Chapter 9
Stochastic Programming Extensions

Abstract This chapter describes PySP, a stochastic programming extension to Pyomo. PySP enables the expression of stochastic programming problems as extensions of deterministic models, which are often formulated first. To formulate a stochastic program in PySP, the user specifies both the deterministic base model and the scenario tree with associated uncertain parameters in Pyomo. Given these two models, PySP provides two paths for solving the corresponding stochastic program. The first alternative involves writing the extensive form and invoking a standard deterministic (mixed-integer) solver. For more complex stochastic programs, PySP includes an implementation of Rockefeller and Wets’ Progressive Hedging algorithm, which provides an effective heuristic for approximating general multi-stage, mixed-integer stochastic programs. By leveraging the combination of a high-level programming language (Python) and the embedding of the base deterministic model in that language (Pyomo), PySP provides completely generic and highly configurable solver implementations.

9.1 Introduction

From the earliest days of using computers to optimization problems, it was recognized that input data is uncertain in most real-world decision problems [20]. In cases where parameter uncertainty is independent of the decisions and information becomes available in a few time stage stages, stochastic programming is an appropriate and widely studied mathematical framework to express and solve uncertain decision problems [11, 58, 39, 54]. Stochastic programming allows the user to explicitly account for the fact that some of the data is uncertain and that the values of parameters may become known over time.

Nevertheless, few solvers devoted to generic stochastic programs include mixed-integer linear or nonlinear problems. With few exceptions, modeling packages that support stochastic programming rely on the translation of problem into the extensive form – a deterministic mathematical programming representation of the stochastic
program in which all scenarios are explicitly represented and solved simultaneously. The extensive form can then be optimized with a standard solver. The number of scenarios, the size of the underlying model, the number of decision stages, and the presence of discrete decision variables are factors that affect the scale of the extensive form. Unfortunately, for real-world problems the direct solution of the extensive form is often (a) too difficult to solve or (b) too large to solve with available system memory.

Iterative decomposition strategies such as the L-shaped method [55] or Progressive Hedging [51] directly address both of these scalability issues, but these methods introduce fundamental parameters and algorithmic challenges. Other approaches include coordinated branch-and-cut procedures [3]. In general, the solution of difficult stochastic programs requires both experimentation with and customization of alternative algorithmic paradigms – thus necessitating for generic and configurable solvers.

PySP is a recently developed stochastic programming extension for Pyomo [59]. To express a stochastic program in PySP, the user specifies both the deterministic base model and the scenario tree with associated uncertain parameters in Pyomo. This separation of deterministic and stochastic problem components is similar to the mechanism proposed in SMPS [12, 25].

Once the deterministic and scenario tree models have been specified, PySP provides two paths for solving the corresponding stochastic program. The first alternative involves writing the extensive form and invoking a deterministic (mixed-integer) linear solver. For more complex stochastic programs, a generic implementation of Rockefeller and Wets’ Progressive Hedging algorithm [51] can be applied. The development of PySP has focused on the use of Progressive Hedging as an effective heuristic for approximating general multi-stage, mixed-integer programs. PySP provides a completely generic and highly configurable solver implementation for Progressive Hedging by leveraging the combination of a high-level programming language (Python) and the embedding of the base deterministic model in that language (Pyomo). Additionally, PySP leverages Pyomo to provide access to the full range of solvers supported within Coopr. Consequently, a broad range of model types can be addressed by PySP.

9.2 Stochastic Programming: Definition and Notations

PySP is designed to express and solve stochastic programming problems, which we now briefly introduce. Readers with no background in stochastic programming will probably need to make use of more comprehensive introductions to both the theoretical foundations and the range of potential applications that can be found in Birge and Louveaux [11], Shapiro et al. [54], and Wallace and Ziemba [58].

We concern ourselves with stochastic optimization problems where uncertain parameters (data) can be represented by a finite set of scenarios $\mathcal{S}$, each of which specifies both (1) a full set of random variable realizations and (2) a corresponding