

Optimization Algorithms for Planar Graphs

Philip N. Klein and Shay Mozes
copyright

February 3, 2024

Contents

1 Rooted forests and trees	11
1.1 Rootward computations	12
1.2 Separators for rooted trees	13
1.2.1 Node separator	13
1.3 Edge separators	14
1.4 Computation time for finding separators	16
1.4.1 Recursive tree decomposition	16
2 Basic graph definitions	17
2.1 Edge-centric definition of graphs	18
2.2 Walks, paths, and cycles	19
2.2.1 Connectedness	20
2.2.2 Two-edge-connectivity and cut-edges	21
2.2.3 Subgraphs and edge subgraphs	21
2.2.4 Deletion of edges and vertices	22
2.2.5 Contraction of edges	22
2.2.6 Minors	23
3 Elementary graph theory	25
3.1 Spanning forests and trees	25
3.1.1 Nontree edges and fundamental cycles	26
3.2 Cuts	26
3.2.1 (Undirected) cuts	26
3.2.2 (Directed) dicuts	26
3.2.3 Dart cuts	26
3.2.4 Simple cuts	26
3.2.5 Tree edges and fundamental cuts	27
3.2.6 Paths and Cuts	27
3.3 Vector Spaces	28
3.3.1 The cut space	29
3.3.2 The cycle space	29
3.3.3 Bases for the cut space and the cycle space	30
3.3.4 Another basis for the cut space	31
3.3.5 Conservation and circulations	31

3.4	Embedded graphs	32
3.4.1	Embeddings	33
3.4.2	Euler characteristic and genus	35
3.4.3	Remark on the connection to geometric embeddings	35
3.4.4	The dual graph	35
3.4.5	Connectedness properties of embedded graphs	36
3.4.6	Cut-edges and self-loops	37
3.4.7	Deletion	37
3.4.8	Compression (deletion in the dual) and contraction	37
3.5	Chapter Notes	39
4	Planar embedded graphs	41
4.1	Planar embeddings	41
4.2	Contraction preserves planarity	41
4.3	Sparsity of planar embedded graphs	42
4.3.1	Strict graphs and strict problems	42
4.3.2	Semi-strictness	43
4.3.3	Orientations with bounded outdegree	43
4.3.4	Maintaining a bounded-outdegree orientation for a dynamically changing graph	44
4.3.5	Analysis of the algorithm for maintaining a bounded-outdegree orientation	44
4.4	Cycle-space basis for planar graphs	47
4.4.1	Representing a circulation in terms of face potentials	48
4.5	Interdigitating trees	49
4.6	Simple-cut/simple-cycle duality	50
4.6.1	Compressing self-loops	51
4.6.2	Compression and deletion preserve planarity	52
4.7	Left, right, and Crossings	53
4.7.1	Emanating and entering from left and right	53
4.7.2	Crossing walks	53
4.8	Faces, edges, and vertices enclosed by a non-self-crossing cycle	53
4.9	Trimming	54
4.10	Biconnectivity	55
4.11	Representing embedded graphs in implementations	56
5	Separators in planar graphs	57
5.1	Triangulation	57
5.2	Weights and balance	57
5.3	Fundamental-cycle separators	57
5.4	Breadth-first search	59
5.5	$O(\sqrt{n})$ -vertex separator	59
5.5.1	Size of the separator	62
5.6	Noncrossing families of subsets	63
5.7	The connected-components tree	63
5.7.1	Vertex and face labels	63

