

Algorithmes de graphes

avec
programmes
en Pascal

Christian
PRINS

disquette
incluse

 EYROLLES

2-005-18-1

2-005-18-1

ALGORITHMES de GRAPHES

Avec programmes en Pascal

Christian PRINS



DEUXIÈME TIRAGE
1997


EYROLLES

Table des matières

Préambule	XV
Chapitre 1 - Introduction aux graphes	1
1.1 Intérêt des graphes et applications	1
1.2 Graphes orientés	2
1.2.1 Les p -graphes	2
1.2.2 Les 1-graphes	3
1.3 Graphes non orientés	4
1.4 Parties de graphes	6
1.5 Parcours et connexité	7
1.5.1 Définitions sur les parcours	7
1.5.2 Connexité et forte connexité	8
1.5.3 Parcours eulériens et hamiltoniens	9
1.6 Quelques graphes particuliers	10
1.7 Références	13
Chapitre 2 - Complexité des algorithmes et problèmes difficiles	15
2.1 Introduction	15
2.2 Notions sur la complexité des algorithmes	15
2.2.1 Comment mesurer l'efficacité ?	15
2.2.2 Taille d'une donnée	16
2.2.3 Ordre d'une fonction	17
2.2.4 Bons et mauvais algorithmes	19
2.3 Problèmes d'optimisation combinatoire	21
2.3.1 Définitions	21
2.3.2 Problèmes combinatoires faciles et difficiles	23
2.3.3 Exemples de problèmes faciles et difficiles	23
2.3.3.1 Problèmes de graphes	23
2.3.3.2 Exemples de problèmes hors des graphes	28
2.4. Notions de théorie de la complexité	31
2.4.1 La théorie de la complexité	31
2.4.2 Les classes P et NP	32
2.4.3 Problèmes NP -complets	33
2.4.3.1 Définition	33
2.4.3.2 Comment montrer qu'un problème est NP -complet ?	34

2.4.3.3 La conjecture $P \neq NP$	35
2.4.4 Impact sur l'approche pratique d'un problème réel	36
2.5 Références	36
 Chapitre 3 - Résolution des problèmes difficiles	 39
3.1 Introduction	39
3.2 Heuristiques	39
3.2.1 Présentation générale	39
3.2.2 Evaluation des heuristiques	40
3.2.2.1 Définitions	40
3.2.2.2 Evaluation <i>a priori</i>	40
3.2.2.3 Evaluation <i>a posteriori</i>	42
3.2.3 Heuristiques construisant une seule solution	46
3.2.3.1 Sac à dos	46
3.2.3.2 Bin packing	47
3.2.3.3 Méthode à pénalités pour le PVC	47
3.2.4 Recherches locales	48
3.2.5 Méthodes de recherche globale	50
3.2.5.1 Recuit simulé	50
3.2.5.2 Méthodes taboues	52
3.2.5.3 Algorithmes génétiques	54
3.3 Méthodes arborescentes	55
3.3.1 Principes généraux	55
3.3.2 Séparation	55
3.3.3 Evaluation	56
3.3.4 Stratégie d'exploration	57
3.3.4.1 Choix du sommet à séparer	57
3.3.4.2 Choix de la décision de séparation en un sommet	59
3.3.5 Arrêt et performances	59
3.3.6 Implémentation des méthodes arborescentes	60
3.3.6.1 Squelette de méthode PSEP	60
3.3.6.2 Squelette de méthode PSES	61
3.3.7 Un exemple: le sac à dos en 0-1	62
3.3.7.1 Rappels et exemple numérique	62
3.3.7.2 Définition des nœuds et règle de séparation	63
3.3.7.3 Evaluation en un nœud S	63
3.3.7.4 Choix du sommet à séparer et de la décision de séparation	63
3.4 Références	64
 Chapitre 4 - Vers une boîte à outils de graphes	 65
4.1 Introduction	65
4.2 Remarques sur le pseudo-langage utilisé	65

