

Table of Contents

1	Introduction	1
1.1	Lower bounding OPT	2
1.1.1	An approximation algorithm for cardinality vertex cover	3
1.1.2	Can the approximation guarantee be improved?	3
1.2	Well-characterized problems and min-max relations	5
1.3	Exercises	7
1.4	Notes	10

Part I. Combinatorial Algorithms

2	Set Cover	15
2.1	The greedy algorithm	16
2.2	Layering	17
2.3	Application to shortest superstring	19
2.4	Exercises	22
2.5	Notes	26
3	Steiner Tree and TSP	27
3.1	Metric Steiner tree	27
3.1.1	MST-based algorithm	28
3.2	Metric TSP	30
3.2.1	A simple factor 2 algorithm	31
3.2.2	Improving the factor to $3/2$	32
3.3	Exercises	33
3.4	Notes	37
4	Multiway Cut and k-Cut	38
4.1	The multiway cut problem	38
4.2	The minimum k -cut problem	40
4.3	Exercises	44
4.4	Notes	46

5	<i>k</i>-Center	47
5.1	Parametric pruning applied to metric <i>k</i> -center	47
5.2	The weighted version	50
5.3	Exercises	52
5.4	Notes	53
6	Feedback Vertex Set	54
6.1	Cyclomatic weighted graphs	54
6.2	Layering applied to feedback vertex set	57
6.3	Exercises	60
6.4	Notes	60
7	Shortest Superstring	61
7.1	A factor 4 algorithm	61
7.2	Improving to factor 3	64
7.2.1	Achieving half the optimal compression	66
7.3	Exercises	66
7.4	Notes	67
8	Knapsack	68
8.1	A pseudo-polynomial time algorithm for knapsack	69
8.2	An FPTAS for knapsack	69
8.3	Strong NP-hardness and the existence of FPTAS's	71
8.3.1	Is an FPTAS the most desirable approximation algorithm?	72
8.4	Exercises	72
8.5	Notes	73
9	Bin Packing	74
9.1	An asymptotic PTAS	74
9.2	Exercises	77
9.3	Notes	78
10	Minimum Makespan Scheduling	79
10.1	Factor 2 algorithm	79
10.2	A PTAS for minimum makespan	80
10.2.1	Bin packing with fixed number of object sizes	81
10.2.2	Reducing makespan to restricted bin packing	81
10.3	Exercises	83
10.4	Notes	83
11	Euclidean TSP	84
11.1	The algorithm	84
11.2	Proof of correctness	87
11.3	Exercises	89
11.4	Notes	89

Part II. LP-Based Algorithms

12	Introduction to LP-Duality	93
12.1	The LP-duality theorem	93
12.2	Min-max relations and LP-duality	97
12.3	Two fundamental algorithm design techniques	100
12.3.1	A comparison of the techniques and the notion of integrality gap	101
12.4	Exercises	103
12.5	Notes	107
13	Set Cover via Dual Fitting	108
13.1	Dual-fitting-based analysis for the greedy set cover algorithm	108
13.1.1	Can the approximation guarantee be improved?	111
13.2	Generalizations of set cover	112
13.2.1	Dual fitting applied to constrained set multicover	112
13.3	Exercises	116
13.4	Notes	117
14	Rounding Applied to Set Cover	118
14.1	A simple rounding algorithm	118
14.2	Randomized rounding	119
14.3	Half-integrality of vertex cover	121
14.4	Exercises	122
14.5	Notes	123
15	Set Cover via the Primal-Dual Schema	124
15.1	Overview of the schema	124
15.2	Primal-dual schema applied to set cover	126
15.3	Exercises	128
15.4	Notes	129
16	Maximum Satisfiability	130
16.1	Dealing with large clauses	131
16.2	Derandomizing via the method of conditional expectation ...	131
16.3	Dealing with small clauses via LP-rounding	133
16.4	A 3/4 factor algorithm	135
16.5	Exercises	136
16.6	Notes	138
17	Scheduling on Unrelated Parallel Machines	139
17.1	Parametric pruning in an LP setting	139
17.2	Properties of extreme point solutions	140
17.3	The algorithm	141

17.4	Additional properties of extreme point solutions	142
17.5	Exercises	143
17.6	Notes	144
18	Multicut and Integer Multicommodity Flow in Trees	145
18.1	The problems and their LP-relaxations	145
18.2	Primal-dual schema based algorithm	148
18.3	Exercises	151
18.4	Notes	153
19	Multiway Cut	154
19.1	An interesting LP-relaxation	154
19.2	Randomized rounding algorithm	156
19.3	Half-integrality of node multiway cut	159
19.4	Exercises	162
19.5	Notes	166
20	Multicut in General Graphs	167
20.1	Sum multicommodity flow	167
20.2	LP-rounding-based algorithm	169
20.2.1	Growing a region: the continuous process	170
20.2.2	The discrete process	171
20.2.3	Finding successive regions	172
20.3	A tight example	174
20.4	Some applications of multicut	175
20.5	Exercises	176
20.6	Notes	178
21	Sparsest Cut	179
21.1	Demands multicommodity flow	179
21.2	Linear programming formulation	180
21.3	Metrics, cut packings, and ℓ_1 -embeddability	182
21.3.1	Cut packings for metrics	182
21.3.2	ℓ_1 -embeddability of metrics	184
21.4	Low distortion ℓ_1 -embeddings for metrics	185
21.4.1	Ensuring that a single edge is not overshrunk	186
21.4.2	Ensuring that no edge is overshrunk	189
21.5	LP-rounding-based algorithm	190
21.6	Applications	191
21.6.1	Edge expansion	191
21.6.2	Conductance	191
21.6.3	Balanced cut	192
21.6.4	Minimum cut linear arrangement	193
21.7	Exercises	194
21.8	Notes	196

