CAHIER DU LAMSADE

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision
(Université Paris-Dauphine)
Unité de Recherche Associée au CNRS ESA 7024

ON-LINE VERTEX-COVERING

CAHIER N° 183
septembre 2001

Marc DEMANGE [1]
Vangelis PASCHOS [2]

received: July 2001.

[1] *ESSEC, BP 105, 95021 Cergy-Pontoise Cedex, France (demange@essec.fr).*

[2] *LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France (paschos@lamsade.dauphine.fr).*

LABORATOIRE D'ANALYSE ET MODÉLISATION DE SYSTÈMES POUR L'AIDE À LA DÉCISION (LAMSADE)

(Unité de Recherche Associée au CNRS UMR 7024)

Directeur : Vangelis PASCHOS

PUBLICATIONS DISPONIBLES [1] [2]

CAHIERS

N° 90 – A. LISSER, P. TOLLA : Variants of Karmarkar's algorithm (*mars 1989, 33 pages*).

N° 91 – D. BOUYSSOU : Problèmes de construction de critères (*avril 1989, 32 pages*).

N° 94 – D.K. DESPOTIS, D. YANNACOPOULOS : Méthode d'estimation d'utilités additives concaves en programmation linéaire multiobjectifs (*septembre 1989, 26 pages*).

N° 95 – B. ROY, R. SLOWINSKI : Criterion of distance between technical programming and socio-economic priority (*deuxième version révisée : mars 1990, 29 pages*).

N° 97 – B. ROY : Science de la décision ou science de l'aide à la décision ? (*troisième version révisée : février 1992, 48 pages*).

N° 98 – R. SLOWINSKI : Outranking-based interactive procedure for multiple objective programs (*avril 1990, 24 pages*).

N° 99 – H. M'SILTI, P. TOLLA : An interactive multiobjective nonlinear procedure (*avril 1990, 31 pages*).

N° 100 – J.M. MARTEL, K. ZARAS : Dominance stochastique en analyse multicritère face au risque (*novembre 1990, 32 pages*).

N° 101 – D. BOUYSSOU, P. PERNY : Ranking methods for valued preference relations: A characterization of a method based on leaving and entering flows (*décembre 1990, 22 pages*).

N° 102 – S. PINSON : A multi-expert architecture for credit risk assessment (*mars 1991, 41 pages*).

N° 104 – J.M. MARTEL, B. AOUNI : Méthode multicritère de choix d'un emplacement : Le cas d'un aéroport dans le Nouveau Québec (*avril 1991, 45 pages*).

N° 106 – B. ROY, D. BOUYSSOU : Decision-aid: an elementary introduction with emphasis on multiple criteria (*novembre 1991, 30 pages*).

N° 107 – J. GONDZIO : An advanced implementation of Cholesky factorization for computing projections in interior point methods of large scale linear programming (*décembre 1991, 42 pages*).

N° 108 – J. GONDZIO, D. TACHAT : The design and application of the IPMLO – A FORTRAN library for linear optimization with interior point methods (*janvier 1992, 38 pages*).

N° 110 – R. SLOWINSKI : A generalization of the indiscernibility relation for rough sets analysis of quantitative information (*mai 1992, 34 pages*).

---

[1] *Pour se les procurer, s'adresser au LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16 (Bureau P 613 – Téléphone : 33 (0)1 44 05 40 89 – e.mail: champ@lamsade.dauphine.fr).*

[2] *Une liste exhaustive des publications peut être consultée sur le serveur web du LAMSADE : http://www.lamsade.dauphine.fr.*

N° 111 – **V. MOUSSEAU** : Are judgments about relative importance of criteria dependent or independent of the set of alternatives? An experimental approach (*mai 1992, 42 pages*).

N° 112 – **V. Th. PASCHOS, L. RENOTTE** : Approximability preserving reductions for NP-complete problems (*janvier 1993, 37 pages*).

N° 113 – **H. M'SILTI, P. TOLLA** : A new heuristic method for solving global optimization problem (*janvier 1993, 34 pages*).

N° 114 – **L. DUCKSTEIN, W. TREICHEL, S. EL MAGNOUNI** : Multicriterion analysis of groundwater management alternatives (*janvier 1993, 40 pages*).

N° 115 – **A. JASZKIEWICZ, R. SLOWINSKI** : The light beam search over a non-dominated set – an interactivepro-cedure for multiple-objective analysis of linear and non-linear programs (*juin 1993, 29 pages*).

N° 116 – **M. DEMANGE, V.Th. PASCHOS** : On an approximation measure founded on the links between optimization and polynomial approximation theory (*septembre 1993, 38 pages*).

N° 117 – **C.A. BANA e COSTA, J.C. VANSNICK** : Sur la quantification des jugements de valeur : l'approche MACBETH (*septembre 1993, 47 pages*).

N° 118 – **O. CORDEIRO-NETTO, E. PARENT, L. DUCKSTEIN** : Evaluation of two MCDM techniques for staging the multicriterion design of long-term water supply in Southern France (*septembre 1993, 35 pages*).

N° 119 – **A. SCHAAL, H. M'SILTI, P. TOLLA** : Modélisation et système interactif de planification de stocks liés (*septembre 1993, 25 pages*).

N° 120 – **J.C. POMEROL, B. ROY, C. ROSENTHAL-SABROUX, A. SAAD** : An "intelligent" DSS for the multicri-teria evaluation of railway timetables (*octobre 1993, 33 pages*).

N° 121 – **V.Th. PASCHOS** : A relation between the approximated versions of minimum set covering, minimum vertex covering and maximum independent set (*octobre 1993, 21 pages*).

N° 122 – **V. MOUSSEAU** : Compensatoriness of preferences in matching and choice (*janvier 1994, 27 pages*).

N° 123 – **A. TSOUKIAS, Ph. VINCKE** : A new axiomatic foundation of the partial comparability theory (*février 1994, 38 pages*).

N° 124 – **B. ROY, H. HASSAN** : Intérêt des méthodes multicritères pour l'aide à la décision concernant les investissements en matière d'infrastructures de transports (*deuxième version révisée : octobre 1994, 25 pages*).

N° 125 – **J. BLOT, W. FERNANDEZ de la VEGA, V. Th. PASCHOS, R. SAAD** : Average case analysis of greedy algorithms for optimisation problems on set systems (*octobre 1994, 40 pages*).

N° 126 – **V. MOUSSEAU** : Eliciting information concerning the relative importance of criteria (*janvier 1995, 37 pages*).

N° 127 – **R. FAIZ, M. GONDRAN** : Modélisation formelle de textes exprimant des concepts temporels : Cas des textes juridiques (*juin 1995, 42 pages*).

N° 128 – **M. DEMANGE, V. PASCHOS** : Exact and approximation results on maximum independent set and minimum vertex covering – Graphs with great stability number (*juin 1995, 38 pages*).

N° 129 – **A. MARSON, A. TSOUKIÀS** : Venice safeguard: the decision process and how to provide decision aid (*juin 1995, 39 pages*).

N° 130 – **B. ROY, D. VANDERPOOTEN** : The European School of MCDA: Emergence, basic features and current works (*juin 1995, 33 pages*).

N° 131 – **B. ROY, V. MOUSSEAU** : A theoretical framework for analysing the notion of relative importance of criteria (*juin 1995, 36 pages*).

N° 132 – **C. ZOPOUNIDIS, A.I. DIMITRAS, L. LE RUDULIER** : A multicriteria approach for the analysis and prediction of business failure in Greece (*novembre 1995, 28 pages*).

N° 133 – **C. HAOUCHE, C. ROSENTHAL-SABROUX** : KBS life-cycle and validation (*janvier 1996, 26 pages*).

N° 134 – **P. JOURNÉE, P. PERNY, D. VANDERPOOTEN** : Une méthodologie multicritère pour la vérification des accords sur le contrôle des armements en Europe (*février 1996, 37 pages*).

N° 135 – **W. FERNANDEZ de la VEGA, V.Th. PASCHOS, A.N. STAFYLOPATIS** : Average-case complexity for the execution of recursive definitions on relational databases (*février 1996, 25 pages*).

N° 136 – **V. GABREL, D. VANDERPOOTEN** : Generation and selection of efficient paths in a multiple criteria graph: the case of daily planning the shots taken by a satellite with an interactive procedure (*février 1996, 22 pages*).

N° 137 – **V.Th. PASCHOS, M. DEMANGE** : Relative hardness of constructive–non-constructive approximation: the case of maximum independent set problem (*avril 1996, 29 pages*).

N° 138 – **F. BALBO, P. MORAÏTIS, S. PINSON** : AMCA : Approche Multicritère pour la Coopération entre Agents (*juin 1996, 29 pages*).

N° 139 – **A. TSOUKIÀS** : A first order, four-valued weakly paraconsistent logic (*juin 1996, 36 pages*).

N° 140 – **V. MOUSSEAU, R. SLOWINSKI** : Inferring an ELECTRE TRI model from assignment examples (*juin 1996, 30 pages*).

N° 141 – **R. MIEŃKO, J. STEFANOWSKI, K. TOUMI, D. VANDERPOOTEN** : Discovery-oriented induction of decision rules (*septembre 1996, 29 pages*).

N° 142 – **A. PARTOUCHE** : Comparaison de deux approches sur un problème de planification d'horaires : la construction de cycles de travail (*octobre 1996, 33 pages*).

N° 143 – **A.B. FERHAT, A. JASZKIEWICZ** : Trichotomy based procedure for multiple criteria choice problems (*décembre 1996, 26 pages*).

N° 144 – **B. ROY** : Une lacune en RO-AD – Les conclusions robustes (*mai 1997, 32 pages*).

N° 145 – **A. SCHAAL, H. M'SILTI, P. TOLLA** : Une approche hybride de résolution de problèmes linéaires généraux en nombres entiers (*mai 1997, 26 pages*).

N° 146 – **E. JACQUET-LAGRÈZE, D. MONTAUT, A. PARTOUCHE** : The shift scheduling problem: Different formulations and solution methods (*juillet 1997, 30 pages*).

N° 147 – **H. M'SILTI, P. TOLLA, A. SCHAAL** : Comparison of simulated annealing and genetic algorithm within a hybrid method for general integer problems (*juillet 1997, 28 pages*).

N° 148 – **D. MAILLES, J. KADAOUI-ABBASSI** : Résolution analytique de file d'attente avec arrivées générées par parallélisation (*septembre 1997, 26 pages*).

N° 149 – **J. FIGUEIRA, M.E. CAPTIVO, J. CLIMACO** : Sur le problème des chemins bi-critères : Une approche interactive a posteriori (*novembre 1997, 42 pages*).

N° 150 – **V. MOUSSEAU, J. FIGUEIRA, J.Ph. NAUX** : Using assignment examples to infer weights for ELECTRE TRI method: Some experimental results (*décembre 1997, 40 pages*).

N° 151 – **A. PARTOUCHE** : A hybrid method for the general rostering problem with a mixed and homogeneous workforce (*décembre 1997, 35 pages*).

N° 152 – **S. GRECO, B. MATARAZZO, R. SLOWINSKI, A. TSOUKIÀS** : Exploitation of a rough approximation of the outranking relation (*décembre 1997, 32 pages*).

N° 153 – **Ph. FORTEMPS, J. TEGHEM** : Multi-objective fuzzy linear program: the MOFAC method (*janvier 1998, 26 pages*).

N° 154 – **A. PARTOUCHE, D. VANDERPOOTEN** : Une classe polynomiale de problèmes de construction de vacations (*février 1998, 32 pages*).

N° 155 – **M.J. BLIN, A. TSOUKIÀS** : Multicriteria methodology contribution to the software quality evaluation (*mai 1998, 37 pages*).

N° 156 – **I. STAMELOS, A. TSOUKIÀS** : Software evaluation problem situations (*septembre 1998, 34 pages*).

N° 157 – **S. AKNINE, S. PINSON** : Conception orientée agent du travail coopératif (*octobre 1998, 35 pages*).

N° 158 – **C. ARONDEL, Ph. GIRARDIN** : Sorting cropping systems on the basis of their impact on groundwater quality (*novembre 1998, 38 pages*).

N° 159 – L. ALFANDARI : The greedy approach for the crew pairing problem: some cases of polynomial subproblems (*janvier 1999, 19 pages*).

N° 160 – V. PASCHOS : Approximation of weighted hereditary induced-subgraph maximization problems (*janvier 1999, 18 pages*).

N° 161 – A. LIKAS, V. PASCHOS : A new greedy-solution representation for the traveling salesman problem (*juin 1999, 18 pages*).

N° 162 – L. ALFANDARI, V. PASCHOS : Master-slave strategy and polynomial approximation (*juin 1999, 26 pages*).

N° 163 – M. DEMANGE, V. PASCHOS : Maximum-weight independent set is as "well-approximated" as the unweighted one (*juin 1999, 29 pages*).

N° 164 – M.J. BLIN, F. FABRET : A coordination mechanism to prevent the violation of distributed constraints (*septembre 1999, 37 pages*).

N° 165 – A. NGO THE, A. TSOUKIÀS, Ph. VINCKE : PQI interval orders (*novembre 1999, 35 pages*).

N° 166 – C. MURAT, V. PASCHOS : A priori optimization for the probabilistic maximum independent set problem (*janvier 2000, 34 pages*).

N° 167 – B. ROY : Réflexion sur le thème Quête de l'optimum et aide à la décision (*février 2000, 32 pages*).

N° 168 – A. NGO THE, V. MOUSSEAU : Using assignment examples to infer category limits for the ELECTRE TRI method (*mai 2000, 35 pages*).