4.3 Codages de la structure de données graphe	67
4.3.1 Matrice d'adjacence	67
4.3.2 Listes d'adjacence	68
4.4 Autres structures de données utiles	70
4.4.1 Piles et files	70
4.4.1.1 Principes	70
4.4.1.2 Une implémentation mixte pile-file	71
4.4.2 Structure de tas	74
4.4.2.1 Principes	74
4.4.2.2 Primitives de tas	75
4.5 Principes et limites d'une boîte à outils	77
4.6 Rappels sur les pointeurs en Pascal	79
4.7 Constantes et types de l'unité de base GPCore	82
4.7.1 Constantes	82
4.7.2 Types de base	82
4.7.3 Type graphe matriciel : Matrix	84
4.7.4 Type graphe-liste : Graph	87
4.7.5 Types pour piles, files et tas	89
4.8 Procédures et fonctions de GPCore	89
4.8.1 Vue générale	89
4.8.2 Lecture de graphes	91
4.8.2.1 Principes généraux	91
4.8.2.2 Lecture de graphes matriciels	91
4.8.2.3 Lecture de graphes-listes	93
4.8.3 Ecriture de graphes	95
4.8.4 Génération aléatoire	96
4.8.4.1 Généralités	96
4.8.4.2 Graphes matriciels aléatoires	97
4.8.4.3 Graphes-listes aléatoires	98
4.8.5 Conversions de graphes	98
4.8.6 Procédures diverses	99
4.9 Listing de la partie Interface de GPCore	100
4.10. Un exemple d'utilisation	103
4.11 Références	104

Chapitre 5 - Manipulations de base sur les graphes 105

5.1 Introduction	105
5.2 Construction des listes de prédécesseurs	105
5.2.1 Définition du problème et intérêt	105
5.2.2 Algorithme	106
5.2.3 Procédure Pascal BuildPreds	107
5.2.4 Exemple d'utilisation	108
5.3 Décomposition d'un graphe en niveaux	109
5.3.1 Définition du problème et applications	109

5.3.2 Algorithme	110
5.3.3 Procédure Pascal GetLayers	111
5.3.4 Exemple : calculs sur un ordinateur parallèle	112
5.4 Exploration de graphes	114
5.4.1 Forme générale d'un algorithme d'exploration	114
5.4.2 Exploration en largeur	116
5.4.3 Exploration en profondeur	116
5.4.4 Applications simples de l'exploration	117
5.4.4.1 Arborescence de visite	117
5.4.4.2 Numéros de visite	118
5.4.4.3 Plus courts chemins en nombre d'arcs	118
5.4.4.4 Détection de circuits	118
5.4.5 Procédures Pascal BFS, DFS, GetCircuit et ExtractPath	119
5.4.6 Exemple d'utilisation	123
5.5 Composantes connexes	124
5.5.1 Le problème et ses applications	124
5.5.2 Algorithme et procédure Pascal AllCC	124
5.5.3 Exemple d'utilisation	126
5.6 Composantes fortement connexes	127
5.6.1 Le problème et ses applications	127
5.6.2 Un algorithme simple	128
5.6.3 Algorithme de Tarjan	129
5.6.4 Procédure Pascal AllSCC	131
5.6.5 Etude de cas : évaluation des deux algorithmes	133
5.7 Test de bipartisme	135
5.7.1 Le problème et ses applications	135
5.7.2 Algorithme	137
5.7.3 Procédure Pascal Bipartite	137
5.7.4 Exemple d'utilisation	138
5.8 Listing de l'interface de l'unité GPMANIP	139
5.9 Références	140

Chapitre 6 - Problèmes de chemins optimaux 141

6.1 Introduction	141
6.2 Les problèmes de chemins optimaux	141
6.2.1 Les grands types de problèmes	141
6.2.2 Les deux familles d'algorithmes	143
6.2.3 Liste des principaux algorithmes	144
6.2.3.1 Problème A	144
6.2.3.2 Problème B	144
6.2.3.3 Problème C	144
6.2.4 Exemples d'applications	145
6.2.4.1 Transports	145
6.2.4.2 Récréations mathématiques	145

