a difficult problem. As a result, most solution approaches for (2.2) settle for approximate solutions. A popular approximation is obtained by replacing the $(1 - \epsilon)$-VaR in (2.2') with the $(1 - \epsilon)$-CVaR:

$$\inf_{x \in X} \{ f(x) \;:\; \text{CVaR}_{1-\epsilon}\left[Q(x, \xi)\right] \leq 0 \}. \tag{2.2''}$$

Since CVaR represents an upper bound on VaR, this formulation provides a conservative approximation to problem (2.2'), that is, any $x \in X$ that is feasible in (2.2") is also feasible in (2.2') and (2.2). Similar to recourse problems, chance constrained problems can be extended to multiple decision stages.

For an in-depth treatment of stochastic programming, the reader is referred to the textbooks [KW94, Pré95, RS03]. We will consider two-stage recourse problems that optimize the expected value and CVaR in Chaps. 3 and 4, respectively. Chapter 5 studies an approximation of VaR that does not rely on scenario discretization.

2.2.2 Robust Optimization

In its basic form, robust optimization studies semi-infinite problems of the following type:

$$\inf_{x \in X} \{ f(x) \; : \; g_i(x; \xi) \leq 0 \;\; \forall \, \xi \in \Xi, \; i = 1, \ldots, I \}. \tag{2.3}$$

We interpret x as a here-and-now decision and ξ as an uncertain parameter vector with support Ξ. The goal is to minimize the deterministic costs $f(x)$ while satisfying the constraints for all possible realizations of ξ. Note that (2.3) is a single-stage problem since it does not contain any recourse decisions. If Ξ constitutes a finite set of scenarios $\xi^1, \xi^2, \ldots$, then (2.3) can be formulated as an explicit optimization problem. If Ξ is of infinite cardinality, then (2.3) can be solved with iterative solution procedures from semi-infinite optimization [HK93, RR98]. One of the key contributions of robust optimization has been to show that for sets Ξ of infinite cardinality but specific structure, one can apply duality theory to transform problem (2.3) into an explicit optimization problem. We illustrate this approach with an example.

Example 2.2.1. Assume that $I = 1$, $X \subseteq \mathbb{R}^n$, $\Xi = \left\{ \xi \in \mathbb{R}^k_+ \; : \; W\xi \leq h \right\}$ for $W \in \mathbb{R}^{m \times k}$ and $h \in \mathbb{R}^m$, and $g_1(x; \xi) = \xi^\top A \, x$ for $A \in \mathbb{R}^{k \times n}$. Also assume that Ξ is nonempty and bounded. We can then reformulate the constraint in problem (2.3) as follows:

$$\begin{aligned}
g_1(x;\xi) \leq 0 \quad \forall \, \xi \in \Xi \quad &\Leftrightarrow \quad \sup_{\xi \in \Xi} \{ g_1(x; \xi) \} \leq 0 \\
&\Leftrightarrow \quad \max_{\xi \in \mathbb{R}^k_+} \left\{ \xi^\top A \, x \; : \; W\xi \leq h \right\} \leq 0 \\
&\Leftrightarrow \quad \min_{\lambda \in \mathbb{R}^m_+} \left\{ h^\top \lambda \; : \; W^\top \lambda \geq Ax \right\} \leq 0 \\
&\Leftrightarrow \quad h^\top \lambda \leq 0, \; W^\top \lambda \geq Ax \qquad \text{for some } \lambda \in \mathbb{R}^m_+.
\end{aligned}$$

Here, the third equivalence follows from strong linear programming duality. We have thus transformed the semi-infinite constraint in problem (2.3) into a finite number of constraints that involve x and new auxiliary variables λ.