N° 169 – B. ROY : Optimisation lexicographique d'une séquence de dysfonctionnements non nécessairement quantitatifs (*mai 2000, 27 pages*).

N° 170 – C. MURAT, V. PASCHOS : The probabilistic minimum vertex covering problem (*juin 2000, 24 pages*).

N° 171 – M. DEMANGE, C. MURAT, V. PASCHOS, S. TOULOUSE : Un nouveau modèle pour la construction de réseau de télécommunication à coût minimum (*juin 2000, 32 pages*).

N° 172 – J. MONNOT, V. PASCHOS, S. TOULOUSE : Differential approximation results for traveling salesman problem (*juin 2000, 23 pages*).

N° 173 – M. DEMANGE, V. PASCHOS : Approximation polynomiale : notions de difficulté et leur impact pour étudier la structure de NP (*juin 2000, 28 pages*).

N° 174 – L. DIAS, V. MOUSSEAU, J. FIGUEIRA, J. CLÍMACO : An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE TRI (*juin 2000, 35 pages*).

N° 175 – C.A. BANA E COSTA, J.C. VANSNICK : Une critique de base de l'approche de Saaty : Mise en question de la méthode de la valeur propre maximale (*février 2001, 23 pages*).

N° 176 – J. MONNOT : The maximum Hamiltonian path problem with specified endpoint(s) (*mars 2001, 23 pages*).

N° 177 – M. DEMANGE, V. PASCHOS : Towards a general formal framework for polynomial approximation (*mars 2001, 49 pages*).

N° 178 – J. CLÍMACO, L. DIAS, J. FIGUEIRA, C. GOMES da SILVA, V. MOUSSEAU : Resolving inconsistencies among constraints on the parameters of an MCDA model (*avril 2001, 24 pages*).

N° 179 – M.J. BLIN, F. FABRET : A framework integrating collaboration and coordination (*avril 2001, 28 pages*).

N° 180 – V. PASCHOS, C. BAZGAN : Differential approximation for satisfiability and related problems (*avril 2001, 19 pages*).

N° 181 – M.E. CAPTIVO, J. CLIMACO, J. FIGUEIRA, E. MARTINS, J.L. SANTOS : Solving multiple criteria {0, 1}-knapsack problems using a labeling algorithm (*mai 2001, 33 pages*).

N° 182 – D. de WERRA, M. DEMANGE, V. PASCHOS, J. MONNOT : Time slot scheduling of compatible jobs (*septembre 2001, 37 pages*).

N° 183 – M. DEMANGE, V. PASCHOS : On-line vertex-covering (*septembre 2001, 36 pages*).

N° 184 – A. NGO THE, A. TSOUKIÀS : Numerical representation of *PQI* interval orders (*septembre 2001, 38 pages*).

DOCUMENTS

N° 50 – **M. DEZA, M. LAURENT** : Facets of the complete cut cone (*avril 1989, 57 pages*).

N° 51 – **B. ROY** : Decision-aid and decision-making (*mai 1989, 30 pages*).

N° 53 – **B. ROY** : The outranking approach and the foundations of ELECTRE methods (*juillet 1989, 40 pages*).

N° 55 – **G. BAROUCH** : Les conflits au risque des théories de Palo Alto : Le cas de l'aménagement de la Cisse (*septembre 1989, 82 pages*).

N° 56 – **E. KONARZEWSKA** : BIPOLAR: Multiple criteria decision aid using BIPOLAR reference system (*septembre 1989, 60 pages*).

N° 57 – **M. DEZA, M. LAURENT** : Facets for the complete cut cone II: Clique-Web inequalities (*octobre 1989, 54 pages*).

N° 61 – **A. DAVID, J.L. GIORDANO** : Rôle des représentations graphiques et analytiques simples dans le développement de la coopération Marketing/Bureau d'Etudes chez un grand constructeur automobile (*avril 1990, 53 pages*).

N° 62 – **Ph. VINCKE** : Exploitation d'une relation non valuée dans une problématique de rangement complet (*janvier 1991, 47 pages*).

N° 65 – **B. ROY, R. SLOWINSKI, W. TREICHEL** : Multi-criteria programming of rural water supply systems (*mai 1991, 54 pages*).

N° 73 – **G.R. D'AVIGNON** : Démarche d'aide à la décision d'un plan directeur : Le cas des palais de justice (*mars 1992, 65 pages*).

N° 74 – **W. YU** : ELECTRE TRI – Aspects méthodologiques et manuel d'utilisation (*avril 1992, 100 pages*).

N° 78 – **Ph. VAN ASBROECK, D. CHAMP-BRUNET** : Un thésaurus français-anglais de termes d'aide multicritère à la décision (mars 1993) (*deuxième version révisée : juillet 1994, 85 pages*).

N° 80 – **C.A. BANA e COSTA** : Les problématiques dans le cadre de l'activité d'aide à la décision (*septembre 1993, 68 pages*).

N° 81 – **C.A. BANA e COSTA, Ph. VINCKE** : Comment prendre en compte l'imprécision des taux de substitution dans un modèle additif d'agrégation des préférences (*novembre 1993, 59 pages*).

N° 82 – **V. GABREL** : La planification à moyen terme des prises de vue à réaliser par un satellite défilant d'observation de la terre : modélisation et solutions algorithmiques (*janvier 1994, 42 pages*).

N° 83 – **L. DUCKSTEIN, G. EDER, H.P. NACHTNEBEL** : Ranking water resources alternatives and criteria by multicriterion Q-analysis: An Austrian case study (*juin 1994, 49 pages*).

N° 84 – **C.J. MAESTRE** : A propos des relations entre culture et gestion – Une illustration opérationnelle : Gestion par produit, gestion par processus (*juin 1994, 73 pages*).

N° 85 – **D. VALLÉE, P. ZIELNIEWICZ** : ELECTRE III-IV, version 3.x – Aspects méthodologiques (tome 1) (*juillet 1994, 67 pages*).

N° 86 – **Ph. VAN ASBROECK, D. CHAMP-BRUNET** : An English-French thesaurus of multicriteria decision aid indexing terms (*octobre 1994, 85 pages*).

N° 87 – **J.D. LÉNARD** : Approche multicritère de la gestion des approvisionnements – Aide à la décision et pilotage global d'un ensemble d'articles (*mai 1995, 79 pages*).

N° 89 – **P. MORAÏTIS** : Intelligence artificielle distribuée et systèmes multi-agents : un état de l'art (*novembre 1995, 75 pages*).

N° 90 – S. GANÇARSKI, M. ZAMFIROIU : Le module ZGraph : un gestionnaire de vues groupées sur un graphe pour le SGBD O_2 (*décembre 1995, 35 pages*).

N° 91 – Numéro Spécial Escargot d'Or 1996 (*février 1996, 38 pages*).

N° 92 – V.Th. PASCHOS, P. GRISONI, M. DEMANGE : Approximation algorithms for some combinatorial optimization problems (*mars 1996, 30 pages*).

N° 93 – Ph. VALLIN : Détermination d'une période économique robuste dans le cadre du modèle de Wilson (*mai 1996, 31 pages*).

N° 94 – J. FIGUEIRA, H. M'SILTI, P. TOLLA : Application d'un algorithme de sous-gradient pour explorer le voisinage d'un point satisfaisant dans le cadre des problèmes de flot multicritères (*juin 1996, 61 pages*).

N° 95 – M. KOSTKOWSKI, R. SLOWINSKI : UTA+ application (v 1.20) – User's manual (*juillet 1996, 32 pages*).

N° 96 – S. HADDAD, P. MOREAUX : Aggregation and decomposition for performance evaluation of synchronous product of high level Petri nets (*septembre 1996, 64 pages*).

N° 97 – Y. SISKOS, N.F. MATSATSINIS, G. BAOURAKIS : Multicriteria analysis in agricultural marketing – The case of French olive oil market (*décembre 1996, 32 pages*).

N° 99 – J. FIGUEIRA, H. M'SILTI, P. TOLLA : Recent developments in the multi-criteria network flow problems (*février 1997, 38 pages*).

N° 100 – Numéro Spécial Escargot d'Or 1997 (*mai 1997, 37 pages*).

N° 101 – F. BEN ABDELAZIZ, P. LANG, R. NADEAU : Notions d'efficacité en programmation linéaire multiobjectif stochastique (*mai 1997, 40 pages*).

N° 102 – S. HADDAD, P. MOREAUX : Aggregation and decomposition for performance evaluation of asynchronous product of high level Petri nets (*mai 1997, 51 pages*).

N° 103 – J. PEREIRA : Flexibilité dans les systèmes de production : Une approche relationnelle et dynamique (*septembre 1997, 31 pages*).

N° 104 – B. ROY : L'aide à la décision aujourd'hui : Que devrait-on en attendre ? (*novembre 1997, 67 pages*).

N° 105 – A. PAPAYANNAKIS : Analyse spatiale et modélisation statistique du trafic du réseau autobus de la banlieue parisienne (*janvier 1998, 77 pages*).

N° 106 – J.M. MARTEL, A. GUITOUNI, I. ABI-ZEID, M. BÉLANGER, Z. CHRAYGANE, K. ZARAS : Construction d'une famille de critères dans le cadre du projet CESA (*mars 1998, 45 pages*).

N° 107 – Rapport Scientifique (1995-1997) (*juin 1998, 292 pages*).

N° 107 bis – Extrait du Rapport Scientifique (1995-1997) (*juillet 1998, 125 pages*).

N° 108 – Numéro Spécial Escargot d'Or 1998 (*juin 1998, 35 pages*).

N° 109 – B. ROY, J. FIGUEIRA : Détermination des poids des critères dans les méthodes de type ELECTRE avec la technique de Simos révisée (*juillet 1998, 45 pages*).

N° 110 – M.P. HUGET, S. PINSON : Une typologie des modèles d'agents (*janvier 1999, 41 pages*).

N° 111 – V. MOUSSEAU, R. SLOWINSKI, P. ZIELNIEWICZ : ELECTRE TRI 2.0a – Methodological guide and user's manual (*février 1999, 78 pages*).

N° 112 – V. MOUSSEAU, B. ROY, I. SOMMERLATT : Elaboration d'un outil d'aide à la décision en vue de l'évolution de la tarification des transports publics en Ile-de-France (*mai 1999, 55 pages*).

N° 113 – E. PASCHETTA, A. TSOUKIÀS : A real world MCDA application: evaluating software (*juin 1999, 5[illegible] pages*).

N° 114 – Numéro spécial Escargot d'Or 1999 (*juin 1999, 34 pages*).

N° 115 – L. VERCOUTER, N. FRANCHESQUIN, J.P. TREUIL, S. PINSON : Simulation multi-agent des processu[illegible] de négociation en agronomie : Un exemple de gestion locale sous contrainte collective (*juillet 1999, 49 pages*[illegible]

N° 116 – R. AZIBI, D. VANDERPOOTEN : L'agrégation de conséquences dispersées pour la construction [illegible] critères : L'étude d'aménagement Loire moyenne (*juillet 1999, 33 pages*).

N° 117 – A. TSOUKIÀS : Sur la généralisation des concepts de concordance et de discordance en aide multicritère à la décision (*novembre 1999, 87 pages*).

N° 118 – A. AÏT YOUNES, R. AZIBI, B. ROY : ELECTRE IS – Manuel d'utilisation (Tome 1) (*mars 2000, 41 pages*).

N° 118 bis – A. AÏT YOUNES, R. AZIBI, B. ROY : ELECTRE IS - Manuel d'utilisation (Tome 2) (*juin 2000, 73 pages*).

N° 119 – Rapport Scientifique (1996-1999) (*mai 2000, 288 pages*).

N° 120 – Numéro Spécial Escargot d'Or 2000 (*juin 2000, 33 pages*).

N° 121 – V. PASCHOS, C. MURAT, V. GABREL : La recherche opérationnelle pour la planification de prises de vue par satellite (*septembre 2001, 41 pages*).

Table of Contents

Résumé ii

Abstract ii

1 Introduction 1

2 Basic definitions and notations 2

3 On-line vertex-covering with $t = n$ 3

3.1 On the approximation ratio of any minimal vertex-covering algorithm against maximum matching . . . 3

3.2 An on-line algorithm for the case $t = n$ . . . 5

3.3 Lower bounds on the competitivity of any algorithm for the case $t = n$ . . . 5

4 On-line vertex covering with $n > t \geqslant 2$ 7

4.1 On-line vertex covering with $n > t > 2$ . . . 7

4.2 On-line vertex-covering with $t = 2$ . . . 10

4.2.1 G has no isolated vertices . . . 10

4.2.2 Clusters have no isolated vertices . . . 11

4.3 Lower bounds for the competitivity ratio . . . 12

4.3.1 Case $t > 2$ . . . 12

4.3.2 Limits on the competitivity for $t = 2$ . . . 12

5 Allowing backtracking 14

6 A simple on-line model based upon edges 17

7 Conclusions 18

A Proof of theorem 6 20

Recouvrement *on-line* d'un graphe

Résumé

En algorithmique on-line, l'instance d'un problème est révélée en différentes étapes. A l'issue de chacune d'elles on doit prendre des décisions irrévocables sur la partie qui vient d'être révélée (qu'elle est la contribution de cette partie à la solution que l'on construit ?). Nous commençons par étudier le problème de couverture de sommets sous deux modèles on-line qui correspondent à deux manières différentes de révéler les sommets. Le premier impose que le graphe instance est révélé sommet par sommet. Dans le second modèle le graphe est révélé par paquets, chaque paquet étant un sous-graphe induit de l'instance complète. Puis, dans le modèle par paquets, nous relaxons la contrainte que le choix des sommets du paquet à inclure dans la solution est irrévocable. Plus précisément nous autorisons un *backtracking* : une décision sur un sommet peut être modifiée modulo un coût qui dépend du nombre global de sommets modifiés. Enfin nous étudions un modèle simple où les instances sont révélées arête par arête.