5.7.2 The connected-components tree	64
5.8 Cycle separators	66
5.8.1 Shortcutting a fundamental-cycle separator	66
5.8.2 Balanced, short, and simple cycle	67
5.9 Division into regions	68
5.10 Computing a Decomposition Tree	69
5.10.1 Number of Holes	72
5.10.2 Number of Vertices and Boundary Vertices	73
5.10.3 Admitting an r -division	73
5.10.4 Running time	76
5.11 Recursive divisions	77
5.12 Chapter Notes	77
6 Shortest paths with nonnegative lengths	79
6.1 Shortest-path basics: path-length property and relaxed and tense darts	79
6.2 Using a division in computing shortest-path distances	80
6.2.1 The algorithm	81
6.3 Correctness	83
6.4 The Dijkstra-like property of the algorithm	84
6.5 Accounting for costs	85
6.6 The Payoff Theorem	89
6.7 Analysis	90
6.8 Parameters	91
6.9 History	93
7 Multiple-source shortest paths	95
7.1 Slack costs, relaxed and tense darts, and consistent price vectors	95
7.1.1 Slack costs	95
7.1.2 Relaxed and tense darts	96
7.1.3 Consistent price vectors	96
7.2 Specification of multiple-source shortest paths	97
7.2.1 Pivots	98
7.3 Contiguity property of shortest-path trees in planar graphs	99
7.4 The abstract MSSP algorithm	99
7.4.1 Analysis of the abstract algorithm	100
7.5 CHANGERoot: the inner loop of the MSSP algorithm	100
7.6 Which darts are candidates for pivoting in?	102
7.7 Efficient implementation	102
7.7.1 CHANGERoot	104
7.7.2 Data structure	105
7.8 Number of pivots - the degenerate case	107
7.9 Using the output of the MSSP algorithm	109
7.9.1 Paths	109
7.9.2 Distances	109
7.9.3 Distance data structure	111

7.10 Chapter Notes	112
8 Shortest paths with negative lengths	113
8.1 Total Monotonicity and the Monge Property	114
8.1.1 Boundary distances and the Monge Property	114
8.1.2 Finding all column minima of a Monge matrix	115
8.1.3 Finding all the column minima of a triangular Monge matrix	117
8.2 The Algorithm	117
8.3 Computing Single-Source Inter-Part Boundary Distances	119
8.4 Computing Single-Source Inter-Part Distances	121
8.5 Correctness and Analysis	121
8.6 Chapter Notes	122
9 Shortest paths in dense distance graphs	123
9.1 Decomposing a DDG into bipartite graphs	124
9.2 The Monge heap	126
9.3 Implementing Dijkstra's algorithm using Monge heaps	126
9.3.1 Analysis	128
9.4 Implementing Monge heaps	128
9.4.1 Analysis	132
9.5 Chapter Notes	132
10 Single-source, single-sink max flow	133
10.1 Flow assignments, capacity assignments, and feasibility	133
10.1.1 Negative capacities	133
10.2 Circulations	133
10.2.1 Capacity-respecting circulations in planar graphs	134
10.3 st -flows	134
10.4 Max limited flow in st -planar graphs	135
10.4.1 st -planar embedded graphs and augmented st -planar embedded graphs	135
10.4.2 The set-up	136
10.4.3 The algorithm	137
10.5 Max flow in general planar graphs	138
10.6 The algorithm	138
10.7 Erickson's analysis	140
10.7.1 Dual tree is shortest-path tree	140
10.7.2 Crossing numbers	140
10.8 Covering space	141
10.8.1 The Universal Cover	141
10.9 Finishing the proof	142
10.10 Efficient Implementation	143
10.11 Chapter Notes	143
11 Multiple-source, multiple-sink max flow	145

12 Distance Oracles	147
12.1 An approximate distance oracle for undirected planar graphs . . .	148
12.1.1 Overall strategy	148
12.1.2 Connections to a shortest path	149
12.1.3 The oracle	151
12.1.4 Efficient construction	152
12.2 An Exact distance oracles with $\tilde{O}(n)$ space and $\tilde{O}(\sqrt{n})$ query time	154
12.3 An exact oracle with $\tilde{O}(n^{4/3})$ space and $O(\log^2 n)$ query time . .	155
12.3.1 Additively weighted Voronoi diagrams	155
12.3.2 Point location in Voronoi diagrams	157
12.3.3 The oracle	161
13 Primal-dual method for approximation algorithm	165
13.1 Goemans and Williamson's analysis of the primal-dual approxi- mation algorithm	165
13.2 Proving the bound for vertex-weighted Steiner tree	166
13.3 Covering all directed cycles	168
14 Branchwidth and local approximation schemes	173
14.1 Dynamic programming on a rooted tree	173
14.2 Carvings	174
14.3 Carving-decomposition: carving of a vertex set	175
14.3.1 Solving edge dominating set on a graph with a carving- decomposition of small width	175
14.4 Carving-decomposition of a planar graph	176
14.5 Branch decomposition: Carving of an edge-set	177
14.5.1 Solving vertex cover on a graph with a branch decompo- sition of small width	178
14.5.2 Biconnectivity and the block-cut tree	178
14.5.3 Biconnected components and branchwidth	179
14.6 A branchwidth bound for planar graphs	179
14.6.1 The face-vertex incidence graph	180
14.6.2 The embedded face-vertex incidence graph	181
14.6.3 The dual of the face-incidence graph	183
14.6.4 From a carving-decomposition of $M(G)$ to a branch-decomposition of G	184
14.6.5 Proof of the Radius-Branchwidth Theorem	186
14.7 Approximation schemes	186
14.7.1 The subgraph induced by k BFS levels has branchwidth at most $2k$	186
14.7.2 An approximation scheme for VERTEX COVER	187
14.7.3 An approximation scheme for maximum-weight indepen- dent set	189
14.7.4 Maximum-weight set of edge-disjoint triangles	191
14.7.5 Summary of approximation-scheme methodology	192