6.2.4.3 Chemins de fiabilité maximale	146
6.2.4.4 Le problème du caboteur	146
6.2.5 Graphe-exemple et notations	147
6.3 Algorithmes à fixation d'étiquettes	147
6.3.1 Algorithme de Dijkstra	147
6.3.1.1 Version de base sans tas	147
6.3.1.2 Version avec tas	152
6.3.2 Algorithme de Sedgewick et Vitter	154
6.3.3 Algorithmes à buckets	155
6.3.3.1 Principes généraux	155
6.3.3.2 Algorithme à buckets de largeur 1	156
6.3.3.3 Procédure Pascal Bucket	159
6.3.3.4 Aperçu sur d'autres algorithmes	161
6.4 Algorithmes à correction d'étiquettes	161
6.4.1 Algorithme de Bellman	161
6.4.1.1 Enoncé de l'algorithme	161
6.4.1.2 Procédure Pascal Bellman	163
6.4.1.3 Exemple d'exécution	164
6.4.2 Algorithme FIFO	164
6.4.2.1 Enoncé de l'algorithme	164
6.4.2.2 Procédure Pascal FIFO	165
6.4.3 Algorithme de d'Esopo et Pape	166
6.4.3.1 Enoncé de l'algorithme	166
6.4.3.2 Procédure Pascal Esopo	167
6.4.4 Algorithme de Floyd	167
6.4.4.1 Enoncé de l'algorithme	167
6.4.4.2 Procédure Pascal Floyd	169
6.4.4.3 Exemple	169
6.5 Application aux ordonnancements	172
6.5.1 Algorithme de Bellman et graphes sans circuits	172
6.5.2 Ordonnements et plus longs chemins	174
6.5.3 Procédure Pascal Schedule	175
6.5.4 Exemple	176
6.6 Evaluation des algorithmes	178
6.7 Listing de l'interface de GPPath	180
6.8 Références	181
Chapitre 7 - Problèmes de flots et couplages	183
7.1 Introduction	183
7.2 Problème du flot maximal	184
7.2.1 Concepts de base	184
7.2.1.1 Définition du problème en termes de graphes	184
7.2.1.2 Formulation en termes de programme linéaire	185
7.2.1.3 Restrictions et cas particuliers	185

7.2.1.4 Coupes et graphes d'écart	187
7.2.2 Exemples d'applications des flots	189
7.2.2.1 Un problème de flotte	189
7.2.2.2 Fiabilité dans les réseaux	189
7.2.2.3 Un problème d'ordonnancement	190
7.2.3 Algorithme de Ford et Fulkerson	190
7.2.3.1 Principe	190
7.2.3.2 Algorithme en pseudo-code et exemple	192
7.2.3.3 Complexité	195
7.2.3.4 Procédure Pascal Fulkerson	196
7.2.3.5 Application à un problème d'adduction d'eau	198
7.2.4 Algorithme d'échelonnement des capacités	202
7.2.5 Algorithme des distances estimées au puits	203
7.2.5.1 Principes	203
7.2.5.2 Complexité	206
7.2.5.3 Algorithme détaillé en pseudo-code	207
7.2.5.4 Procédure Pascal Ahuja	211
7.2.6 Problèmes de flots compatibles	214
7.2.6.1 Calcul d'un flot compatible	214
7.2.6.2 Flot maximal avec capacités minimales sur les arcs	216
7.2.7 Comparaison des algorithmes	216
7.3 Problèmes de flot de coût minimal	218
7.3.1 Formulation du problème	218
7.3.2 Graphe d'écart - Condition d'optimalité	219
7.3.3 Exemples d'applications	219
7.3.3.1 Problèmes de transport et de transbordement	219
7.3.3.2 Un problème d'ordonnancement	220
7.3.4 Algorithme de Busacker et Gowen	221
7.3.4.1 Principe et complexité	221
7.3.4.2 Algorithme en pseudo-code	223
7.3.4.3 Procédure Pascal Busacker	225
7.3.4.4 Exemple d'utilisation et performances	227
7.4 Problèmes de couplage	229
7.4.1 Définitions et principaux problèmes	229
7.4.2 Principes généraux des algorithmes de couplage	230
7.4.2.1 Recherches de CAA	230
7.4.2.2 Conversion en problèmes de flot	231
7.4.3 Exemples d'applications	232
7.4.3.1 Le problème de bin packing	232
7.4.3.2 Affectation de postes de travail	233
7.4.4 Algorithme pour le couplage biparti maximal	233
7.4.4.1 Principes et algorithme théorique	233
7.4.4.2 Procédure Pascal BipMatch	235
7.4.5 Un algorithme pour le problème d'affectation	237
7.4.6 Evaluations des algorithmes de couplage	239
7.4.6.1 Couplage biparti de cardinal maximal	239