Mots-clé : algorithme *on-line*, recouvrement d'un graph, rapport de compétitivité, rapport d' approximation.

On-line vertex-covering

Abstract

In on-line computation, the instance of a problem is revealed step-by-step and one has, at the end of each step, to irrevocably decide on the part of the final solution dealing with this step. We first study the minimum vertex-covering problem under two on-line models corresponding to two different ways vertices are revealed. The former one implies that the input-graph is revealed vertex-by-vertex. The second model implies that the input-graph is revealed per clusters, i.e., per induced subgraphs of the final graph. Under the cluster-model, we then relax the constraint that the choice of the part of the final solution dealing with each cluster has to be irrevocable, by allowing backtracking. We assume that one can change decisions upon a vertex membership of the final solution, this change implying, however, some cost depending on the number of the vertices changed. Finally we study simple model where instance is revealed edge-by-edge.

Keywords: on-line algorithm, vertex-covering, competitivity ratio, approximation ratio.

1 Introduction

On-line computation is very natural in operational research since there exist situations modeled as problems for which the final data-set is not a priori known. In other words, data are revealed step-by-step. Frequently, when one tries to solve such problems many types of constraints (for example, deadlines on the final solution delivery, deadlines on the implementation of the solution computed) force her/him to start problem's solution before the whole set of data is completely revealed. On the other hand, these constraints may be strict enough forcing so the problem solver to irrevocably decide on the part of the final solution dealing with each part of data revealed, or may be relatively weak, allowing her/him to go back over decisions previously taken about the partial solution computed at each step.

Let Π be an **NP** optimization graph-problem. The *on-line* version of Π, denoted by LΠ, is the pair $(\Pi, \boldsymbol{R})$ where $\boldsymbol{R}$ is a set of rules dealing with:

1. information on the value of some parameters of the final graph,
2. how the final graph is revealed.

An *on-line* algorithm $\mathtt{A}$ decides at each step which of the data (vertices or edges) revealed during this step will belong to the final solution. Its performance is measured in terms of the so-called competitivity ratio $c_{\mathtt{A}}$ defined, for an instance G and a set $\boldsymbol{R}$, as the ratio of the worst[1] value of the solution computed by it when running on G to the value of a solution computed off-line, i.e., by an algorithm running once the final graph is completely known. In this paper we deal with deterministic on-line algorithms. The above measure of the quality of an on-line algorithm has been originally introduced in [12], in order to study a fundamental computer science problem, the paging problem. Interesting reviews on on-line algorithms (under competitive analysis) are presented in [9, 6]. Also, an interested financial application of on-line computing is solved in [5]. On-line graph problems studied until now are, to our knowledge, the traveling salesman ([1]), the graph-coloring ([7, 10, 8] and the independent set ([4]).

Let us consider a company receiving manufacturing orders from its clients. These orders have to be accepted or rejected as soon as they arrive. Acceptance or rejection of the orders can be provided either immediately, i.e., as soon as any order arrives (*alternative 1*), or at the end, say of each month (*alternative 2.1*), or after a fixed number of orders (say 100 orders) has arrived (*alternative 2.2*). Incompatibilities between orders (due to the fabrication time, the materials required, etc.) are pairwise incompatibilities; so a set of globally compatible orders is an independent set in the order-incompatibility graph (where orders are its vertices). If the objective of the company is to maximize the number of the compatible orders accepted during, say one year's period (or the global profit implied by them), then the problem to be solved is the *on-line maximum independent set problem.* However, assume a public company or even a company operating with privileged clients. In both situations, the company is constrained to accept any order emanating from its clients and has either to manufacture orders by itself, or to use sub-contractors. Then, its objective is not to maximize the profit, but to minimize the subcontracting cost during, for example, one year's period. So, with respect to the incompatibility graph mentioned just above, one has to minimize the complement of a maximum independent set, i.e., a vertex-cover, and the problem to solve is the *on-line minimum vertex-covering.* This is the problem we deal with in this paper.

Minimum vertex-covering problem, denoted by VC in the sequel, is defined as follows: *given a graph $G(V, E)$, compute the minimum-cardinality set $V' \subseteq V$ such that, $\forall v_i v_j \in E$, at least one of the v_i, v_j belongs to V'.* We consider that G (we set $n = |V|$ and suppose n known

[1] Over all the ways G is revealed according to $\boldsymbol{R}$.

at the beginning of the game) is revealed per non-empty clusters, i.e., per induced subgraphs $G_1(V_1, E_1), G_2(V_2, E_2), \ldots$ of G (we denote by n_i the order of G_i, $i = 1, 2, \ldots$). Every time a new cluster G_i is revealed, the edges linking the vertices of G_i with the vertices of G_j, $j < i$ are also revealed. We denote by t the number of clusters needed so that the whole graph is completely revealed.

We first focus ourselves on the case where graph is revealed by means of its vertices and consider $t = n$, i.e., that G is revealed vertex-by-vertex. This is what we have called *alternative 1* in our company model described previously. We establish a general result about the performance of every minimal VC-algorithm (i.e., an algorithm computing a minimal vertex-cover) in comparison with the maximum matching algorithm for VC, informally, the ratio of any minimal vertex-cover against the vertex-cover computed by the maximum matching algorithm is bounded above by $\Delta/2$. Using this result, we establish the competitivity ratio of a very simple but very natural on-line algorithm entering a newly presented vertex v in the covering C under construction, if there exists an edge incident to v (hence revealed together with v) the already revealed endpoint u of which does not belong to C (following our assumption about the way G is revealed, u has arrived before v).

Next, we generalize our study assuming $t < n$ and study the competitivity ratio of (more complicated) on-line algorithms for LVC against an optimal off-line algorithm. Here we distinguish two cases: $2 < t < n$ and $t = 2$. With respect to our company model, the former represents *alternative 2.1*; *alternative 2.2*, not studied here, could represent a situation where all the clusters are of the same order. Then, we analyze the case $t = 2$. This case has, as we shall see, its own mathematical interest. Furthermore, even in the framework of our company model it is very natural. Revisit this model and suppose that in order that the products start to be manufactured, say at the instant $t + 15$, orders have to be arrived at instant t. But for some reasons (e.g., organizational or promotional ones), clients have been granted some extra delay, for example $t + 10$, in order to send orders to be manufactured at $t + 15$. Here, company has to answer in two times: first, for the orders arrived up to instant t and second, for the orders arrived from $t + 1$ to $t + 10$. The incompatibility graphs of these two sets of orders are the two clusters.

We continue the paper by assuming non-irrevocability in the construction of the on-line solution, i.e., by allowing backtracking. This means that the algorithm can interchange a number of vertices in the solution computed by a number of vertices not included in it. But we consider that changes performed imply a cost on the vertices changed. This, in our company example, becomes in deciding, at the last moment, to give some additional manufacturing work in its sub-contractors. But since it does not meet the deadline for these orders, it has to pay some extra cost. We study the competitivity (against an optimal off-line algorithm) of two algorithms under two cost models. The former implies that the cost paid for any change is fixed, while the latter implies that for any vertex changed one has to pay a cost equal to the total number of vertices changed.

Finally, we study a slightly different on-line model, where we assume that the input-graph is revealed edge-by-edge. Together with the arrival of a new edge, are revealed the links of its endpoints with the ones of the edges already revealed. Here also we devise an on-line algorithm and study its competitivity ratio against an optimal off-line one.

2 Basic definitions and notations

The following definitions will be frequently used throughout the paper. For reasons of legibility they are grouped here, before entering to its purely technical part.

Matching. A matching is a set of mutually disjoint edges of G.

Exposed vertices. A vertex is called exposed with respect to a matching M, if it is not endpoint of any edge of M, in other words, if it is not saturated by M.

Augmenting path. A path $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ is augmenting with respect to a maximal matching M if k is even, v_{i_1}, v_{i_k} are exposed with respect to M and if $v_{i_l}v_{i_{l+1}} \in M$, $l = 2q$, $q = 1, \ldots, (k-2)/2$; in other words, the set $M \setminus \{v_{i_l}v_{i_{l+1}} : l = 2q, q = 1, \ldots, (k-2)/2\} \cup \{v_{i_{l+1}}v_{i_{l+2}} : l = 2q, q = 1, \ldots, (k-2)/2\}$ is also a matching with cardinality equal to $|M|+1$.

Independent set. An independent set is a subset of $V' \subseteq V$ such that, for any $(v_i, v_j) \in V' \times V'$, $v_iv_j \notin E$.

Minimal (resp., maximal) set. A set will be called minimal (resp., maximal) with respect to a property π, if it satisfies π, while deletion (resp., insertion) of an element from (resp., in) S results in a set not satisfying π.

Fact 1. **([3])** Any (maximal) independent set is the complement, with respect to V, of a (minimal) vertex-cover. ▮

In what follows, we denote by Δ the maximum degree of G, by $\tau(G)$ the cardinality of a minimum vertex cover of G, by M (resp., M_i) a maximum matching of G (resp., G_i) and by P (resp., P_i) the set of the exposed vertices of G (resp., G_i) with respect to M (resp., M_i). Denote also by $X(M)$ (resp., $X(M_i)$) the set of the endpoints of M (resp., M_i).

Fact 2. **([3])** Consider a graph G, fix a maximal matching M and let $P = V \setminus X(M)$. Then:

1. P is independent for G;
2. $X(M)$ is a vertex-cover of G with $|X(M)| = 2m$. ▮

For $v \in V$, we denote by $\Gamma(v)$ the set of neighbors of v, i.e., $\Gamma(v) = \{u : uv \in E\}$, $\Delta = \max\{|\Gamma(v)| : v \in V\}$.

3 On-line vertex-covering with $t = n$

We first consider that G is revealed into $t = n$ clusters, i.e., vertex-by-vertex. Before specifying an on-line algorithm for this case, we establish a general result for any algorithm (on-line or off-line) computing a minimal vertex-cover, i.e., a vertex-cover that cannot be reduced by elimination of some of its vertices.

3.1 On the approximation ratio of any minimal vertex-covering algorithm against maximum matching

We denote by `MAX_MATCHING` an algorithm computing a maximum matching M of G (the problem of finding a maximum matching of a graph is polynomial ([11])). By item 2 of fact 2, $X(M) = \{v_i, v_j : v_iv_j \in M\}$ is a vertex-cover of size $2|M|$ for G. Denote by m the size of M. Finally, set $p = |P| = |V \setminus X(M)|$. The following lemma will be used in theorem 1, just below, and later.

Lemma 1. *For any graph without isolated vertices, $p \leqslant m(\Delta - 1)$. If, in addition, the graph contains ι isolated vertices, then $p - \iota \leqslant m(\Delta - 1)$.*

Proof. Fix an edge $v_iv_j \in M$ such that at least one of v_i, v_j has neighbors in P. Let $P_i = P \cap \Gamma(v_i) = \{p_{i_1}, p_{i_2}, \ldots, p_{i_{|P_i|}}\}$; P_j is defined similarly. Suppose $|P_i| \geqslant |P_j|$. Then, if $P_j \neq \emptyset$, the following holds:

$$\begin{cases} P_j \subset P_i \\ |P_j| = 1 \end{cases} \qquad (1)$$

In fact, suppose $P_j \not\subseteq P_i$. Then, there exists at least one vertex $p_{j_k} \in P_j \setminus P_i$ and at least one vertex $p_{i_l} \in P_i \setminus P_j$. The path $p_{i_l}, v_i, v_j, p_{j_k}$ is augmenting with respect to M, contradicting so the fact that M is maximum.

On the other hand, suppose $P_j \subset P_i$ but $|P_j| > 1$. Then, with the same arguments, for any two vertices p_{i_l} and p_{j_k} in $P_j \cap P_i$, the path $p_{i_l}, v_i, v_j, p_{j_k}$ is augmenting with respect to M.

Consider now the following graph $B(N, E_B)$ constructed as follows:

- for every edge of M we draw a vertex (let N_M be the set of vertices so drawn); we also consider vertices of P as vertices of N; in other words $N = N_M \cup P$;
- let n_{ij} be the vertex of N_M associated with $v_i v_j \in M$; if there exists an edge linking either v_i or v_j with a vertex $p_k \in P$, then $n_{ij} p_k \in E_B$.

By expression (1) and the discussion above, any $n_{ij} \in N_M$ is linked with at most $\Delta - 1$ vertices of P (the maximum degree of G is Δ and one edge – the matching one – links v_i with v_j). This remains true if $P_j = \emptyset$. Consequently, $p \leqslant m(\Delta - 1)$.

On the other hand, if G contains a set I of ι isolated vertices, then the argument developed above remains valid on the graph $G'(V \setminus I, E)$, q.e.d. ∎

Since $n = 2m + p$ and, by lemma 1, $p \leqslant m(\Delta - 1)$ (resp., $p - \iota \leqslant m(\Delta - 1)$), one easily gets $n \leqslant m(\Delta + 1)$ (resp., $n - \iota \leqslant m(\Delta + 1)$) and reaches the following lemma.