15 Approximation scheme for the traveling salesman problem	193
15.1 Cutting into small pieces	193
15.1.1 First attempt	194
15.1.2 Approximation through deletion decomposition	195
15.2 The traveling salesman problem	197
15.3 Approximating unit-weight EULERIAN BISUBGRAPH	199
15.4 Beyond unit-weight graphs: a sparsifier	201
15.5 Contraction decomposition	203
15.6 The framework	203
15.6.1 Dual framework	205
15.7 Properties of tours	205
15.8 Lifting for TSP	206
15.9 Spanner	207
15.10 TSP on bounded-branchwidth planar graphs	213
16 The brick decomposition and approximation schemes for Steiner problems	215
16.1 Introducing the brick decomposition	216
16.2 Portals	217
16.3 Portalization	218
16.4 Spanners for Steiner TSP and Steiner tree	220
16.5 Beyond spanners: A more efficient PTAS for Steiner tree	221
16.6 Brick decomposition: the construction	221
16.6.1 Strip decomposition	222
16.6.2 Columns	225
16.7 Statement of subroutine lemmas for Steiner tree structure theorem	226
16.8 Structure of Steiner tree within bricks	227
16.8.1 Paths $\bar{P}_0, \dots, \bar{P}_\kappa$	228
16.8.2 The forest F' and paths $Q_0, \dots, Q_\kappa$	228
16.8.3 The forest $\tilde{F}$	231
16.8.4 Type-1 and type-2 components	231
16.8.5 Construction of $\tilde{K}$	232
16.8.6 Decomposition of K_i into K_i^N and K_i^S	232
16.8.7 SPAN1	234
16.8.8 SPAN2	240
17 Approximation Schemes for some problems with optional connectivity	243
17.1 Introduction	243
17.2 PC Clustering	245
17.2.1 The <i>PC Clustering</i> algorithm	246
17.2.2 The weight of the output from the <i>PC Clustering</i> algorithm	247

18 Appendix: Splay trees and link-cut trees	249
18.1 Binary Search Trees	251
18.1.1 Delta Representation of weights	251
18.1.2 Supporting searches for small weight	252
18.1.3 Delta Representation of min-weights	253
18.1.4 Delta representation of left-right	253
18.1.5 Rotation: An order-preserving structural change to a binary search tree	254
18.1.6 Updating Delta representations in a rotation	254
18.1.7 Updating Δ_{minw} representations in a rotation	256
18.2 Splay trees	256
18.2.1 Potential Functions for Amortized Analysis	257
18.2.2 Analysis of splay trees	258
18.3 Representation of link-cut trees	261
18.3.1 High-level analysis of the <i>expose</i> operation	263
18.3.2 Representation of trees	264
18.3.3 Link-cut trees that do not support descendant search	264
18.3.4 Implementing the <i>expose</i> operation for trees not supporting descendant search	266
18.3.5 Analysis of EXPOSE(<i>u</i>) for trees not supporting descendant search	267
18.4 Link-cut trees that support descendant search	268
18.5 Topological updates in link-cut trees	271
18.5.1 Analysis of link and cut operations	271
18.5.2 Evert	272
18.6 Weight updates for link-cut trees	272
18.6.1 Supporting ADDTODESCENDANTS	272
18.6.2 Supporting ADDTOANCESTORS	273
18.6.3 Getting the weight of a node	274
18.7 Weight searches in link-cut trees	274
18.7.1 Supporting ancestor searches	276
18.7.2 Supporting descendant searches	278
18.7.3 Representing trees with dart weights	280
18.8 Chapter Notes	280