7.4.6.2 Couplage biparti de coût minimal	242
7.5 Listing de l'interface de GPFlow	243
7.6 Références	243

Chapitre 8 - Arbres et arborescences 245

8.1 Introduction	245
8.2 Définitions - Enoncés des problèmes	245
8.2.1 Arbres et arborescences	245
8.2.2 Le problème de l'arbre de poids minimal	246
8.2.3 Le problème de l'arborescence de poids minimal	247
8.3 Exemples d'applications	247
8.3.1 Construction de lignes électriques	247
8.3.2 Réseau d'irrigation par gravité	248
8.4 Le problème de l'ARPM	248
8.4.1 Théorème d'optimalité des ARPM	248
8.4.2 Algorithme de Prim	249
8.4.2.1 Principe	249
8.4.2.2 Convergence et optimalité	249
8.4.2.3 Une implémentation efficace	250
8.4.2.4 Procédure Pascal Prim	251
8.4.3 Algorithme de Kruskal	252
8.4.3.1 Principe	252
8.4.3.2 Convergence et optimalité	253
8.4.3.3 Une implémentation efficace	253
8.4.3.4 Procédure Pascal Kruskal	259
8.4.4 Utilisation et comparaison des procédures Prim et Kruskal	261
8.5 Arborescence de poids minimal	263
8.5.1 Principes de l'algorithme d'Edmonds	263
8.5.2 Un exemple	265
8.5.3 Une implémentation efficace	266
8.5.3.1 Gestion des graphes contractés et des graphes partiels	266
8.5.3.2 Détection et contraction des circuits	268
8.5.3.3 Récupération de l'arborescence et de son poids	270
8.5.3.4 Algorithme résultant et complexité	270
8.5.4 Procédure Pascal Edmonds	271
8.5.5 Exemple d'utilisation	274
8.6 Listing de la partie interface de GPTree	275
8.7 Références	275

Chapitre 9 - Parcours eulériens et hamiltoniens 277

9.1 Introduction	277
9.2 Parcours eulériens et chinois	277
9.2.1 Définitions et conditions d'existence	277

9.2.2 Parcours eulérien d'un graphe non orienté	279
9.2.2.1 Principe	279
9.2.2.2 Procédure Pascal EulerChain	282
9.2.3 Parcours eulérien d'un graphe orienté	285
9.2.3.1 Principe	285
9.2.3.2 Procédure Pascal EulerPath	286
9.2.4 Exemple d'utilisation d'EulerChain et EulerPath	287
9.2.5 Postier chinois non orienté	289
9.2.5.1 Principe	289
9.2.5.2 Algorithme et complexité	290
9.2.6 Postier chinois orienté	290
9.2.6.1 Principe	290
9.2.6.2 Algorithme et complexité	291
9.2.6.3 Procédure Pascal Postman	292
9.2.6.4 Exemple d'utilisation	295
9.3 Le problème du voyageur de commerce (PVC)	296
9.3.1 Le PVC comme archétype du problème difficile	296
9.3.2 Exemples d'application du PVC	298
9.3.2.1 Un problème de perçage de circuit imprimé	298
9.3.2.2 Un problème de fabrication de peinture	298
9.3.3 Heuristiques gloutonnes	299
9.3.3.1 Principes	299
9.3.3.2 Procédures Pascal	300
9.3.4 Recherches locales	303
9.3.4.1 Principe et algorithme	303
9.3.4.2 Procédure Pascal TwoOpt	306
9.3.5 Méthode de recuit simulé	307
9.3.5.1 Principe et algorithme	307
9.3.5.2 Procédure Pascal SimAn	308
9.3.6 Méthode taboue	308
9.3.6.1 Principe et algorithme	308
9.3.6.2 Procédure Pascal Tabu	311
9.3.7 Evaluation des heuristiques pour le PVC	313
9.4 Listing de l'interface de GPTour	316
9.5 Références	317
 Chapitre 10 - Problèmes de coloration	 319
10.1 Introduction	319
10.2 Généralités sur la coloration de graphes	319
10.2.1 Définition des problèmes et complexité	319
10.2.2 Quelques propriétés simples	320
10.2.2.1 Coloration des arêtes	320
10.2.2.2 Coloration des sommets	320
10.3 Deux exemples d'applications	321

10.3.1 Problème d'emploi du temps	321
10.3.2 Carrés latins	322
10.4 Heuristiques séquentielles	322
10.4.1 Méthode générique	322
10.4.2 Deux bons ordres	323
10.4.3 Algorithme DSatur	324
10.4.4 Procédures Pascal	325
10.5 Méthode exacte	329
10.5.1 Principes	329
10.5.2 Procédure Pascal Backtrack	331
10.6 Méthode de recuit simulé	333
10.7 Recherche taboue	335
10.7.1 Principes	335
10.7.2 Procédure Pascal TabuCol	336
10.8 Evaluation des méthodes de coloration	339
10.9 Listing de l'interface de GPColor	343
10.10 Références	344
 Annexe 1 - Listing de l'unité GPCore	 345
 Annexe 2 - Bibliographie	 371
 Index	 377