Lemma 2. *In any graph with no isolated vertices, $m \geqslant n/(\Delta + 1)$. If the graph has ι isolated vertices, then $m \geqslant (n - \iota)/(\Delta + 1)$.*

Theorem 1. *For any graph G, the ratio of the size of any minimal vertex-cover to the size of the vertex-cover induced by* **`MAX_MATCHING(G)`** *is bounded above by $\Delta/2$.*

Proof. Assume first that G has no isolated vertices, denote by C a minimal vertex cover of G and by s the size of the independent set associated with C, i.e., $|V \setminus C| = s$. Denote by m the size of a maximum matching M of G. By item 2 of fact 2, algorithm `MAX_MATCHING` induces a vertex-cover of size $2m$. Recall finally that, by item 1 of fact 2, $P = V \setminus X(M)$ is independent.

Since C is supposed minimal, the independent set $V \setminus C$ is maximal. Consequently, using $s \geqslant n/(\Delta + 1)$ ([3]), we get

$$|C| = n - s \leqslant (\Delta + 1)s - s = \Delta s \tag{2}$$

We now distinguish the following two cases depending on the values of s and m: (i) $s \leqslant m$, (ii) $s > m$.

For case $\boldsymbol{s \leqslant m}$, using expression (2), we get:

$$\frac{|C|}{2m} \leqslant \frac{\Delta s}{2m} \leqslant \frac{\Delta}{2} \tag{3}$$

For case $\boldsymbol{s \geqslant m}$, using lemma 1, we get:

$$\frac{|C|}{2m} = \frac{n - s}{2m} \leqslant \frac{2m + p - m}{2m} \leqslant \frac{m + m(\Delta - 1)}{2} = \frac{\Delta}{2} \tag{4}$$

Combining expressions (3) and (4), one proves the theorem.

Consider now that G contains a set I of isolated vertices. Then, C being minimal, it does not contain any isolated vertex. Furthermore, `MAX_MATCHING`$(G[V \setminus I])$ = `MAX_MATCHING`(G). Hence the analysis performed just above remains valid and concludes the proof of the theorem. ∎

It is well-known ([3]) that, for any graph G and for any maximal matching M (of cardinality m) of G,

$$\tau(G) \geqslant m \tag{5}$$

Consequently, by theorem 1, the following corollary holds immediately.

Corollary 1. *For any graph, the ratio of the size of any minimal vertex-cover to the size of the optimal one is bounded above by Δ.*

3.2 An on-line algorithm for the case $t = n$

We are ready now to specify and analyze the following on-line algorithm for LVC.

```
BEGIN *OLVC*
        C ← ∅;
        t ← 1;
        G_t ← ({v_t}, ∅);
        WHILE t < n DO
                v ← v_{t+1};
                t ← t + 1;
                E_t ← E_t ∪ the set of edges linking v to vertices of V_t;
                V_t ← V_t ∪ {v_{t+1}};
                G_t ← (V_t, E_t);
                IF ∃u ≠ v : uv ∈ E_t ∧ u ∉ C THEN C ← C ∪ {v} FI;
        OD
        OUTPUT C;
END.    *OLVC*
```

Proposition 1. *The competitivity ratio of algorithm* **OLVC** *against an optimal off-line algorithm for VC is bounded above by Δ, while its competitivity ratio against* **MAX_MATCHING** *(assuming that* **MAX_MATCHING** *returns both a maximum matching M and $X(M)$) is bounded above by $\Delta/2$. Both bounds are tight.*

Proof. Following algorithm `OLVC`, $\forall v \in C$, $\exists uv \in E$ such that $u \notin C$. Hence, the vertex-cover C computed is minimal. Then, applications of corollary 1 and of theorem 1, respectively, conclude the ratios claimed.

Fix now a $\Delta \in \mathbb{N}$, consider a star $S_{\Delta+1}$ on $\Delta + 1$ vertices. Obviously, $\tau(S_{\Delta+1}) = 1$. Suppose that its center is the first vertex revealed; the rest of vertices can be revealed in any order. Then, algorithm `OLVC` will not include the star-center in C, while it will include all the remaining vertices of $S_{\Delta+1}$. Therefore, the competitivity ratio achieved in this case is equal to Δ. Furthermore, since $\tau(S_{\Delta+1}) = m = 1$, the competitivity ratio $\Delta/2$ against `MAX_MATCHING` is also achieved. ∎

3.3 Lower bounds on the competitivity of any algorithm for the case $t = n$

Suppose that vertices are numbered in the order they arrive; in step i, vertex v_i is revealed. Also consider that, in step i, $\{v_1, \ldots, v_i\} = C_i \cup S_i$, where C_i draws the vertex-set included in the vertex cover under construction and $S_i = \{v_1, \ldots, v_i\} \setminus C_i$. The final graph is denoted, as usually, by $G(V, E)$ and its maximum degree by Δ. The purpose of this section is to provide limits in the competitivity (against an optimal off-line algorithm) of any on-line algorithm solving LVC with $t = n$ (over all the ways the input-graph is revealed). Let us consider the solution of LVC as a two-players game, where the first one (player 1) reveals the instance and the second one (player 2) constructs the solution. Then, we prove the following theorem.

Theorem 2.

1. *No algorithm can achieve competitivity ratio strictly better than Δ, even if an isomorphic of G is known in advance.*

2. *No algorithm can achieve competitivity ratio strictly better than $\Delta - 2$, even if G is a tree and n is known in advance.*

Proof of item 1. The isomorphic of G revealed in advance consists of a disjoint collection of p stars, each of order $\Delta + 1$ and of $\Delta - 1$ isolated vertices, where Δ and p are fixed integers. Obviously, the degree of G is Δ and its order $n = p(\Delta+1)+\Delta-1$. Assume that player 1 reveals the graph with respect to the following rules:

i if C_i contains Δ isolated vertices (for the graph already revealed), then v_{i+1} is linked to all these vertices;

ii if $v_i \in S_i$ (in other words, v_i has not been taken in the solution) and v_i is not linked to any vertex v_j, $j < i$, and if $i \leqslant n - \Delta$, then vertices $v_i, v_{i+1}, \ldots, v_{i+\Delta}$ form a star rooted in v_i;

iii if p stars have been revealed, the rest of the vertices revealed are isolated;

iv if rules **i** and **ii** cannot be applied and $i \leqslant n - 1$, then vertex v_{i+1} is isolated with respect to the graph already revealed.

Applications of rules **i** to **iv** above implies that player 2 cannot do better than covering edges of any star by its leaves, while optimal off-line solution consists of the star-centers. Therefore a ratio of Δ is achieved at best and this completes the proof of item 1 of the theorem.

Proof of item 2. Let Δ be an integer greater than, or equal to, 3 and set $n = \Delta(\Delta + 1) + 1$. Consider that player 1 reveals the graph following the rules below:

(i) if C_i contains Δ isolated vertices (with respect to the graph already revealed) and $i \leqslant n-2$, then v_{i+1} is linked to all these isolated vertices;

(ii) if $v_i \in S_i$ (in other words, v_i has not been taken in the solution) and v_i is not linked to any vertex v_j, $j < i$, and if $i \leqslant n - \Delta - 1$, then vertices $v_i, v_{i+1}, \ldots, v_{i+\Delta}$ form a star rooted in v_i;

(iii) consider $v_i \in S_i$, v_i isolated with respect to the graph already revealed, and $n - 2 \geqslant i \geqslant n - \Delta$; set $A = \{v_j : j < i, v_j \in C_i, \forall k \leqslant i, v_j v_k \notin E\}$ (i.e., A is the set of the isolated vertices, at instant t, taken in C_i) and $B = \{v_{i+2}, \ldots, v_{n-1}\}$; then:

   (a) v_{i+1} is linked to v_i and to any element of set A;

   (b) the elements of B form an independent set and are linked to v_i

(iv) if rules **(i)** and **(ii)** do not apply and if $i \leqslant n - \Delta$, then vertex v_{i+1} is isolated with the graph already revealed;

(v) v_n is linked to Δ vertices of degree 1 picked in the several connected components of the graph revealed until step $n - 1$.

In figure 1 a graph G fitting rules **(i)** to **(v)** is shown for $n = 31$ and $\Delta = 5$. Vertices are numbered in the order they have been revealed. Here, rule **(iii)** is applied for vertex 27. Then, $A = \{19, 20\}$ and $B = \{29, 30\}$. The circle vertices represent the LVC-solution, while the square ones represent the independent set associated with it. A graph fitting item 1 of the theorem could be as the one of figure 1 induced by the set of vertices $\{1, \ldots, 27\}$ plus one isolated square vertex.

If rule **(iii)** is not applied, then in step $n-1$, the graph contains Δ stars, their vertices of degree 1 making part of the solution constructed by player 2. In this case, $\tau(G) = \Delta + 1$ (the roots of

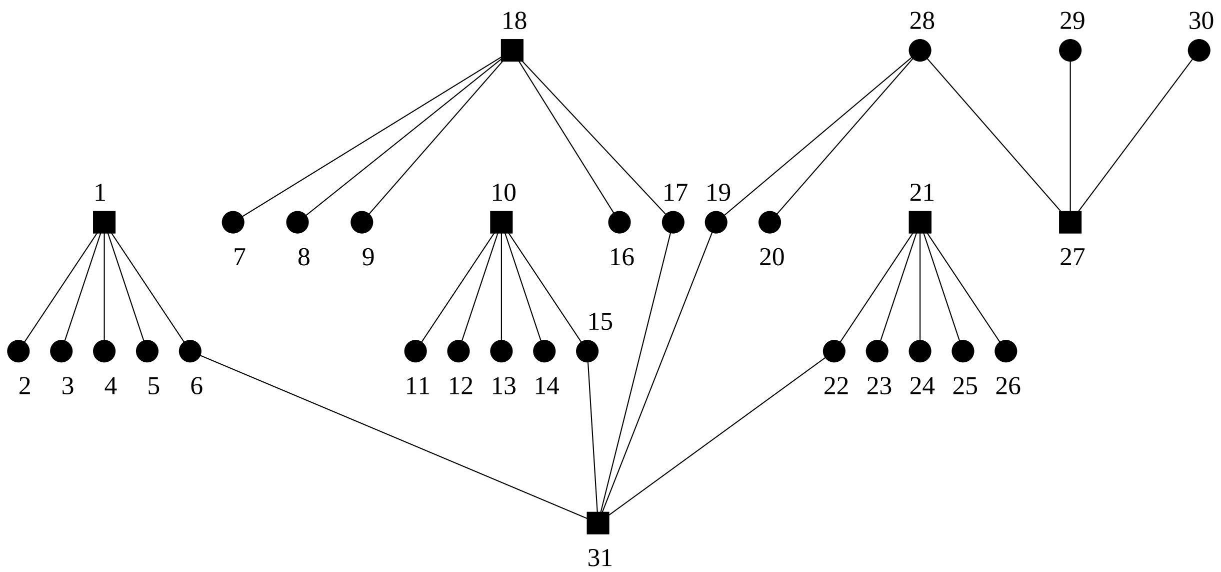

Figure 1: A graph fitting rules **(i)** to **(v)** with $n = 31$ and $\Delta = 5$.

the stars plus vertex v_n), while the solution constructed is of size $\Delta(\Delta + 1)$. The competitivity ratio is in this case Δ.

Suppose now that rule **(iii)** is applied (recall that this is the case for vertex 27 in figure 1). Then in step i, the graph consists of k stars (their leaves making part of the solution constructed by player 2) plus the vertices of $A \cup \{i\}$. In this case, the total number of vertices verifies $n = \Delta(\Delta + 1) + 1 = k(\Delta + 1) + |A| + |B| + 3$, with $|A| < \Delta$ (if not, rule **(i)** would be applied one more time) and $|B| \leqslant \Delta - 2$ (because $i \geqslant n - \Delta$). We deduce $k = \Delta - 1$. In this case, $\tau(G) = \Delta - 1 + 3 = \Delta + 2$, while the solution finally constructed by player 2 has at least $(\Delta - 1)\Delta + \Delta = \Delta^2$ vertices (the leaves of the $\Delta - 1$ stars plus set A plus set B plus x_{i+1}). The competitivity ratio implied is then at least $\Delta - 2$. In figure 1, $\tau(G) = 7$ (the optimal solution is the set $\{1, 10, 18, 21, 27, 28, 31\}$, while the solution constructed by player 2 (the set of the circle vertices of figure 1) is of cardinality 25.

In order to conclude, let us note that the graph at step $n - 1$ consists of Δ acyclic connected components, each of them containing at least one vertex of degree 1. This makes that rule **(v)** is feasible and guarantees that the final graph is a tree. So, the proofs of item 2 and of the theorem are complete. ∎

4 On-line vertex covering with $n > t \geqslant 2$

We assume in this section that G is revealed by non-empty clusters G_i, $i = 1, \ldots, t$, with $2 \leqslant t < n$. We first study the case $t > 2$. The case $t = 2$, being interesting by itself, is examined separately in section 4.2. We suppose that t is known at the beginning of the game.

