

Contents

List of Figures	xiii
List of Tables	xv
List of Algorithms	xvii
Preface	xix
1 Introduction	1
1.1 Assignments	1
1.2 Linear assignment problems	4
1.3 Quadratic assignment problems	7
1.4 Multi-index assignment problems	8
1.5 Research lines for assignment problems	11
2 Theoretical foundations	13
2.1 The marriage theorem and the existence of perfect matchings	13
2.2 The assignment polytope	24
3 Bipartite matching algorithms	35
3.1 Bipartite matching	35
3.2 A labeling method for finding a maximum cardinality matching	36
3.3 The Hopcroft–Karp algorithm	42
3.4 Improvements by Alt, Blum, Mehlhorn, and Paul	47
3.5 Matchings in convex bipartite graphs	52
3.5.1 Algorithms	53
3.5.2 Applications	55
3.6 Maximum matchings and matrix algorithms	57
3.7 Perfect matchings in bipartite random graphs	59
3.8 Applications of maximum matching problems	64
3.8.1 Vehicle scheduling problems	65
3.8.2 Time slot assignment problem	66
4 Linear sum assignment problem	73
4.1 Introduction	73
4.1.1 Mathematical model	74

4.1.2	Complementary slackness	75
4.1.3	Historical notes, books, and surveys	77
4.2	Primal-dual algorithms	79
4.2.1	The Hungarian algorithm	79
4.2.2	An $O(n^3)$ implementation of the Hungarian algorithm	85
4.2.3	Cost-scaling algorithms	87
4.3	The Dinic–Kronrod algorithm	89
4.4	Shortest path implementation of the Hungarian algorithm	93
4.4.1	Transformation to a minimum cost flow problem	93
4.4.2	Basic implementation	94
4.4.3	Efficient implementations	98
4.4.4	Preprocessing	99
4.5	Primal algorithms	104
4.5.1	Non-simplex algorithms	104
4.5.2	Simplex-based algorithms	106
4.6	Dual algorithms	112
4.6.1	Non-simplex algorithms	112
4.6.2	Simplex-based algorithms	114
4.6.3	Auction algorithms	119
4.6.4	Pseudoflow algorithms	123
4.7	Other algorithms	126
4.8	Summary of sequential algorithms	127
4.9	Software	127
4.9.1	Computer codes	129
4.10	Experimental analysis	131
4.10.1	Classes of instances	132
4.10.2	Experiments	133
4.11	Parallel algorithms	138
4.11.1	Theoretical results	138
4.11.2	Auction algorithms	139
4.11.3	Shortest path algorithms	141
4.11.4	Primal simplex algorithms	142
5	Further results on the linear sum assignment problem	145
5.1	Asymptotic analysis	145
5.1.1	Expected optimum value	145
5.1.2	Asymptotic analysis of algorithms	149
5.2	Monge matrices and the linear sum assignment problem	150
5.3	Max-algebra and the linear sum assignment problem	153
5.4	Variants	158
5.4.1	Ranking solutions	158
5.4.2	k -cardinality assignment problem	163
5.4.3	Semi-assignment problem	164
5.4.4	Rectangular cost matrix	165
5.5	Applications	165
5.5.1	Mean flow time minimization on parallel machines	166

5.5.2	Categorized assignment scheduling	167
5.5.3	Optimal depletion of inventory	167
5.5.4	Personnel assignment with seniority and job priority	168
5.5.5	Navy personnel planning	168
6	Other types of linear assignment problems	171
6.1	Introduction	171
6.2	Bottleneck assignment problem	172
6.2.1	Background	172
6.2.2	Threshold algorithms	174
6.2.3	A dual method	177
6.2.4	Augmenting path methods	178
6.2.5	Sparse subgraph techniques	185
6.2.6	Special cases	186
6.2.7	Asymptotic results	187
6.2.8	Variants and applications	188
6.2.9	Software	191
6.3	Algebraic assignment problem	191
6.4	Sum- k assignment problem	195
6.5	Balanced assignment problem	195
6.6	Lexicographic bottleneck assignment problem	198
7	Quadratic assignment problems: Formulations and bounds	203
7.1	Introduction	203
7.1.1	Models and applications	203
7.1.2	Traveling salesman problem	206
7.1.3	Minimum weight feedback arc set problem	206
7.1.4	Graph partitioning and maximum clique problem	207
7.1.5	Graph isomorphism and graph packing problems	209
7.1.6	Quadratic bottleneck assignment problem	210
7.1.7	Complexity issues	210
7.2	Formulations	211
7.2.1	Trace formulation	212
7.2.2	Kronecker product formulation	213
7.2.3	Convex and concave integer models	214
7.2.4	Mean objective value of feasible solutions	216
7.3	Linearizations	217
7.3.1	Kaufman–Broeckx	218
7.3.2	Balas–Mazzola	219
7.3.3	Frieze–Yadegar	221
7.3.4	Adams–Johnson	221
7.3.5	Higher level linearizations	222
7.4	Quadratic assignment polytopes	223
7.5	Gilmore–Lawler bound and reduction methods	225
7.5.1	Gilmore–Lawler bound	225
7.5.2	Reduction methods	229

