

Contents

1	Introduction	1
1.1	Mathematical Modeling	1
1.2	Modeling Languages for Optimization	3
1.3	Modeling Graph Coloring	4
1.4	Motivating Pyomo	6
1.4.1	Open Source	7
1.4.2	Customizable Capability	7
1.4.3	Solver Integration	8
1.4.4	Modern Programming Language	8
1.5	Getting Started	9
1.6	Book Summary	10
1.7	Discussion	11
2	Pyomo Modeling Strategies	13
2.1	Modeling Components	13
2.2	Concrete Models: Specifying Components Via Expressions	15
2.3	Concrete Models: Specifying Components Via Rules	17
2.4	Abstract Models	18
2.5	Optimizing Models	19
2.5.1	Optimization with the <code>pyomo</code> Command	19
2.5.2	Optimization Scripts	20
2.6	Discussion	21
2.A	Examples	22
2.A.1	Concrete Pyomo Model with Explicit Variables	22
2.A.2	Concrete Pyomo Model with Indexed Variables	22
2.A.3	Concrete Pyomo Model with External Data	23
2.A.4	Concrete Pyomo Model with Constraint Rules	23
2.A.5	Concrete Pyomo Model with Abstract Component Declarations	24
2.A.6	Using a Function to Construct a Concrete Pyomo Model	25
2.A.7	Abstract Pyomo Model	25
2.A.8	A Python Script that Optimizes a Concrete Pyomo Model	26
2.A.9	A Python Script that Optimizes an Abstract Pyomo Model	27

3	Model Components: Variables, Objectives, and Constraints	29
3.1	Modeling with Components	29
3.2	Variables	30
3.2.1	Var Declarations	31
3.2.2	Variable Initialization	31
3.2.3	Working with Variables	32
3.3	Objectives	33
3.3.1	Simple Declarations	33
3.3.2	Declarations with Rule Functions	34
3.4	Constraints	35
3.4.1	Declarations with Logical Expressions	36
3.4.2	Declarations with Expression Tuples	37
3.5	Constraint Lists	39
3.6	Discussion	40
4	Model Components: Sets and Parameters	43
4.1	Set Data	43
4.1.1	Set Declarations	43
4.1.2	Set Initialization	45
4.1.3	Set Data Validation	47
4.1.4	Set Options	48
4.1.5	RangeSet	49
4.1.6	Discussion of Index Sets	50
4.2	Parameter Data	51
4.2.1	Param Declarations	51
4.2.2	Parameter Initialization	52
4.2.3	Data Validation	54
4.3	Discussion	54
5	Miscellaneous Model Components and Utility Functions	57
5.1	Miscellaneous Components	57
5.2	Advanced Component Indexing	58
5.3	Functions to Support Modeling	59
5.3.1	Generalized Dot Products	60
5.3.2	Generating Sequences	61
5.3.3	Helper Functions	62
5.4	Discussion	63
5.A	Examples	63
5.A.1	Error Checks in a Concrete Model	63
5.A.2	Error Checks in an Abstract Model	64
6	Initializing Abstract Models with Data Command Files	67
6.1	Model Data	67
6.2	The <code>set</code> Command	68
6.2.1	Simple Sets	68

6.2.2	Sets of Tuple Data	70
6.2.3	Set Arrays	71
6.3	The <code>param</code> Command	71
6.3.1	One-dimensional Parameter Data	72
6.3.2	Multi-Dimensional Parameter Data	74
6.4	The <code>import</code> Command	76
6.4.1	Simple Import Examples	77
6.4.2	Import Syntax Options	78
6.4.3	Interpreting Relational Tables	80
6.4.4	Importing from Spreadsheets and Relational Databases	81
6.5	The <code>include</code> Command	84
6.6	Data Namespaces	84
6.7	Discussion	84
6.A	Examples	86
6.A.1	Namespace Data Commands	86
6.A.2	The Diet Problem	87
7	The Pyomo Command-line Interface	91
7.1	Overview	91
7.2	Building a Model Instance	92
7.3	Specifying the Model Object	93
7.4	Selecting Data with Namespaces	93
7.5	Customizing Pyomo's Workflow	94
7.6	Customizing Solver Behavior	98
7.7	Analyze Solver Results	99
7.8	Managing Diagnostic Output	99
7.9	Discussion	101
7.A	Examples	102
7.A.1	Model Object with Non-Default Name	102
7.A.2	Pyomo Data Commands with Multiple Namespaces	103
8	Nonlinear Programming with Pyomo	105
8.1	Introduction	105
8.2	Building Nonlinear Programming Formulations	106
8.2.1	Nonlinear Expressions	106
8.2.2	The Rosenbrock Example	106
8.3	Solving Nonlinear Programming Formulations	109
8.3.1	Nonlinear Solvers	110
8.3.2	Tips for Nonlinear Programming	111
8.4	Nonlinear Programming Examples	112
8.4.1	Variable Initialization in Minimization of Multimodal Function	112
8.4.2	Optimal Quotas for Sustainable Harvesting of Deer	113
8.4.3	Estimation of Parameters in Infectious Disease Models	117
8.4.4	Reactor Design	119