4.1 On-line vertex covering with $n > t > 2$

For the case we deal with in this section, we propose the following algorithm.

```
BEGIN *t_OLVC*
*arrival of G_1*
(1)    C ← X(MAX_MATCHING(G_1));
*arrival of G_i, i = 2..., t*
```

```
(2)    FOR i ← 2 TO t DO
(3)        C ← C ∪ X(MAX_MATCHING(G_i));
(4)        FOR u ∈ V_i \ X(MAX_MATCHING(G_i)) DO
(5)            IF ∃v ∈ (∪_{1⩽j⩽i−1} V_j) \ C : uv ∈ E THEN C ← C ∪ {u} FI
(6)        OD
(7)    OD
(8)    OUTPUT C
END    *t_OLVC*
```

Obviously, the set C finally computed by t_OLVC is a vertex-cover, although not necessarily minimal. So, proposition 1 does not represent the worst case for its competitivity ratio. Note that for the case where clusters are assumed without any restriction, setting $|C| \leqslant n - |I|$ (where I denotes the set of isolated vertices, if any) and using lemma 2, competitivity ratio $\Delta + 1$ is immediately deduced.

Theorem 3. *Let λ_i be the number of the isolated vertices of G_i introduced in C and set $\lambda = \sum_{i=1}^{t} \lambda_i$ and $\rho = \lambda/|A|$. Then, the competitivity ratio of algorithm t_OLVC against an optimal VC algorithm is bounded above by $2 + (\Delta - 2)/(2 - \rho)$.*

Proof. Denote by A_i, $i = 2, \ldots, t$, the vertex-sets introduced in C during the execution of lines (4) to (6) of algorithm t_OLVC and set $A = \cup_{i=2}^{t} A_i$. Observe that vertex-set A is exposed with respect to the (non-maximum) matching $\cup_{i=1}^{t} M_i$; moreover, it does not contain any isolated vertex. Observe also that any isolate vertex is exposed with respect to any matching of G; hence

$$\lambda \leqslant |A| \Longleftrightarrow \rho \leqslant 1 \tag{6}$$

Denote by M_i, $i = 1, \ldots, t$, a maximum matching of G_i, set $m_i = |M_i|$, $i = 2, \ldots, t$. Then, the cardinality of the on-line solution C computed by t_OLVC can be written as follows:

$$|C| = 2 \sum_{i=1}^{t} m_i + |A| \tag{7}$$

Let E' be the set of edges that have entailed introduction of the vertices of A in C and denote by $B(A \cup (V_1 \setminus X(M_1)), E')$ the partial subgraph of G induced by $A \cup (V_1 \setminus X(M_1))$ and by E'. Also, denote by M_B a maximum matching of B and by m_B the cardinality of M_B. Remark also that $M_B \cup_{i=1}^{t} M_i$ is a matching of G, not nessecarily maximum, and that the graph B is bipartite with color-classes A and $V_1 \setminus X(M_1)$. Expression (5) can, for the purposes of our proof, be rewritten as

$$\tau(G) \geqslant \left(\sum_{i=1}^{t} m_i \right) + m_B \tag{8}$$

and, using expressions (7) and (8), the competitivity ratio of t_OLVC becomes

$$c_{\texttt{t\_OLVC}} = \frac{|C|}{\tau(G)} \leqslant \frac{|C|}{\left(\sum_{i=1}^{t} m_i \right) + m_B} = \frac{2 \sum_{i=1}^{t} m_i + |A|}{\left(\sum_{i=1}^{t} m_i \right) + m_B} = 2 + \frac{|A| - 2m_B}{\left(\sum_{i=1}^{t} m_i \right) + m_B} \tag{9}$$

Finally, using lemma 2, we get

$$m_B \geqslant \frac{|A \cup (V_1 \setminus X(M_1))|}{\Delta} \geqslant \frac{|A|}{\Delta} \tag{10}$$

On the other hand, let $i \in \{1, \ldots, t\}$ and let I be the set of the isolated vertices of cluster G_i. We then have $|I| = \lambda_i + (|I| - |A_i \cap I|)$, or $|A_i| - \lambda_i = |A_i| + (|I| - |A_i \cap I|) - |I| \leqslant p_i - |I| \leqslant m_i(\Delta - 1)$, where the first inequality holds because $A_i \subseteq P_i$ and, as we have already mentioned in the beginning of the proof, the set $I \setminus (A_i \cap I)$, being isolated in G_i, it is exposed with respect to any matching of G_i; the second inequality holds thanks to lemma 1. Summing inequalities $|A_i| - \lambda_i \leqslant m_i(\Delta - 1)$ for $i = 1, \ldots, t$, we obtain

$$\sum_{i=1}^{t} m_i \geqslant \frac{|A| - \lambda}{\Delta - 1} \tag{11}$$

Using expressions (6), (10) and (11), expression (9) becomes

$$\begin{aligned} c_{\texttt{t\_OLVC}} &\leqslant 2 + \frac{|A| - 2m_B}{\left(\sum_{i=1}^{t} m_i\right) + m_B} \leqslant 2 + \frac{|A| - \frac{2|A|}{\Delta}}{\frac{|A|-\lambda}{\Delta-1} + \frac{|A|}{\Delta}} \\ &\leqslant 2 + \frac{\frac{\Delta-2}{\Delta}}{\frac{\Delta(1-\rho)+\Delta-1}{\Delta(\Delta-1)}} = 2 + \frac{(\Delta-2)(\Delta-1)}{\Delta(2-\rho)-1} \leqslant 2 + \frac{\Delta-2}{2-\rho} \end{aligned} \tag{12}$$

and the proof of the theorem is complete. ∎

Recall that $\rho \leqslant 1$ (expression (6)). Furthermore, expression (12) is increasing with ρ; hence, setting $\rho = 1$ the following result is immediately obtained.

Corollary 2. *The competitivity ratio of algorithm* `t_OLVC` *against an optimal VC algorithm is bounded above by* Δ.

On the other hand, consider that clusters arrive without isolated vertices. In this case, for any $i = 1, \ldots, t$, $\lambda_i = 0$, so, $\rho = 0$ and, using expression 12, the following holds.

Corollary 3. *Whenever clusters arrive without isolated vertices, the competitivity ratio of algorithm* `t_OLVC` *against an optimal VC algorithm is bounded above by* $(\Delta + 2)/2$.

Note that the solution C computed by algorithm `t_OLVC` is not necessarily minimal. Consequently, the result of corollary 2 cannot be derived by direct application of corollary 1.

The bound of corollary 3 can be slightly improved by the following way. Since, for $i = 1, \ldots, t$, the vertices of A_i are exposed with respect to M_i, using lemma 1 ($A_i \cap I = \emptyset$), we get

$$|A_i| \leqslant m_i(\Delta - 1) \tag{13}$$

On the other hand, since no cluster contains isolated vertices, the bipartite graph $B(A \cup (V_1 \setminus X(M_1)), E')$ considered in the proof of theorem 3 has maximum degree bounded above by $\Delta - 1$ (at least one edge per vertex in $A \cup (V_1 \setminus X(M_1))$, links it to vertices of $X(M_i)$, $i = 1, \ldots, t$). Consequently,

$$|A| \leqslant (\Delta - 1)\,|V_1 \setminus X(M_1)| \leqslant (\Delta - 1)m_B \tag{14}$$

where the last inequality holds because, in a bipartite graph, the cardinality of a maximum matching is smaller than the cardinality of the smallest among the color-classes. So, combining expressions (13) and (14), expression (9) becomes:

$$c_{\texttt{t\_OLVC}} \leqslant 2 + \frac{|A| - 2m_B}{\left(\sum_{i=1}^{t} m_i\right) + m_B} \leqslant 2 + \frac{|A| - 2\frac{|A|}{\Delta-1}}{\frac{2|A|}{\Delta-1}} = 2 + \frac{\Delta-3}{2} = \frac{\Delta+1}{2} \tag{15}$$

and expression (15) leads immediately to the following final corollary.

Corollary 4. *Whenever clusters arrive without isolated vertices, the competitivity ratio of algorithm* `t_OLVC` *against an optimal VC algorithm is bounded above by* $(\Delta + 1)/2$.

4.2 On-line vertex-covering with $t = 2$

Suppose now that the input graph is revealed within two clusters $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$. Assume also that n, the order of the final graph, is known at the beginning of the game. We recall that, following our assumptions, one has to decide which vertices of the first cluster will belong to the final solution before the arrival of the second cluster.

4.2.1 G has no isolated vertices

In this section we suppose that no additional hypotheses are admitted on the forms of the clusters and analyze the competitivity ratio of the following algorithm.

```
BEGIN *2_OLVC*
*arrival of G₁*
(1)   CASE V₁ ⩽ n/2 DO C ← V₁;
(2)                    GOTO (6);
(3)        V₁ > n/2 DO C ← X(MAX_MATCHING(G₁));
(4)                    GOTO (7);
(5)   OD
*arrival of G₂*
(6)   OUTPUT C ← C ∪ X(MAX_MATCHING(G₂));
(7)   OUTPUT C ← C ∪ V₂;
END.  *2_OLVC*
```

Theorem 4. *If G has no isolated vertices, then the competitivity ratio of algorithm* `2_OLVC` *against an optimal VC algorithm verifies $c_{2\_\texttt{OLVC}} \leqslant (\Delta + 5)/2$. This ratio is tight.*

Proof. Denote, for $i = 1, 2$, by M_i the matchings computed by algorithm `MAX_MATCHING` on G_i at lines `(3)` and `(6)` and by m_i their sizes.

Suppose lines `(1)` and `(2)` of algorithm `2_OLVC` are executed. Then, the solution returned is $C = V_1 \cup X(M_2)$ with $|C| \leqslant n/2 + 2m_2$. Combining expression for C with expression (5) and taking into account lemma 2 and the fact that $m_2 \leqslant m$, the following holds:

$$c_{2\_\texttt{OLVC}} = \frac{|C|}{\tau(G)} \leqslant \frac{\frac{n}{2} + 2m_2}{m} \leqslant \frac{n}{2m} + 2 \leqslant \frac{\Delta + 1}{2} + 2 = \frac{\Delta + 5}{2} \tag{16}$$

Suppose now that lines `(3)` and `(4)` of algorithm `2_OLVC` are executed instead. Then, $|V_2| \leqslant n/2$ and the solution returned is the one of line `(7)`, i.e., $C = X(M_1) \cup V_2$. In this case also the arguments previously developed hold. Hence, expression (16) always gives the competitivity ratio achieved.

Let us now show that the analysis above is asymptotically tight. Consider a graph $G(V, E)$ collection of R stars, each of maximum degree Δ. Consider the subgraph G_1 of G consisting of a set of $n/2$ exposed vertices with respect to a maximum matching M of G. Remark that M contains one edge per star and that $V(G_1)$ is a set of isolated vertices of size not larger than $n/2$. Set $G_2 = G[V \setminus V(G_1)]$ and assume that G is revealed per clusters G_1 and G_2. Then,

$$|C| = \frac{n}{2} + 2R = \frac{n}{2} + \frac{2n}{\Delta + 1} \tag{17}$$

$$\tau(G) = \frac{n}{\Delta + 1} = R \tag{18}$$

$$\frac{|C|}{\tau(G)} = \frac{\Delta + 5}{2} \tag{19}$$

This completes the proof of the theorem. ∎

4.2.2 Clusters have no isolated vertices

Suppose now that clusters G_i arrive with no isolated vertices. Let n_i be the order of G_i. Denote by M_i, $i = 1, 2$, a maximum matching of G_i, by P_i the set of the exposed vertices with respect of M_i, by p_i its cardinality and consider the following on-line algorithm.

```
BEGIN *C2_OLVC(n, ϵ)*
(1)   fix an ϵ > 1;
(2)   C ← ∅;
*arrival of G₁*
(3)   M₁ ← MAX_MATCHING(G₁);
(4)   IF n₁ ⩽ n/ϵ THEN C ← C ∪ V₁;
(5)               ELSE C ← C ∪ X(M₁);
*arrival of G₂*
(6)   M₂ ← MAX_MATCHING(G₂);
(7)   C ← C ∪ X(M₂);
(8)   A₂ ← {v ∈ V₂ \ X(M₂) : ∃u ∈ V₁ \ C, uv ∈ E};
(9)   OUTPUT C ← C ∪ A₂;
END.  *C2_OLVC(n, ϵ)*
```

Theorem 5. *Under the hypothesis that clusters arrive with no isolated vertices, there exists ϵ_0, the largest among the roots of the polynomial $\epsilon^2 - 3\epsilon + 1$, such that the competitivity ratio of algorithm `C2_OLVC(n,`ϵ_0`)` against an optimal VC algorithm is bounded above by $2 + ((\Delta + 1)/\epsilon_0)$.*

Proof. Set, for $i = 1, 2$, $m_i = |M_i|$ and note the following fact that can be immediately deduced from algorithm `C2_OLVC`.