22	Steiner Forest	197
	22.1 LP-relaxation and dual	197
	22.2 Primal–dual schema with synchronization	198
	22.3 Analysis	203
	22.4 Exercises	206
	22.5 Notes	211
23	Steiner Network	212
	23.1 LP-relaxation and half-integrality	212
	23.2 The technique of iterated rounding	216
	23.3 Characterizing extreme point solutions	218
	23.4 A counting argument	220
	23.5 Exercises	223
	23.6 Notes	230
24	Facility Location	231
	24.1 An intuitive understanding of the dual	232
	24.2 Relaxing primal complementary slackness conditions	233
	24.3 Primal–dual schema based algorithm	234
	24.4 Analysis	235
	24.4.1 Running time	237
	24.4.2 Tight example	237
	24.5 Exercises	238
	24.6 Notes	241
25	k-Median	242
	25.1 LP-relaxation and dual	242
	25.2 The high-level idea	243
	25.3 Randomized rounding	246
	25.3.1 Derandomization	247
	25.3.2 Running time	248
	25.3.3 Tight example	248
	25.3.4 Integrality gap	249
	25.4 A Lagrangian relaxation technique for approximation algorithms	249
	25.5 Exercises	250
	25.6 Notes	253
26	Semidefinite Programming	255
	26.1 Strict quadratic programs and vector programs	255
	26.2 Properties of positive semidefinite matrices	257
	26.3 The semidefinite programming problem	258
	26.4 Randomized rounding algorithm	260
	26.5 Improving the guarantee for MAX-2SAT	263
	26.6 Exercises	265
	26.7 Notes	268

Part III. Other Topics

27	Shortest Vector	273
27.1	Bases, determinants, and orthogonality defect	274
27.2	The algorithms of Euclid and Gauss	276
27.3	Lower bounding OPT using Gram–Schmidt orthogonalization	278
27.4	Extension to n dimensions	280
27.5	The dual lattice and its algorithmic use	284
27.6	Exercises	288
27.7	Notes	292
28	Counting Problems	294
28.1	Counting DNF solutions	295
28.2	Network reliability	297
28.2.1	Upperbounding the number of near-minimum cuts	298
28.2.2	Analysis	300
28.3	Exercises	302
28.4	Notes	305
29	Hardness of Approximation	306
29.1	Reductions, gaps, and hardness factors	306
29.2	The PCP theorem	309
29.3	Hardness of MAX-3SAT	311
29.4	Hardness of MAX-3SAT with bounded occurrence of variables	313
29.5	Hardness of vertex cover and Steiner tree	316
29.6	Hardness of clique	318
29.7	Hardness of set cover	322
29.7.1	The two-prover one-round characterization of NP	322
29.7.2	The gadget	324
29.7.3	Reducing error probability by parallel repetition	325
29.7.4	The reduction	326
29.8	Exercises	329
29.9	Notes	332
30	Open Problems	334
30.1	Problems having constant factor algorithms	334
30.2	Other optimization problems	336
30.3	Counting problems	338
30.4	Notes	343

Appendix

A	An Overview of Complexity Theory for the Algorithm Designer	344
A.1	Certificates and the class NP	344
A.2	Reductions and NP-completeness	345
A.3	NP-optimization problems and approximation algorithms ...	346
A.3.1	Approximation factor preserving reductions	348
A.4	Randomized complexity classes	348
A.5	Self-reducibility	349
A.6	Notes	352
B	Basic Facts from Probability Theory	353
B.1	Expectation and moments	353
B.2	Deviations from the mean	354
B.3	Basic distributions	355
B.4	Notes	355
	References	357
	Problem Index	373
	Subject Index	377

ing optimal solutions efficiently, they may still offer footholds for finding near-optimal solutions efficiently. So, at a high level, the process of design of approximation algorithms is not very different from that of design of exact algorithms. It still involves unraveling the relevant structure and finding algorithmic techniques to exploit it. Typically, the structure turns out to be more abstract, and often the algorithmic techniques result from generalizing and extending some of the powerful algorithmic tools developed in the study of exact algorithms.

On the other hand, looking at the process of designing approximation algorithms a little more closely, one can see that it has its own general principles. We illustrate some of these principles in Section 1.1, using the following simple setting.

Problem 1.1 (Vertex cover) Given an undirected graph $G = (V, E)$, and a cost function on vertices $c: V \rightarrow \mathbb{Q}^+$, find a minimum cost vertex cover, i.e., a set $V' \subseteq V$ such that every edge has at least one endpoint incident at V' . The special case, in which all vertices are of unit cost, will be called the *cardinality vertex cover problem*.

Since the design of an approximation algorithm involves delicately attacking NP-hardness and salvaging from it an efficient approximate solution, it will be useful for the reader to review some key concepts from complexity theory. Appendix A and some extracts in Section 1.3 have been provided for this purpose.