Much of the early work on robust optimization focuses on generalizations of the reformulation scheme illustrated in Example 2.2.1. Unfortunately, single-stage models such as (2.3) are too restrictive for decision problems in temporal networks. Indeed, the task start times can typically be chosen as a wait-and-see decision, and optimization problems that account for this flexibility provide significantly better solutions. We discuss this issue in more detail in Chaps. 3–6. We are therefore interested in *two-stage robust optimization problems* such as the following one:

$$\inf_{x \in X} \sup_{\xi \in \Xi} \inf_{y \in Y(x,\xi)} \{f(x) + q(y; x, \xi)\}. \tag{2.4}$$

Here, q is the objective function of the second-stage problem Q defined in (2.1b), and $Y(x, \xi) \subseteq \mathbb{R}^l$. In this problem, the here-and-now decision x is accompanied by a wait-and-see decision $y \in Y(x, \xi)$ that can be selected under full knowledge of ξ. The temporal structure of this problem is similar to the two-stage recourse problem (2.1), see Fig. 2.1, left. The goal is to minimize the sum of first-stage costs $f(x)$ and worst-case second-stage costs $\sup_{\xi \in \Xi} Q(x; \xi)$, see (2.1b), where the worst case is taken with respect to ξ. Two-stage robust optimization problems are generically intractable, see [BTGGN04]. A tractable approximation can be derived from the following identity.

Observation 2.2.1 *For any $X \subseteq \mathbb{R}^n$, $\Xi \subseteq \mathbb{R}^k$ and $Y(x, \xi) \subseteq \mathbb{R}^l$, we have*

$$\inf_{x \in X} \sup_{\xi \in \Xi} \inf_{y \in Y(x,\xi)} \{f(x) + q(y; x, \xi)\} = \inf_{\substack{x \in X,\\ y \in \mathcal{Y}(x)}} \sup_{\xi \in \Xi} \{f(x) + q(y(\xi); x, \xi)\}, \tag{2.5a}$$

where for $x \in X$,

$$\mathcal{Y}(x) = \left\{(y : \Xi \mapsto \mathbb{R}^l) : y(\xi) \in Y(x, \xi) \quad \forall \xi \in \Xi\right\}. \tag{2.5b}$$

Observation 2.2.1 allows us to reduce the min–max–min problem (2.4) to the min–max problem on the right-hand side of (2.5a) at the cost of augmenting the set of first-stage decisions. For a given here-and-now decision $x \in X$, $\mathcal{Y}(x)$ denotes the space of all functions on Ξ that map parameter realizations to feasible wait-and-see decisions. A function y is called a *decision rule* because it specifies the second-stage decision in (2.4) as a function of the uncertain parameters ξ. Note that the choice of an appropriate decision rule on the right-hand side of (2.5a) is part of the first-stage decision. The identity (2.5a) holds regardless of the properties of X and Ξ because $\mathcal{Y}(x)$ does not impose any structure on the decision rules (such as measurability).

Since $\mathcal{Y}(x)$ constitutes a function space, further assumptions are required to ensure that the problem on the right-hand side of (2.5a) can be solved. A popular approach is to restrict $\mathcal{Y}(x)$ to the space of affine or piecewise affine functions of ξ, see [BTGN09, CSSZ08, KWG]. As we will show in Chap. 6, this restriction allows us to reformulate the model on the right-hand side of (2.5a) as an explicit optimization problem. Figure 2.5 compares the scenario approximation from the previous section with the decision rule approximation. In the left chart of that figure, the support Ξ of the random parameters ξ is replaced with a discrete-valued approximation. For each possible realization (scenario) ξ^k, an individual second-stage decision $y(x; \xi^k)$ may be chosen. In the right chart of Fig. 2.5, the

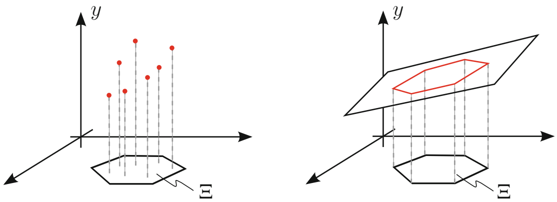

Fig. 2.5 Approximations employed by two-stage recourse problems (*left*) and two-stage robust optimization problems (*right*) for a random vector ξ with a continuous probability distribution

support Ξ of the random parameters ξ remains unchanged, but the second-stage decision $y(x;\xi)$ is restricted to be an affine function of ξ.