Fact 3. Whenever line `(4)` is executed, then line `(8)` computes $A_2 = \emptyset$; therefore the final covering C computed in line `(9)` satisfies $C = V_1 \cup X(M_2)$. ▮

Let us first suppose that line `(4)` is executed. Then, using fact 3, expression (5) and lemma 2, we get:

$$\frac{|C|}{\tau(G)} = \frac{n_1 + 2m_2}{m} \leqslant \frac{\frac{n}{\epsilon} + 2m_2}{m} \leqslant \frac{\frac{n}{\epsilon}}{m} + 2 \leqslant \frac{\frac{n}{\epsilon}}{\frac{n}{\Delta+1}} + 2 = \frac{\Delta + 1}{\epsilon} + 2 \tag{20}$$

Let us now suppose that line `(5)` is executed instead. Then, the set C finally computed in line `(9)` by algorithm `C2_OLVC` verifies:

$$|C| = |X(M_1)| + |X(M_2)| + |A_2| = 2(m_1 + m_2) + |A_2| \tag{21}$$

Denote by $Q_1 \subseteq V_1 \setminus X(M_1)$ the set of vertices of V_1 that has entailed the introduction of set A_2 in C, and by $B(Q_1, A_2, E_B)$, the subgraph of G induced by $Q_1 \cup A_2$. Since they are both independent (subsets of P_1 and P_2, respectively), B is bipartite. Denote also by M_B a maximum matching of B and set $m_B = |M_B|$. Since $M_1 \cup M_2 \cup M_B$ is a matching for G,

$$\tau(G) \geqslant m_1 + m_2 + m_B \tag{22}$$

Consider set $X(M_B) \cap Q_1$; obviously, $|X(M_B) \cap Q_1| = m_B$. Since G_1 is supposed without isolated vertices, any vertex of $X(M_B) \cap Q_1$ has at most $\Delta - 1$ neighbors in A_2. On the other hand, M_B being maximum for B, any vertex of A_2 receives edges from at least one vertex of $X(M_B) \cap Q_1$. So,

$$|A_2| \leqslant m_B(\Delta - 1) \tag{23}$$

Also, since G_1 and G_2 are both assumed without isolated vertices, application of lemma 2 gives:

$$\begin{array}{rcl} m_1 & \geqslant & \frac{n_1}{\Delta+1} \\ m_2 & \geqslant & \frac{n_2}{\Delta+1} \end{array} \tag{24}$$

Combining expressions (21), (22), (23) and (24), performing some little and easy algebra and taking into account $n_1 + n_2 = n$, one gets:

$$\frac{|C|}{\tau(G)} \leqslant \frac{2(m_1+m_2)+|A_2|}{m_1+m_2+m_B} = 2 + \frac{|A_2|-2m_B}{m_1+m_2+m_B} \leqslant 2 + \frac{\frac{\Delta-3}{\Delta-1}|A_2|}{\frac{n}{\Delta+1}+\frac{|A_2|}{\Delta-1}} \tag{25}$$

Recall that we are currently considering case $n_1 \geqslant n/\epsilon$, i.e.,

$$n_2 \leqslant n - \frac{n}{\epsilon} = n\frac{\epsilon-1}{\epsilon} \tag{26}$$

Using expression (24) for m_2, denoting by p_2 the number of the exposed vertices of V_2 with respect to M_2, and using expression (26), we obtain

$$|A_2| \leqslant p_2 = n_2 - 2m_2 \leqslant n_2 - \frac{2n_2}{\Delta+1} = \frac{\Delta-1}{\Delta+1}n_2 \leqslant \frac{(\Delta-1)(\epsilon-1)}{(\Delta+1)\epsilon}n \tag{27}$$

Remark also that expression (25) is increasing with $|A_2|$. So, combining expressions (25) and (27):

$$\frac{|C|}{\tau(G)} \leqslant 2 + \frac{\frac{(\Delta-3)(\epsilon-1)}{(\Delta+1)\epsilon}n}{\frac{n}{\Delta+1}+\frac{(\epsilon-1)n}{\epsilon(\Delta+1)}} = 2 + \frac{(\Delta-3)(\epsilon-1)}{2\epsilon-1} \tag{28}$$

Note that, for a fixed ϵ, $(\Delta-3)(\epsilon-1)/(2\epsilon-1) \leqslant (\Delta+1)(\epsilon-1)/(2\epsilon-1)$ and that expression (20) is decreasing with ϵ, while expression (28) is increasing. These two expressions asymptotically coincide when

$$\frac{\Delta+1}{\epsilon} + 2 = 2 + \frac{(\Delta+1)(\epsilon-1)}{2\epsilon-1} \Longleftrightarrow \epsilon^2 - 3\epsilon + 1 = 0 \overset{\epsilon>1}{\Longrightarrow} \epsilon_0 \simeq 2.62 \tag{29}$$

Setting $\epsilon_0 = 2.62$, we get $c_{\texttt{C2\_OLVC}} \leqslant (\Delta+6.24)/2.62$. This, for large values of Δ, is asymptotically equal to $\Delta/2.62$. ∎

4.3 Lower bounds for the competitivity ratio

4.3.1 Case $t > 2$

We provide in this section a competitivity upper bound, given in theorem 6, for the case where the number of clusters needed for the revealing of the whole graph is $O(\log n\sqrt{n})$. The proof of the theorem being quite technical, it is given in appendix.

Theorem 6. *When $t = O(\log n\sqrt{n})$, and any cluster is non-empty, no on-line algorithm for LVC can achieve competitivity ratio smaller than $\Delta - 2$ against an optimal off-line algorithm, even if the input-graph is a tree and n is known in advance.*

4.3.2 Limits on the competitivity for $t = 2$

As previously in section 3.3, we bring to the fore a graph and a strategy for revealing it in two steps such that every on-line VC-algorithm cannot achieve competitivity ratio better than the bound provided.

Theorem 7. *For $t = 2$ and for all $\Delta \geqslant 2$, no algorithm can achieve competitivity ratio strictly better than $(\Delta + 1)/2$ for a graph of maximum degree Δ, even if it is bipartite with no isolated vertices, both clusters have the same size and an isomorphic of the input-graph is known in advance.*

Proof. Given an integer k and two sets $A = \{1, \dots |A|\}$ and $B = \{1, \dots |B|\}$ such that $|B| = k|A|$, we set $A \times_k B = \{(a_i b_{(i-1)k+j}) \in A \times B, i \in \{1, \dots, |A|\}, j \in \{1, \dots, k\}\}$. In other words, if A and B are vertex-sets, the graph $(A \cup B, A \times_k B)$ consists of $|A|$ stars of size $k+1$ rooted in the vertices of A.

Let $\Delta \geqslant 2$ be a fixed integer and set

$$n = 2\Delta(\Delta + 1) \tag{30}$$

We define $H = (V_1 \cup V_2, E)$ where:

- $V_1 = N_1^1 \cup N_1^2$ (sets N_1^1 and N_1^2 are mutually disjoint) with $|N_1^1| = \Delta$, $|N_1^2| = \Delta^2$, $N_1^1 \cap N_1^2 = \emptyset$;
- $V_2 = N_2^1 \cup N_2^2$ (sets N_2^1 and N_2^2 are mutually disjoint) with $|N_2^1| = \Delta^2$, $|N_2^2| = \Delta$, $N_2^1 \cap N_2^2 = \emptyset$;
- $E = (N_1^1 \times_\Delta N_2^1) \cup (N_2^2 \times_\Delta N_1^2)$.

The so-constructed graph H is bipartite, without isolated vertices and a minimum cardinality vertex covering of H is of size $\tau(H) = n/(\Delta + 1)$.

Let us consider an on-line algorithm `OLVC`. We will show that there exists a bipartite graph $G(V_1, V_2, E)$ isomorphic to H for which the competitivity ratio of `OLVC` is at least $(\Delta+1)/2$ when V_1 is revealed in step 1 and V_2 is revealed in step 2. More precisely, the solution computed by `OLVC` in G contains at least $n/2$ vertices, where $n = |V_1| + |V_2|$. Assume $|V_1| = n/2$, let N_1 be the set of vertices of V_1 introduced in the solution by `OLVC` and set $n_1 = |N_1|$. Define integers p, r, k such that

$$\begin{aligned} n_1 &= p\Delta + r && (31)\\ r &\leqslant \Delta - 1 && (32)\\ k &= (\Delta + 1) - p && (33) \end{aligned}$$

Note that $k \geq 0$ and that, using expressions (30) and (33), one gets the following expression (34); expression (35) is immediate by the discussion above:

$$\begin{aligned} (k + p)\Delta &= \frac{n}{2} && (34)\\ k &\leqslant \frac{n}{2} - n_1 && (35) \end{aligned}$$

Indeed, expression (35) obviously holds if $k = 0$ ($n_1 = n/2$), or $k = 1$ ($n_1 < n/2$). Suppose now $k \geqslant 2 \geqslant \Delta/(\Delta - 1)$. We then have, using expressions (31) to (33):

$$\frac{n}{2} - n_1 = \Delta(\Delta + 1) - p\Delta - r > \Delta(\Delta - p) = \Delta(k - 1) \geqslant k \tag{36}$$

Set V_1 as the partition of sets N_1^p, N_1^k and R_1, where $N_1^p \subset N_1$, $N_1^k \cap N_1 = \emptyset$ and

$$\begin{cases} |N_1^p| = p\Delta \\ |N_1^k| = k \end{cases} \tag{37}$$

Such decomposition is possible because of expression (35). Then, $|R_1| = k(\Delta - 1)$ and we have:

$$V_1 = N_1^p \cup N_1^k \cup R_1 \tag{38}$$

Set also V_2 as the partition of sets N_2^p, N_2^k and R_2 where $|N_2^p| = p$, and $|N_2^k| = \Delta k$. Then $|R_2| = p(\Delta - 1)$. In all,

$$V_2 = N_2^p \cup N_2^k \cup R_2 \tag{39}$$

We now consider three cases, namely, (i) $k > 0$ and $p > 0$, (ii) $k = 0$ and (iii) $p = 0$.

For case (i) ($k > 0$ and $p > 0$), using expression (34), $p\Delta < \Delta(\Delta + 1)$ and, consequently, $p \leq \Delta$. Then, we set $a = p - 1$ and $b = \Delta - p$; a and b are positive integers satisfying:

$$\begin{cases} k(\Delta - 1) & = & a + b\Delta \\ p(\Delta - 1) & = & a\Delta + b \end{cases} \tag{40}$$

Revisit expressions (38) and (39) and set $R_1 = R_1^a \cup R_1^b$, $R_2 = R_2^a \cup R_2^b$ with $|R_1^a| = a$, $|R_1^b| = b\Delta$, $|R_2^a| = a\Delta$, $|R_2^b| = b$ where a and b are as in expression (40); consider also that R_1^a, R_1^b, R_2^a, and R_2^b are mutually disjoint. Furthermore, the edge-set E of G is considered as follows: $E = N_2^p \times_\Delta N_1^p \cup N_1^k \times_\Delta N_2^k \cup R_1^a \times_\Delta R_2^a \cup R_2^b \times_\Delta R_1^b$. For G (which is isomorphic to H), the on-line solution computed by OLVC necessarily contains set $N_1 \cup N_2^k$ of cardinality at least $n/2$. On the other hand, $N_2^p \cup N_1^k \cup R_1^a \cup R_2^b$ is a minimum-size vertex cover.

For case (ii) ($k = 0$), expression (33) gives $p = \Delta + 1$ and expression (37) implies $n_1 = n/2$. Then E is defined so that G is isomorphic to H.

For case (iii) ($p = 0$), by expressions (32) and (36), $n/2 - n_1 > \Delta(\Delta + 1) - \Delta \geqslant \Delta$. So we can define N_1^1, N_1^2, N_2^1 and N_2^2 as in H (see the three items at the beginning of the proof) with the supplementary condition that $N_1^1 \cap N_1 = \emptyset$.

In all of the above cases, the on-line solution computed by OLVC contains at least $n/2$ vertices. On the other hand, the corresponding graphs G being isomorphic to H, $\tau(G) = n/(\Delta + 1)$. Consequently, the competitivity ratio of OLVC is at least $(\Delta + 1)/2$, and the proof of the theorem is complete. ∎

Note that if we allow isolated vertices in G, one can easily show that one cannot guarantee competitivity ratio strictly better than $1/\Delta$.

5 Allowing backtracking

In this section we somewhat change the working hypotheses adopted and suppose that one can go back over the solution constructed during previous steps. We assume that one can change this solution but she/he has to pay some cost for doing it.

Our on-line algorithm for the case of the backtracking is basically algorithm t_OLVC. The spirit of our thought process can be outlined as follows. The best approximation ratio known for VC is bounded above by 2 (this ratio is equal to $2 - (\log\log n / \log n)$ ([2])). On the other hand, LVC being computationally harder, it is a priori worse approximated than VC. So, one can "restrain" her/himself in searching for competitivity ratios as near as possible to 2. The maximum matching performed on each cluster of G by algorithm t_OLVC obviously guarantees approximation ratio 2 on any cluster. The fact that the whole competitivity ratio is finally "deteriorated" is due to the vertices of the graph B that have to be taken into account in order to cover cross-edges, i.e., edges between clusters. So the algorithm we propose in what follows starts with running MAX_MATCHING on each cluster and by delaying its decision on the cross-edges (in other words, the exposed vertices of any cluster are firstly considered as not belonging to the solution under construction). Next, once all clusters revealed, graph G_B is formed and MAX_MATCHING(B) is run. The final solution is the union of the endpoints of all the edges retained by the successive runs of MAX_MATCHING. In what follows, for $V' \subseteq V$, we denote by $G[V']$ the subgraph of G induced by V'.

```
BEGIN *Bt_OLVC*
        C ← ∅;
        FOR i ← 1 TO t DO C ← C ∪ X(MAX_MATCHING(G_i)) OD
        B ← G[∪_{i=1}^t (V_i \ X(MAX_MATCHING(G_i)))];
        OUTPUT C ← C ∪ X(MAX_MATCHING(B));
END     *Bt_OLVC*
```

Let us denote by κ the cost due to the change of the status of a non-covering vertex to a covering one. Also, as previously, we denote by M_i the edge-set computed by MAX_MATCHING(G_i) and by m_i its cardinality, $i = 1, \ldots, t$; furthermore, we denote by M_B the edge-set computed by MAX_MATCHING(B) and by m_B its cardinality.

Theorem 8. *The competitivity ratio of algorithm Bt_OLVC against an optimal off-line algorithm for VC is bounded above by* 2κ.