7.6	Admissible transformations and other bounds	233
7.6.1	Admissible transformations	233
7.6.2	General Gilmore–Lawler bound	235
7.7	Eigenvalue bounds	238
7.7.1	Symmetric Koopmans–Beckmann problems	238
7.7.2	Nonsymmetric Koopmans–Beckmann problems	244
7.8	Bounds by semidefinite programming	245
7.9	Bounds by convex quadratic programming	247
8	Quadratic assignment problems: Algorithms	249
8.1	Exact algorithms	249
8.1.1	Benders’ decomposition	249
8.1.2	Branch-and-bound	250
8.1.3	Branch-and-cut	253
8.1.4	Parallel and massively parallel algorithms	253
8.2	Heuristics	255
8.2.1	Construction algorithms	255
8.2.2	Limited exact algorithms	255
8.2.3	Basic elements of metaheuristic methods	257
8.2.4	Simulated annealing	259
8.2.5	Tabu search	260
8.2.6	Genetic algorithms	261
8.2.7	Greedy randomized adaptive search procedure	262
8.2.8	Ant colony optimization	263
8.2.9	Scatter search and path relinking	264
8.2.10	Large scale and variable neighborhood search	265
8.3	Computer codes	266
8.4	Easy and hard special cases	267
8.4.1	Sum and product matrices	267
8.4.2	Diagonally structured matrices	275
8.4.3	Ordered matrices	279
8.4.4	Problems generated by a special underlying graph	279
9	Other types of quadratic assignment problems	281
9.1	Quadratic bottleneck assignment problem	281
9.1.1	Gilmore–Lawler bound for the Koopmans–Beckmann form	282
9.1.2	Gilmore–Lawler bound for the general form	284
9.2	Asymptotic results	285
9.2.1	Generic combinatorial optimization problem	285
9.2.2	Quadratic assignment problem	290
9.2.3	Quadratic bottleneck assignment problem	291
9.3	Cubic and quartic assignment problem	291
9.4	Quadratic semi-assignment problem	293
9.4.1	Applications	295
9.4.2	Solution methods	300

10	Multi-index assignment problems	305
10.1	Introduction	305
10.2	Axial 3-index assignment problem	305
10.2.1	Applications	307
10.2.2	Polyhedral aspects	307
10.2.3	Complexity and approximation	308
10.2.4	Lower bounds and exact solution methods	308
10.2.5	Heuristic algorithms	310
10.2.6	Special cases	310
10.2.7	Asymptotic behavior	311
10.3	Planar 3-index assignment problem	312
10.3.1	Applications and special cases	313
10.3.2	Polyhedral aspects, lower bounds, and algorithms	314
10.4	General multi-index assignment problems	315
10.4.1	Applications	317
10.4.2	Algorithms and asymptotic behavior	317
	Bibliography	319
	Author Index	367
	Index	377

List of Figures

1.1	Different representations of assignments	2
1.2	Perfect matching and corresponding network flow model	4
1.3	Axial (a) and planar (b) 3-index assignment problems	9
2.1	The left figure shows a maximal matching which cannot be extended by any edge. The right figure shows a maximum matching of larger size . .	14
2.2	Maximum matching (bold edges) and minimum vertex cover (square vertices)	17
2.3	Vertex cover in Example 2.13	21
2.4	A maximum flow and a minimum cut in the network of Example 2.13. The forward arcs crossing the cut are dotted	21
2.5	Example for the Mendelsohn-Dulmage theorem: (a) original matchings M_1 (solid lines) and M_2 (dashed lines); (b) final matching	23
2.6	Construction of a maximum matching which contains all matched vertices of an arbitrary given matching	24
2.7	Construction of a labeled rooted tree after the insertion of 5 arcs. As $n = 10$, we have $10 - 5 = 5$ connected components	31
3.1	(a) Matching M ; (b) augmented matching $M \oplus P$, for path $P = (b, b', c, c', d, d')$	37
3.2	Bipartite graph for Examples 3.5, 3.13, and 3.14	39
3.3	The upper figure (a) shows a flow in a network stemming from a maximum matching problem. Figure (b) shows the complete incremental network including dashed arcs that never occur in an augmenting path. Figure (c) shows only those arcs that are relevant for finding a maximum flow . . .	41
3.4	Layered graph	45
3.5	Incremental graph in the second iteration of Example 3.13	47
3.6	Convex bipartite graph at left and doubly convex bipartite graph at right .	53
3.7	Convex bipartite graph used in Example 3.18	54
3.8	The local density at location x is 3; the local density at location y is 4. The density d of this assignment is $d = 4$	56
3.9	This graph shows entry vertices on the upper line and exit vertices on the lower line. The edges are chosen according to Lemma 3.21 for testing whether a terminal assignment with density $d = 2$ does exist	57