For an introduction to robust optimization, see [BS04, BTGN09]. Two-stage robust optimization problems are discussed in [BTGN09, CSSZ08, JLF07, LJF04, LLMS09, Sti09]. In recent years, the theory of robust optimization has been extended to recourse problems and chance constrained problems. For further details, see [BP05, BTGN09, CSST10, DY10, GS10]. In Chap. 6 we will solve a makespan minimization problem as a two-stage robust optimization problem. Instead of approximating the optimal second-stage decision via decision rules, however, this chapter presents a technique that provides convergent lower and upper bounds on the optimal value of the problem. The upper bounds correspond to feasible solutions whose objective values are bracketed by the bounds. We will compare that method with two popular classes of decision rules.

2.2.3 Stochastic Dynamic Programming

Stochastic dynamic programming studies the modeling and solution of optimization problems via *Markov decision processes* (MDPs). MDPs model dynamic decision problems in which the outcomes are partly random and partly under the control of the decision maker. At each time period, the MDP is in some state s, and the decision maker takes an action a. The state s' in the successive time period is random and depends on both the current state s and the selected action a. However, the new state does *not* depend on any other past states or actions: this is the *Markov property*. For each transition of the MDP, the decision maker receives a reward that depends on the old state, the new state and the action that triggered the transition.

In the following, we will restrict ourselves to discrete-time MDPs with finite state and action spaces. We therefore assume that an MDP is defined through its state space $\mathcal{S} = \{1, \ldots, S\}$, its action space $\mathcal{A} = \{1, \ldots, A\}$, and a discrete planning horizon $\mathcal{T} = \{0, 1, 2, \ldots\}$ that can be finite or infinite. The initial state is a random variable with known probability distribution p_0. If action $a \in \mathcal{A}$ is chosen in state $s \in \mathcal{S}$, then the subsequent state is $s' \in \mathcal{S}$ with probability $p(s'|s,a)$. We assume that the probabilities $p(s'|s,a)$, $s' \in \mathcal{S}$, sum up to one for each state–action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$. The decision maker receives an expected reward of $r(s,a,s') \in \mathbb{R}$ if action $a \in \mathcal{A}$ is chosen in state $s \in \mathcal{S}$ and the subsequent state is $s' \in \mathcal{S}$. Without loss of generality, we can assume that every action is admissible in every state. Indeed, if action $a \in \mathcal{A}$ is not allowed in state $s \in \mathcal{S}$, then we can "forbid" this action by setting all rewards $r(s, a, s')$, $s' \in \mathcal{S}$, to a large negative value. Figure 2.6 visualizes the structure of an MDP.

At each stage, the MDP is controlled through a policy $\pi = (\pi_t)_{t \in \mathcal{T}}$, where $\pi_t(a|s_0, a_0, \ldots, s_{t-1}, a_{t-1}; s_t)$ represents the probability to choose action $a \in \mathcal{A}$ if the current state is s_t and the state-action history is given by the vector $(s_0, a_0, \ldots, s_{t-1}, a_{t-1})$. Note that contrary to the state transitions of the MDP, the policy π need not be Markovian. If the planning horizon $\mathcal{T}$ is infinite, then we can evaluate a policy π in view of its *expected total reward* under the discount factor $\lambda \in (0, 1)$:

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \lambda^t r(s_t, a_t, s_{t+1}) \,\middle|\, s_0 \sim p_0\right]. \tag{2.6}$$

Here, $\mathbb{E}$ denotes the expectation with respect to the random process defined by the transition probabilities p and the policy π. The notation $s_0 \sim p_0$ indicates that the initial state s_0 is a random variable with probability distribution p_0. If the planning horizon $\mathcal{T}$ is finite, say $\mathcal{T} = \{0, 1, \ldots, T\}$, then we can evaluate a policy π in view of its *expected total reward* without discounting:

$$\mathbb{E}\left[\sum_{t=0}^{T} r(s_t, a_t, s_{t+1}) \,\middle|\, s_0 \sim p_0\right]. \tag{2.7}$$

For a fixed policy π, the *policy evaluation problem* asks for the value of expression (2.6) or (2.7). The *policy improvement problem*, on the other hand, asks for a policy π that maximizes (2.6) or (2.7). For both objective functions, the policy evaluation and improvement problems can be solved efficiently via policy and value iteration.