Proof. As one can see from algorithm Bt_OLVC, the vertices changed are the ones of the set $\cup_{i=1}^{t}(V_i \setminus X(M_i))$. Among these vertices, exactly $|X(M_B)| = 2m_B$ vertices pass from non-covering to covering ones. Suppose that for each of them a cost κ has to be paid. Then, the competitivity ratio of Bt_OLVC can be written (using expression (8)) as

$$c_{\mathtt{Bt\_OLVC}} = \frac{\left(2\sum_{i=1}^{t} m_i\right) + 2\kappa m_B}{\left(\sum_{i=1}^{t} m_i\right) + m_B} \leqslant 2\kappa \tag{41}$$

and the proof of the theorem is complete. ∎

Theorem 8 draws the general case where no further specification of the type of the cost is given. We now assume two cost-models:

- κ is a fixed constant;
- κ is at most equal to the total number of vertices changed.

For the first cost-model, using expression (41), the following result is directly proved.

Corollary 5. *If a fixed cost has to be paid for any vertex-status modification, then the competitivity ratio of algorithm Bt_OLVC against an optimal off-line VC-algorithm is constant.*

Let us now focus ourselves on the second of the cost-models specified above. For this case, we somewhat modify algorithm Bt_OLVC as shown just below, and assume n known in advance.

```
BEGIN *Mt_OLVC*
*arrival of G_1*
(1)    C ← X(MAX_MATCHING(G_1));
*arrival of G_i, i = 2..., t*
(2)    FOR i ← 2 TO t DO
(3)        M_i ← MAX_MATCHING(G_i);
(4)        P_i ← V_i \ X(M_i);
(5)        B ← G[∪_{j=1}^i P_j];
```

```
(6)         M_B ← MAX_MATCHING(B);
(7)         IF m_B ⩽ √n THEN C ← C ∪ X(M_i);
(8)                       ELSE C ← C ∪ V_i;
(9)         FI
(10)   OD
(11)   i_0  ← last i for which line (7) is executed;
(12)   B′ ← G[∪_{j=1}^{i_0} P_j];
(13)   OUTPUT C ← C ∪ X(M_B′);
END    *Mt_OLVC*
```

Theorem 9. *If for any vertex changed, the cost of the change equals the total number of vertices changed, then the competitivity ratio of algorithm* **Mt_OLVC** *against an optimal off-line VC-algorithm is bounded above by* $3\sqrt{n}$.

Proof. Remark first that the only vertex-changes performed by algorithm `Mt_OLVC` are on $X(M_{B'})$ (where B' is the graph constructed in line `(12)`) and, furthermore, that $m_{B'}$ always satisfies

$$m_{B'} \leqslant \sqrt{n} \tag{42}$$

If line `(8)` of algorithm `Mt_OLVC` is not executed at all, i.e., if $i_0 = t$ (line `(11)`), then $m_{B'} \leqslant \sqrt{n}$. Consequently, using expressions (41) and (42):

$$c_{\texttt{Mt\_OLVC}} \leqslant 2m_{B'} \leqslant 2\sqrt{n} \tag{43}$$

On the other hand, suppose that line `(8)` is executed at least once. Then,

$$m_B > \sqrt{n} \tag{44}$$

Using expression (42), setting $\kappa = m_{B'}$ and denoting by $v(C)$ the value of the set C (in $v(C)$ any vertex non changed counts 1 and any vertex changed counts κ), we get:

$$v(C) \leqslant n - 2m_{B'} + 2\kappa m_{B'} \leqslant n + 2m_{B'}^2 \tag{45}$$

Denote by m the cardinality of a maximum matching of G and use expressions (8), (44) and (45). Then, the following holds for the competitivity ratio of `Mt_OLVC`:

$$c_{\texttt{Mt\_OLVC}} = \frac{v(C)}{\tau(G)} \leqslant \frac{n + 2m_{B'}^2}{m} \leqslant \frac{n}{m_B} + 2m_{B'} \leqslant 3\sqrt{n} \tag{46}$$

Expressions (43) and (46) conclude then the proof of the theorem. ∎

We now show that the result of theorem 9 is quite tight, since no on-line algorithm can achieve competitivity ratio (against an optimal off-line one) better than $O(\sqrt{n})$.

Theorem 10. *Under the hypotheses of theorem 9, no on-line algorithm for VC can achieve, against an optimal off-line algorithm, competitivity ratio* $\sqrt{2n}$*, even if the input-graph is bipartite without isolated vertices and* n *is known in advance.*

Proof. Let $(\Delta, n_1) \in \mathbb{N} \times \mathbb{N}$ and set $n = (1 + \Delta)n_1$. At the first step, V_1 is an independent set of size n_1. If player 2 selects some vertices of V_1, then the whole instance is a graph without any edge. In this case, the optimal value $\tau^*(G)$ is 0, whereas the on-line value is positive. The resulting ratio equals ∞ and the theorem holds.

Consequently, we can focus ourselves on the case where player 2 selects no vertices during the first step. In this case, the instance graph consists of n_1 stars of size $(1+\Delta)$ rooted in V_1, one star per vertex in V_1. Then, the optimal value satisfies (recall that $n = (1+\Delta)n_1$)

$$\tau^*(G) = \frac{n}{\Delta+1} = n_1 \tag{47}$$

Denote by V_1' the set of vertices of V_1 that are changed in order to be included in the final solution (i.e., the vertices introduced in the solution after the backtracking). Then, solution C can be written as $C = V_1' \cup V_2 \setminus \Gamma(V_1')$ for a total cost of

$$v(C) = \left|V_1'\right|^2 + \Delta\left(n_1 - \left|V_1'\right|\right) \tag{48}$$

Consequently, player 2 chooses, at best, a set V_1' of cardinality

$$\beta^* \in \operatorname*{Argmin}_{\lambda \in \left[0, \frac{n}{\Delta+1}\right]} \left\{\beta^2 - \Delta\beta + \frac{\Delta n}{\Delta+1}\right\} \tag{49}$$

Define $\Delta = 2n_1$. One can easily show that expression (49) implies $\beta^* = n_1$, and combination of expressions (47) and (48) with $\beta^* = n_1$ induces competitivity ratio satisfying

$$\frac{v(C)}{\tau^*(G)} = n_1 = \frac{-1+\sqrt{1+8n}}{4} \geqslant \sqrt{2n} \tag{50}$$

where the last inequality in expression (50) holds because function $\sqrt{\cdot}$ being concave, $(-1/4) + \sqrt{(1+8n)/2} \geqslant (-1/4) + ((\sqrt{1}+\sqrt{8n})/2) \geqslant \sqrt{2n}$. This completes the proof of the theorem. ∎

6 A simple on-line model based upon edges

We consider in this section an on-line model supposing that the input-graph is revealed by means of its edges. They arrive one at a time. For every new edge, the links of its endpoints with the endpoints of the edges already present are also revealed. We suppose that $|E|$ is known in advance, we set $E = \{e_1, \ldots, e_{|E|}\}$, where e_i are numbered in order of their arrival, and devise the following algorithm.

```
BEGIN *E_OLVC*
      t ← 1;
      C ← X(e_t);
      FOR t ← 2 TO |E| DO IF X(e_t) ∩ C = ∅ THEN C ← C ∪ X(e_t) FI OD
      OUTPUT C;
END.  *E_OLVC*
```

As one can see from the algorithm above, the irrevocability in the construction of the on-line solution deals with the endpoints of an edge as a whole. With respect to a model based upon arrival of vertices it is as one allows, for every edge arriving, a backtracking of level one.

Proposition 2. *The competitivity ratio of algorithm E_OLVC against an optimal off-line algorithm is bounded above by 2. This bound is tight.*

Proof. Assume that the IF-instruction of algorithm E_OLVC is executed q times. It is easy to see that the q edges entailing these executions form a maximal matching of G. The set C finally computed by the algorithm verifies $|C| = 2q$. On the other hand, by expression (5), $\tau(G) \geqslant q$. The competitivity bound 2 is then immediately deduced.

Consider a star revealed edge-by-edge. Algorithm `E_OLVC` will introduce in C the endpoints of the first edge revealed and no new vertex will be introduced in C later. The optimal vertex-cover for any star consists of its center. So here, the bound 2 is attained. ∎

It is easy to see that the on-line model just described is equivalent to the one where all vertices are present from the beginning of the game and edges are presented one-by-one. Here, whenever an edge arrives none of the endpoints of which are in C, then both of its endpoints enter C.

7 Conclusions

On-line computation is actually a very active area of the theoretical computer science and largely interests the operational researchers from both theoretical and practical points of view, since the mathematical problems here emanate from models expressing reality more richly than the conventional ones. The vertex-covering problem dealt in this paper is one of the central problems in combinatorial optimization in its off-line version. As we have seen at the beginning of the paper, it remains very natural even in its on-line version. There exists a number of open problems that seem interesting for further studies. First of all, the improvement, if possible, of the competitivity ratios obtained and the achievement of lower bounds for the case where $t < n$. Also, a further generalization of the vertex-covering is the one where we consider weights on the vertices of the input-graph and we search for a minimum total-weight vertex cover. In the company model presented in section 1, this generalization has a very natural interpretation if we consider that the manufacturing of an order has its proper cost and the company wishes to minimize the cost of the manufacturing in subcontracting. Perform a sensitive analysis of on-line algorithms for this weighted version of LVC seems to us a very interesting open problem.

References

[1] G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, and M. Talamo. Algorithms for the on-line traveling salesman problem. *Algorithmica*, 29(4):560–581, 2001.

[2] R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Ann. Discrete Appl. Math.*, 25:27–46, 1985.

[3] C. Berge. *Graphs and hypergraphs.* North Holland, Amsterdam, 1973.

[4] M. Demange, X. Paradon, and V. T. Paschos. On-line maximum-order induced hereditary subgraph problems. In V. Hlaváč, K. G. Jeffery, and J. Wiedermann, editors, *SOFSEM 2000—Theory and Practice of Informatics*, volume 1963 of *LNCS*, pages 326–334. Springer-Verlag, 2000.

[5] R. El-Yaniv, A. Fiat, R. E. Karp, and G. Turpin. Competitive analysis of financial games. In *Proc. FOCS'92*, pages 327–333, 1992.

[6] A. Fiat and G. Wœginger, editors. *Online algorithms: the state of the art*, volume 1442 of *LNCS*. Springer-Verlag, 1998.

[7] A. Gyárfás and J. Lehel. Online and first-fit colorings of graphs. *J. Graph Theo.*, 12:217–227, 1988.

[8] M. M. Halldórsson and M. Szegedy. Lower bounds for on-line graph coloring. *Theoret. Comput. Sci.*, 130(1):163–174, 1994.

[9] S. Irani and A. R. Karlin. Online computation. In D. S. Hochbaum, editor, *Approximation algorithms for NP-hard problems*, chapter 13, pages 521–565. PWS Publishing Company, 1997.

[10] L. Lovász, M. Saks, and W. Trotter. An on-line graph coloring algorithm with sublinear performance ratio. *Discrete Math.*, 75(1-3):319–325, 1989.

[11] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity.* Prentice Hall, New Jersey, 1981.

[12] D. Sleator and R. E. Tarzan. Amortized efficiency of list update and paging rules. *Commun. ACM*, 28(2):202–208, 1985.

A Proof of theorem 6

The main idea of the proof is analogous to the one of the proof of item 2 in theorem 2. The extra difficulty here is because of $t = o(n)$. It should be noted that the fact $t = n$ in theorem 2 importantly simplifies the respective proof.

As in section 3.3, we consider a two-player game. Player 1 reveals the instance by clusters while player 2 constructs the solution, i.e., it partitions V_i into two subsets: S_i and C_i, the former one denoting the set of the independent vertices of V_i and the latter one denoting the set of the covering vertices of V_i. We denote by PLAY2 the procedure, representing construction of the decision of player 2, about the partition of V_i. In other words, the decision of player 2 will be denoted by $(\mathtt{C_i}, \mathtt{S_i}) \leftarrow \mathtt{PLAY2}(\mathtt{V_i})$. Moreover, we call *badly covered* a d-leaves star whose all leaves are included in the vertex cover under construction. The revealing strategy of player 1 is based upon the following module denoted by ONE_VERTEX and dealing with the revealing of a single vertex. It is called with inputs $\Delta \in \mathbb{N}$, $G[\cup_{j=1,\dots,i} V_i]$, i.e., the graph already revealed, and two sets of vertices R_S and R_C; it returns a new vertex y, its links with the vertices already revealed and the sets R_S and R_C updated. The neighbors of x considered in the third IF of algorithm ONE_VERTEX deal with the graph already revealed. For reasons of simplicity, we set in what follows $G[\cup_{j=1,\dots,i} V_i] = [G]_i$.

```
BEGIN *ONE_VERTEX*
      IF |R_C| = Δ THEN y is connected to any vertex of |R_C|;
                ELSE IF |R_S| ≠ ∅ THEN choose x ∈ R_S;
                                        y is linked only with x;
                                        IF x has Δ neighbors THEN R_S ← R_S \ x FI
                                   ELSE y is isolated;
      FI
END   *ONE_VERTEX*
```

In the overall revealing algorithm, we use an additional module called $\mathtt{UPDATE}(\mathtt{X_1}, \mathtt{Y_1}, \mathtt{X_2}, \mathtt{Y_2})$ and specified in what follows.

```
BEGIN *UPDATE*
      IF X_1 contains an isolated vertex THEN add it in X_2 FI
      IF Y_1 contains an isolated vertex THEN add it in Y_2 FI
END   *UPDATE*
```