8.A	Examples	123
8.A.1	Rosenbrock	123
8.A.2	Multimodal	124
8.A.3	Deer Harvesting	124
8.A.4	Disease Estimation	126
8.A.5	Reactor Design	128
9	Stochastic Programming Extensions	131
9.1	Introduction	131
9.2	Stochastic Programming: Definition and Notations	132
9.3	Modeling in PySP	134
9.3.1	The Deterministic Reference Model	134
9.3.2	The Scenario Tree	135
9.3.3	Scenario Parameter Specification	136
9.3.4	Compilation of the Scenario Tree Model	137
9.4	Generating and Solving the Extensive Form	138
9.5	Progressive Hedging: A Generic Decomposition Strategy	142
9.5.1	The <code>runph</code> Script	143
9.6	Progressive Hedging Extensions: Advanced Configuration	148
9.6.1	Watson and Woodruff Extensions	148
9.6.2	Solving a Constrained Extensive Form	154
9.6.3	Alternative Convergence Criteria	155
9.6.4	User-Defined Extensions	156
9.7	Solving PH Scenario Sub-Problems in Parallel	157
9.8	Discussion	158
9.A	Examples	159
9.A.1	Farmer Model	159
9.A.2	Farmer Data File	162
9.A.3	Scenario Tree Model	163
9.A.4	Scenario Data Command File	164
10	Scripting and Algorithm Development	165
10.1	Introduction	165
10.2	Scripting Basics	166
10.2.1	A Canonical Optimization Script	166
10.3	Common Scripting Tasks	167
10.3.1	Printing and Comparing Variable Values	167
10.3.2	Looping Over Variables	169
10.3.3	Initializing and Fixing Variables	170
10.3.4	Adding and Dropping Constraints	171
10.3.5	Sharing Solution Results Across Multiple Models	171
10.3.6	Plotting Data with Matplotlib	172
10.3.7	Generating Different Models with a Function	174
10.4	Hybrid Optimization	177
10.5	Benders Decomposition	178

10.6	Discussion	182
10.A	Examples	182
10.A.1	A Simple Optimization Script	182
10.A.2	A Simple Optimization Script with Multiple Data Files . .	183
10.A.3	A More Comprehensive Optimization Script	184
10.A.4	A Comprehensive Optimization Script that Mimics the pyomo Command	185
10.A.5	An Optimization Script with Multiple Models	186
10.A.6	An Optimization Script that Prints the Value of Variables .	187
10.A.7	A Script that Performs Optimization from Two Starting Points	188
10.A.8	A Script that Performs Optimization from a Grid of Starting Points	189
10.A.9	A Script that Reoptimizes a Model after Fixing Variables .	191
10.A.10	An Iterative Optimization Process that Explicitly Activates Select Constraints	192
10.A.11	Sharing Results Between Models	194
10.A.12	Plotting Solver Results with Matplotlib	196
10.A.13	A Sudoku Problem Solved by Iteratively Adding Cuts . .	198
10.A.14	Hybrid Optimization for Parameter Estimation	202
A	Installing Coopr	205
A.1	Installation Overview	205
A.2	Using an Installer	206
A.3	Installing Coopr as a Site Package	207
A.4	Installing a Coopr Release Using <code>coopr_install</code>	207
A.5	Installing a Development Branch Using <code>coopr_install</code>	208
A.6	Discussion	209
B	A Brief Python Tutorial	211
B.1	Overview	211
B.2	Installing and Running Python	212
B.3	Python Line Format	213
B.4	Variables and Data Types	214
B.5	Data Structures	216
B.5.1	Strings	216
B.5.2	Lists	216
B.5.3	Tuples	217
B.5.4	Sets	217
B.5.5	Dictionaries	218
B.6	Conditionals	218
B.7	Iterations and Looping	219
B.8	Generators and List Comprehensions	220
B.9	Functions	221
B.10	Objects and Classes	222

B.11 Modules	222
B.12 Python Resources	223
C Pyomo and Coopr: The Bigger Picture	225
C.1 Coopr Overview	225
C.2 Optimization Solvers	226
References	229
Index	233

Pyomo – Optimization Modeling in Python

Hart, W.E.; Laird, C.; Watson, J.-P.; Woodruff, D.L.

2012, XVIII, 238 p., Hardcover

ISBN: 978-1-4614-3225-8