Example 2.2.2 (Inventory Management). Consider the following infinite horizon inventory problem. At the beginning of each time period, the decision maker can order $a \in \mathbb{N}_0$ units of a product at unit costs c. The ordered products arrive at the beginning of the next period. During each period, an independent and identically distributed random demand δ arises for the product. This demand is served at a unit

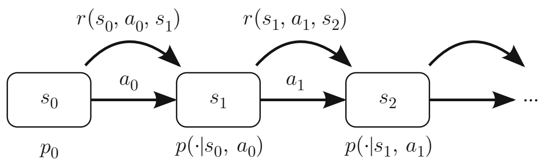

Fig. 2.6 Structure of a Markov decision process. The process starts in state $s_0 \in \mathcal{S}$, which follows the probability distribution p_0. After the action $a_0 \in \mathcal{A}$ is chosen, the new state $s_1 \in \mathcal{S}$ follows the probability distribution $p(\cdot|s_0, a_0)$, and an expected reward $r(s_0, a_0, s_1)$ is received

price p from the current inventory, and there is no backlogging (i.e., demand that cannot be satisfied within the period is lost). The inventory can hold at most I units of the product. The goal is to find an inventory control policy that maximizes the expected total reward under some discount factor λ.

We can formulate this problem as an infinite horizon MDP as follows. The state set $\mathcal{S} = \{0, \ldots, I\}$ describes the inventory level at the beginning of each time period. In state $s \in \mathcal{S}$, the admissible actions $\{0, \ldots, I - s\}$ determine the order quantity. Note that the actions are state-dependent in this example. The transition probabilities are

$$p(s'|s,a) = \begin{cases} \mathbb{P}(\delta = s + a - s') & \text{if } s' \neq 0, \\ \sum_{i=s+a}^{\infty} \mathbb{P}(\delta = i) & \text{otherwise,} \end{cases}$$

and the rewards are given by $r(s,a,s') = p(s + a - s') - ca$. Here we assume that the random demand δ is nonnegative with probability one. A policy π could order $\omega \in \mathbb{N}$ units whenever the current inventory falls below some threshold $\Omega \in \mathbb{N}_0$. This policy is defined through $\pi_t(a|s_0, a_0, \ldots, s_{t-1}, a_{t-1}; s_t) = 1$ if $s_t < \Omega$ and $a = \omega$, or $s_t \geq \Omega$ and $a = 0$, and $\pi_t(a|s_0, a_0, \ldots, s_{t-1}, a_{t-1}; s_t) = 0$ otherwise. Note that this policy π is Markovian.

There are numerous variations of the Markov decision process defined in this section. For an overview of the major models and solution approaches, see [Ber07, Put94].

In recent years, an approximation scheme called "approximate dynamic programming" has received much attention. The interested reader is referred to the textbooks [BT96, Pow07]. In this book, we will not consider the application of Markov decision processes to temporal networks. The reader is referred to [BR97, KA86, TSS06] and there references therein.

2.3 Optimization of Temporal Networks under Uncertainty

Decisions in temporal networks are often taken under significant uncertainty about the network structure (i.e., the tasks and precedences of the network), the task durations, the ready times and deadlines of the tasks, the cash flows and the availability of resources. In this book, we focus on problems in which the task durations (Chaps. 3–6), the cash flows (Chaps. 3 and 4), the network structure (Chap. 4), and the tasks' ready times and deadlines (Chap. 3) are uncertain. A problem that accounts for uncertain resource availabilities is described in [Yan05]. Further reviews on problems with uncertain network structure are provided in [Neu79, Neu99, Pri66].

An optimization problem under uncertainty needs to specify *when* information about the uncertain parameters becomes available, and *what* information is revealed about them. Both issues are straightforward in the optimization problems reviewed in Sect. 2.2. In a multi-stage recourse problem, for example, we observe the subvector ξ^t of the uncertain parameters ξ at the beginning of time period $t+1$, see Fig. 2.1, right. Likewise, in a stochastic dynamic program, we observe the current state of the MDP at the beginning of each time period.