3.10	Figure (a) shows a network with (a minimum number) of node disjoint paths (bold) which cover all nodes; Figure (b) shows the corresponding maximum matching	65
4.1	Node-edge incidence matrix of G	74
4.2	(a) Graph G^0 ; (b) alternating tree; (c) new graph G^0	81
4.3	(a) Alternating tree; (b) new graph G^0 ; (c) augmenting tree	84
4.4	Values $c_{ij} - \Delta_j$ during execution of the Dinic–Kronrod algorithm	92
4.5	Values $c_{ij} - v_j$ in Augmenting_row_reduction with $\hat{t} = 1$	102
4.6	Values $c_{ij} - v_j$ in Augmenting_row_reduction with $\hat{t} = 2$	103
4.7	(a) Strongly feasible tree on G ; (b) an easier view	107
4.8	Degenerate pivoting on a backward edge	108
4.9	Non-degenerate pivoting on a forward edge	108
4.10	Intermediate solution of Example 4.19	110
4.11	Final solution of Example 4.19	110
4.12	First cycle of Example 4.20	113
4.13	Second cycle of Example 4.20	114
4.14	First cycle of Example 4.23	117
4.15	Second cycle of Example 4.23	117
4.16	Input matrix and scaled matrix for Example 4.26	122
5.1	Four elements involved in the Monge property	151
5.2	Branching rule of the Murty’s algorithm	159
5.3	(a) instance for Example 5.14; (b) matchings	161
5.4	Branching rule of the Chegireddy–Hamacher algorithm	161
6.1	Bipartite graph $G[4]$	176
6.2	Threshold matrix and bipartite graph $G[7]$	176
6.3	Threshold matrix and bipartite graph $G[5]$	176
6.4	Four elements involved in the Monge property	186
7.1	Graph of Example 7.2 and its weighted adjacency matrix	208
7.2	(a) A star graph; (b) a double star graph	228
7.3	A shortest path triple (j, l, r) in a rectangular grid	228
9.1	Form of the permuted cost matrix for $R \sum w_i C_i$	299
10.1	Pictorial representation of the constraints of an axial 3AP	306
10.2	Pictorial representation of the constraints of a planar 3AP	313

List of Tables

4.1	Cost evolution in a scaling algorithm	88
4.2	Evolution of algorithms for LSAP	128
4.3	The tested codes	129
4.4	Dense uniformly random instances	133
4.5	Dense geometric instances	134
4.6	Dense no-wait flow-shop instances with 10 machines and n jobs	135
4.7	Dense two-cost instances	135
4.8	Dense Machol and Wien instances. Time limit = 3000 seconds	135
4.9	Sparse uniformly random instances	136
4.10	Sparse geometric instances	137
4.11	Sparse no-wait flow-shop instances with 10 machines and n jobs	137
4.12	Sparse two-cost instances	137
6.1	Complexity of threshold-based bottleneck assignment algorithms	185
8.1	Complexity status of Koopmans–Beckmann problems involving Monge and inverse Monge matrices	275

List of Algorithms

Algorithm 3.1	Cardinality_matching	37
Algorithm 3.2	Hopcroft_Karp	46
Algorithm 3.3	Revised_cardinality_matching	48
Algorithm 3.4	Convex	53
Algorithm 3.5	Test_perfect_matching	58
Algorithm 3.6	Equalize_sums	67
Algorithm 3.7	Decompose	67
Algorithm 4.1	Procedure Basic_preprocessing	76
Algorithm 4.2	Procedure Alternate(k)	80
Algorithm 4.3	Hungarian	82
Algorithm 4.4	Procedure Augment(k)	85
Algorithm 4.5	Hungarian_3	86
Algorithm 4.6	Procedure DK_basic(s)	90
Algorithm 4.7	Procedure DK_reduce(s)	90
Algorithm 4.8	Dinic_Kronrod	91
Algorithm 4.9	Procedure Shortest_path(k)	95
Algorithm 4.10	Hungarian_SP	97
Algorithm 4.11	Procedure Three_edge_preprocessing	99
Algorithm 4.12	Procedure Reduction_transfer	101
Algorithm 4.13	Procedure Augmenting_row_reduction	102
Algorithm 4.14	Primal_simplex	109
Algorithm 4.15	Hung_Rom	112
Algorithm 4.16	Signature	116
Algorithm 4.17	Pseudoflow	124
Algorithm 4.18	Procedure Double_push(k)	125
Algorithm 4.19	Procedure Multi-path Updating	142
Algorithm 5.1	Murty	159
Algorithm 5.2	Revised_ranking	162
Algorithm 6.1	Threshold	174
Algorithm 6.2	Dual_LBAP	177
Algorithm 6.3	Augmenting_LBAP	181
Algorithm 6.4	Procedure Dijkstra(i)	182
Algorithm 6.5	Algebraic_assignment	194

Algorithm 6.6	Balanced_assignment	196
Algorithm 6.7	Lex_BAP	200
Algorithm 8.1	Procedure BB_QAP	256