The final graph G has a form analogous to the one of figure 1. It consists of:

- Δ blocks of size $\Delta + 1$, the $\Delta - 1$ first ones being stars and the last one being just a tree and
- a root-vertex such that the whole graph is a tree of degree Δ

Assume $\Delta \in \mathbb{N}$, set $n = \Delta(\Delta + 1) + 1$ and $t = \lceil 2\Delta \log \Delta \rceil + (2 + K')(1 + \Delta) + \Delta + 2$, for $K' = \lceil 1 + 2 \log \Delta \rceil$, and consider the following strategy played by player 1 for revealing the graph in t steps.

```
BEGIN *GAME*
(1)   K' ← ⌈1 + 2logΔ⌉;
(2)   K ← Δ − 2K' − 3;
(3)   S ← ∅;
```

```
C ← ∅;
i ← 0;
A ← 0;
*PHASIS 1: main phasis*
WHILE A < KΔ DO
        i ← i + 1;
        V_i ← independent set of size ⌈K − A/Δ⌉;
        (C_i, S_i) ← PLAY2(V_i);
        S ← S ∪ S_i);
        C ← C ∪ C_i);
        A ← A + Δ|S_i| + |C_i|;
OD
r ← A − KΔ;
R_C^1 ← a subset of C of cardinality r;
X_S ← S;
X_C ← C \ R_C^1;
R_S^1 ← ∅;
t ← ⌈2ΔlogΔ⌉ − i;
*PHASIS 2: adjustment of the number t of steps*
FOR j ← 1 TO t DO
     y ← ONE_VERTEX(Δ, [G]_i, R_S^1, R_C^1);
     i ← i + 1;
     V_i ← {y};
     (C_i, S_i) ← PLAY2(V_i);
     *in this case |V_i| = 1*
     S ← S ∪ S_i;
     C ← C ∪ C_i;
     A ← A + Δ|S_i| + |C_i|;
     UPDATE(S_i, C_i, R_S^1, R_C^1)
OD
*PHASIS 3: adjustment of the congruence of|A|modΔ with constant number of clusters*
R_C^2 ← ∅;
R_S^2 ← ∅;
FOR j ← 1 TO K′(1 + Δ) DO
     IF A ⩽ Δ(K + K′) THEN s ← 1
                       ELSE s ← 2;
                            V_i^1 ← ∅;
     FI
     FOR k ← s TO 2 DO
          y ← ONE_VERTEX(Δ, [G]_i, R_S^k, R_C^k);
          V_i^k ← {y};
     OD
     V_i ← V_i^1 ∪ V_i^2;
     i ← i + 1;
     (C_i, S_i) ← PLAY2(V_i);
     *in this case |V_i| ⩽ 2*
     S ← S ∪ S_i;
     C ← C ∪ C_i;
     A ← A + |C_i ∩ V_i^1|;
```

```
        FOR k ← s TO 2 DO UPDATE(S_i ∩ V_i^k, C_i ∩ V_i^k, R_S^k, R_C^k) OD
    v_3 ← |R_C^1 ∪ R_C^2|;
*PHASIS 4: from R_S^1, R_S^2, R_C^1, R_C^2 to R_S, R_C*
    R_S ← ∅;
    R_C ← R_C^1 ∪ R_C^2;
    FOR j ← 1 TO 2(Δ + 1) − v_3 DO
        i ← i + 1;
        y ← ONE_VERTEX(Δ, [G]_i, R_S, R_C);
        V_i ← {y};
        (C_i, S_i) ← PLAY2(V_i);
        S ← S ∪ S_i;
        C ← C ∪ C_i;
        UPDATE(S_i, C_i, R_S, R_C);
    OD
    R_S^1 ← R_S^1 ∪ R_S^2;
    WHILE R_S^1 ≠ ∅ DO
          i ← i + 1;
          y ← ONE_VERTEX(Δ, [G]_i, R_S^1, ∅);
          V_i ← {y};
          (C_i, S_i) ← PLAY2(V_i);
          C ← C ∪ C_i;
    OD
*PHASIS 5: last block*
    bool ← false;
    j ← 0;
    REPEAT
           y ← ONE_VERTEX(Δ, [G]_i, R_S, R_C);
           i ← i + 1;
           j ← j + 1;
           V_i ← {y};
           (C_i, S_i) ← PLAY2(V_i);
           S ← S ∪ S_i;
           C ← C ∪ C_i;
           UPDATE(S_i, C_i, R_S, R_C)
           bool ← (R_S ≠ R)
           IF C_i contains an isolated vertex THEN add x in R_C FI
    UNTIL  bool OR j = 1 + Δ;
    IF bool THEN i ← i + 1;
                  V_i ← {y} where y is connected to x and to any vertex of R_C;
                  (C_i, S_i) ← PLAY2(V_i);
                  C ← C ∪ C_i;
    FI
    FOR k ← 1 TO Δ−j DO
        i ← i + 1;
        V_i ← {y} where y is connected only to x;
        (C_i, S_i) ← PLAY2(V_i);
        C ← C ∪ C_i;
    OD
*PHASIS 6: completion of G by the last cluster*
```

```
(94)  i ← i + 1;
      *V_i is the set of vertices to be revealed*
(95)  V_i ← ∅;
(96)  for any x ∈ X_S, V_i contains Δ vertices connected only to x;
(97)  arbitrarily partition vertices of X_C into clusters of Δ vertices;
(98)  for any of the clusters created in line (97) add one vertex in V_i;
(99)  for each vertex v added in line (98)
          link v with the vertices of the corresponding cluster;
(100) for any connected component c of the graph already revealed
          add to G a vertex connected to a vertex of c of degree 1;
(101) (C_i, S_i) ← PLAY2(V_i);
(102) S ← S ∪ S_i;
(103) C ← C ∪ C_i;
END   *GAME*
```

Lemma 3.

1. *Lines (7) to (20) (*`PHASIS 1`*) of* `GAME` *take at most* $\lceil 2\Delta \log \Delta \rceil$ *steps. At line (20),* $K\Delta \leqslant A < (K+1)\Delta$, $r < \Delta$, $|C_1|/\Delta \in \mathbb{N}$ *and, moreover,* $|S_1| + |C_1|/\Delta = K$.

2. *At line (30) (end of* `PHASIS 2`*),* $i = \lceil 2\Delta \log \Delta \rceil$ *and* $A \leqslant (K' + K)\Delta$.

Proof of item 1. From line (9) one gets $\Delta|S_i| + |C_i| \leqslant \Delta|V_i| < (K+1)\Delta - A$. Hence, the current value of A satisfies $A < (K+1)\Delta$, so $r < \Delta$. On the other hand, clearly, $A = \Delta|S| + |C|$. Consequently, at the end of the `WHILE`-loop in line (14), one has $|S| \leqslant K$ and, moreover, $|C| = q\Delta + r$ with $q + |S| = K$. So, $|C_1| = q\Delta$.

To conclude the proof of item 1, we have to show that the `WHILE`-loop of line (7) is executed at most $\lceil 2\Delta \log \Delta \rceil$ times. Denote by A_i the value of A at the end of the ith execution of the loop. Sequence $(A_i)_i$ satisfies, $\forall i$, such that $A_i < K\Delta$:

$$A_{i+1} \geqslant A_i + K - \frac{A_i}{\Delta} \tag{A.1}$$

Let now $B_i = K\Delta - A_i$. Sequence $(B_i)_i$ satisfies (using expression (A.1)) the following induction rule

$$\begin{cases} B_0 &= K\Delta \\ B_{i+1} &\leqslant B_i\left(1 - \frac{1}{\Delta}\right) \quad \forall i \text{ such that } B_i > 0 \end{cases} \tag{A.2}$$

From expression (A.2), one can deduce that, $\forall i$, $B_{i-1} > 0$ and $B_i \leqslant K\Delta(1-\Delta^{-1})^i$. Furthermore, from expressions (A.1)) and (A.2), B_i becomes non-positive for $i > \log(K\Delta)/-\log((1-\Delta^{-1})^i)$. This last quantity is smaller than $2\Delta\log\Delta$. Consequently, the `WHILE`-loop of line (7) is not executed more than $\lceil 2\Delta \log \Delta \rceil$ times and the proof of item 1 is complete.

Proof of item 2. The value of i claimed follows immediately from the total number of the iterations of the `FOR`-loop. At each iteration, A does not increase by more than 1. Therefore, using item 1, $A < (K+1)\Delta + \lceil 2\Delta \log \Delta \rceil \leqslant (K + K')\Delta$ and the proofs of item 2 and of the lemma are complete. ∎

From algorithm `ONE_VERTEX` we can deduce that at line (30), the graph revealed consists of $|X_S|$ isolated independent vertices, of $|X_C|$ (a multiple of Δ) isolated covering vertices, of s_2 badly covered stars of size $\Delta + 1$, of σ_2 badly covered stars of size $u_2 + 1$, $u_2 < \Delta$ and of v_2 isolated covering vertices. Moreover, $\sigma_2 = |R_1^S| \in \{0, 1\}$ and $v_2 = |R_1^C| \leqslant \Delta$.

Lemma 4. *At line (49) (end of `PHASIS 3`), $i = K'(1+\Delta) + \lceil 2\Delta \log \Delta \rceil$ and $A = \Delta(K + K')$.*

Proof. The value of i immediately follows from the total number of iterations of the `FOR`-loop (lines `(33)` to `(41)`). From algorithm `ONE_VERTEX`, if $s = 1$ from line `(31)` to `(49)`, then the graph revealed during every loop with $k = 1$ consists of λ_1 badly covered stars of size $\Delta + 1$, $\sigma'_1 \leqslant 1$ badly covered stars of size $1 + u'_1$, $u'_1 < \Delta$ and v'_1 isolated covering vertices. The increasing of A corresponds to the number of covering vertices added in C during execution of lines `(31)` to `(49)`, and this number is equal to $K'\Delta$. So, $A \geqslant \Delta K + \Delta K'$. If $s = 2$ at line `(49)`, we also have $A = (K + K')\Delta$ and the proof of the lemma is complete. ∎

We so have $A = \Delta(K + K') = |X_S|\Delta + |X_C| + s_2\Delta + \lambda_1\Delta + u'_1 + v'_1 \Rightarrow u'_1 + v'_1 \in \{0, \Delta\}$. Furthermore, the vertices revealed during every loop with $k = 2$ induce a subgraph consisting of λ_2 badly covered stars of size $\Delta + 1$, $\sigma'_2 \leqslant 1$ badly covered stars of size $1 + u'_2$, $u'_2 < \Delta$ and v'_2 isolated vertices, with $u'_2 + v'_2 \in \{0, \Delta\}$. The graph revealed until the end of `PHASIS 3` (line `(49)`), consists of $|X_S|$ isolated independent vertices, of $|X_C|\Delta$ isolated covering vertices, s_3 badly-covered stars of size $\Delta + 1$, of $\sigma_3 \leqslant 1$ badly covered stars of size $1 + u_3$, $u_3 < \Delta$, of $\sigma'_3 \leqslant 1$ badly covered stars of size $1 + u'_3$, $u'_3 < \Delta$, and of $v_3 \leqslant 2\Delta$ isolated covering vertices. Furthermore, $v_3 + u_3 + u'_3 \in \{0, \Delta, 2\Delta\}$ and $\lambda_1 + \lambda_2 + \sigma'_1 + \sigma'_2 = 2K'$. Really, $\sigma'_1 = |R^1_S|$, $\sigma'_2 = |R^2_S|$ and $v'_1 + v'_2 = |R^1_C \cup R^2_C| = v_3$.

Lemma 5. *From line (50) to line (68) (`PHASIS 4`), $2(\Delta + 1)$ clusters have been revealed. Moreover, at line (68), $|R_C| < \Delta$.*

Proof. The lemma follows immediately from the fact that if $v_3 \neq 0$, then $v_3 = 2\Delta - v'_1 - v'_2$. ∎

We now describe the graph revealed from line `(68)` to the end of algorithm `GAME` (`PHASE 5` and `PHASE 6`). It always contains the $|X_S|$ isolated independent vertices, the $|X_C|$ isolated covering vertices; it also contains $2K' + 2$ or $2K' + 1$ (depending on whether R_C is empty or not) badly covered stars of size $\Delta + 1$. If $R_C \neq \emptyset$, then $R_S = \{x\}$ and the graph contains one badly covered star, rooted at x, of size $1 + \Delta - |R_C|$. Remark finally that from line `(68)` to the end of algorithm, $\Delta + 2$ clusters are revealed. So, the total number of clusters is exactly $N = \lceil 2\Delta \log \Delta \rceil + (K'+2)(\Delta+1) + \Delta + 2 = O(\Delta \log \Delta)$. From line `(69)` to line `(93)` (`PHASIS 5`), a tree of size $\Delta + 1$ has been revealed with only one vertex in S (this vertex is x). Then, the completion of G at lines `(94)` to line `(103)` (`PHASIS 6`) is such that the whole graph contains $K + 2K' + 2 = \Delta - 1$ badly covered stars of size $\Delta + 1$, plus the tree just mentioned, plus one root connected to one vertex in each of the Δ connected components revealed. Consequently, the final graph is a tree of maximum degree Δ and of order $\Delta(\Delta + 1) + 1$; so $N = O(\log n \sqrt{n})$.

The final vertex cover constructed by player 2 has size $|C| \geqslant \Delta^2$ ($|S| \leqslant \Delta + 1$). On the other hand, a minimum vertex cover of the graph is of size $\Delta + 2$ and the proof of theorem 6 is complete.