The situation is different for temporal networks, and it is this difference that complicates the modeling and solution of decision problems in temporal networks. It is customary to assume that the duration and cash flow of a task is observed when the task is completed. However, the completion time of a task depends on the task's start time, which is chosen by the decision maker. Hence, in contrast to the problems studied in Sect. 2.2, the times at which we learn about the random parameters depend on the chosen decision. Recourse problems with decision-dependent uncertainty are studied in [GG06, JWW98], and a robust optimization problem with decision-dependent uncertainty is formulated in [CGS07]. However, the resulting optimization problems are computationally demanding, and they typically have to undergo drastic simplifications before they can be solved.

Apart from the time points at which information becomes available, optimization problems in temporal networks differ from other problems in the type of the revealed information. In many cases, the task durations and cash flows in a temporal network do not correspond to individual parameters, but they are functions of multiple parameters (as is the case in factor models). In such problems, we do not observe the uncertain parameters themselves, but we accumulate knowledge about them with the completion of each task. We can use this information to exclude parameter realizations that are not compatible with the observed durations and cash flows. In contrast, the multi-stage recourse and robust optimization problems reviewed in Sect. 2.2 assume that the decision maker can directly observe the uncertain parameter vector ξ.

Similar to the problems in Sect. 2.2, optimization problems in temporal networks can contain here-and-now as well as wait-and-see decisions. Here-and-now decisions are taken before any of the network tasks are started, whereas a wait-and-see decision associated with task $i \in V$ (e.g., its start time or resource assignment) may depend on all information that is available at the time task i is started. Since the early start schedule optimizes regular objective functions (see Sect. 2.1), it is relatively straightforward to model the task start times as a wait-and-see decision in makespan minimization problems. We will consider problems with a here-and-now resource allocation and wait-and-see task start times in Chaps. 4–6. The situation is fundamentally different in net present value maximization problems where the early start schedule is no longer guaranteed to be optimal. In Chap. 3 we consider a net present value problem in which the resource allocation is fixed, while the task start times can be chosen as a wait-and-see decision.

We close this section with an overview of the literature on temporal networks under uncertainty. Detailed reviews of specific topics will be provided in later chapters.

Although temporal networks have been analyzed for more than 50 years, see for example [Ful61, Kel61, MRCF59], the literature on temporal networks under uncertainty is surprisingly sparse. Until recently, most research on temporal networks under uncertainty assumed a fixed resource allocation and focused on the makespan of the early start schedule. Following the classification in [MÖ1], we can categorize the literature into methods that identify "critical" tasks or task paths [Elm00], simulation techniques to approximate the makespan distribution [AK89], approaches that bound the expected makespan [BM95, BNT02, MN79], and methods that bound the cumulative distribution function of the makespan [LMS01, MÖ1].

Optimization problems that maximize a network's net present value under uncertainty generally model the task start times as a wait-and-see decision, while the resource allocation is assumed to be fixed. The problem has been approximated by a two-stage recourse model in [Ben06], where an optimal delay policy is sought that prescribes how long each task should be delayed beyond its earliest start time. Under the assumption that the task durations are independent and exponentially distributed, the net present value maximization problem is formulated as a continuous-time Markov decision process in [BR97, TSS06]. Finally, approximate solutions for net present value maximization problems have been obtained with a number of heuristics, see [Bus95, OD97, Ö98, TFC98, WWS00]. For an overview of net present value maximization problems in temporal networks, see [HDD97].

Makespan minimization problems under uncertainty typically assume that a resource assignment is selected here-and-now, while the task start times are modeled as a wait-and-see decision. For non-renewable resources, the makespan minimization problem has been formulated as a two-stage recourse model in [Wol85] and as a robust optimization problem in [CGS07, CSS07, JLF07, LJF04]. Except for [CGS07], all of these contributions model the resource assignment as a here-and-now decision. A makespan minimization problem with renewable resources is studied in [MS00].

For reviews of different aspects of optimization problems in temporal networks under uncertainty, see [AK89, BKPH05, Elm05, HL04, HL05, JW00, LI08, MÖ1, Pin08, Sah04].

Springer

http://www.springer.com/978-3-642-23426-2

Optimization of Temporal Networks under Uncertainty
Wiesemann, W.
2012, XII, 160 p., Hardcover
ISBN: 978-3-642-23426-2