R A P P O R T   D E   R E C H E R C H E

L R I

SEPARATION COMBINATOIRE DISJONCTIVE DANS UN ALGORITHME D'ARBRE DE SOUS-GRADIENT POUR LE PROBLEME DCMST AVEC DES BORNES VNS-LAGRANGIENNES

ANDRADE R / FREITAS A

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud –LRI

11/2012

Rapport de Recherche N° 1552

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 650
91405 ORSAY Cedex (France)

Laboratoire de Recherche en Informatique, Université de Paris-Sud
Bât. 650, 91405, Orsay Cedex France.

Rapport de Recherche

Titre

Séparation combinatoire disjonctive dans un algorithme d'arbre de sous-gradient pour le problème DCMST avec des bornes VNS-Lagrangiennes [1]

Auteurs

Rafael Andrade et Adriano Freitas[2]

2012

[1]Projet réalisé en collaboration avec l'Université Fédérale de Ceará (Fortaleza - Brésil). Recherche financée par le Conseil National de Recherche Scientifique (CNPq/Brésil – Proc. 201347/2011-3).

[2]Programa de Mestrado e Doutorado em Ciência da Computação, Universidade Federal do Ceará, Campus do Pici - Bloco 910, 60455-760, Fortaleza, Ceará, Brasil.

Résumé : Le problème de l'arbre couvrant de poids minimum avec contraintes de degré (DCMST) dans un graphe non orienté $G = (V, E)$ ayant pour ensemble de noeuds V et pour ensemble d'arêtes pondérés E, avec le poids d'une arête $e \in E$ étant donné par $c_e \geq 0$, consiste en trouver un arbre couvrant de poids minimum de G soumis à des contraintes de degré maximal $d_v \in \mathbb{N}$ sur le nombre d'arêtes connectées à chaque noeud $v \in V$. Nous proposons une heuristique VNS-Lagrangienne plus performante que les heuristiques les plus connues dans la littérature pour ce problème ainsi qu'un algorithme exact en arbre de sous-gradient (SGT) afin de prouver l'optimalité des solutions trouvées. L'algorithme SGT utilise une relaxation combinatoire pour évaluer des bornes inférieures pour chaque solution relaxée des nœuds dans l'arbre SGT. En outre, nous proposons un nouveau schéma de branchement qui sépare une solution relaxée entière (i.e. un arbre couvrant) du domaine de arbres couvrants faisables tout en générant des nouvelles partitions disjointes. Nous montrons l'optimalité pour des nombreuses instances de la littérature et améliorons la valeur des limites inférieures et supérieures pour d'autres instances dont leur solution optimale reste inconnue.
Mots-clés: arbre de sous-gradient, DCMST, heuristique VNS-Lagrangienne.

Disjunctive combinatorial branch in a subgradient tree algorithm for the DCMST problem with VNS-Lagrangian bounds

Rafael Castro de Andrade [1]

Department of Statistics and Applied Mathematics, Federal University of Ceará, Campus do Pici, BL 910, CEP 60.455-760 - Fortaleza, Ceará, Brazil.

Adriano Tavares de Freitas [2]

Department of Computer Science, Federal University of Ceará, Campus do Pici, BL 910, CEP 60.455-760 - Fortaleza, Ceará, Brazil.

Abstract

The degree constrained minimum spanning tree (DCMST) problem in an undirected graph $G = (V, E)$ of set of nodes V and set of weighted edges E, with the weight of an edge $e \in E$ being denoted by $c_e \geq 0$, consists in finding a minimum spanning tree of G subject to maximum degree constraints $d_v \in \mathbb{N}$ on the number of edges connected to each node $v \in V$. We propose a VNS-Lagrangian heuristic that outperforms the best known heuristics in the literature for this problem and an exact subgradient tree (SGT) algorithm in order to prove solution optimality. The SGT algorithm uses a combinatorial relaxation to evaluate lower bounds on each relaxed solution. Thus, we propose a new branching scheme that separates an integral relaxed solution from the domain of spanning trees while generating new disjoint SGT-node partitions. We prove optimality for many benchmark instances and improve lower and upper bounds for the instances whose optimal solution remain unknown.

Keywords: subgradient tree, DCMST, VNS-Lagrangian heuristic.

1 Introduction

The degree constrained minimum spanning tree (DCMST) problem in an undirected graph $G = (V, E)$, V is the set of nodes and E is the set of weighted edges, with the weight of an edge $e \in E$ being denoted by $c_e \geq 0$, consists in finding a minimum spanning tree of G subject to a maximum degree constraint $d_v \in \mathbb{N}$ on the number of edges connected to each node $v \in V$. We know that when $d_v = 2$, for all $v \in V$, this problem is equivalent to find a Hamiltonian path of minimum weight in G, which is NP-hard [5].

The reader is referred to [7], [2], [8], [1], [3] and [4] for a selective literature review on applications and solution approaches for this problem.

The contribution of this work is twofold. First, we develop a VNS-Lagrangian heuristic based on a dynamic variable neighborhood descent (VND) method. The core of our algorithm is the Lagrangian heuristic in [1] where we introduce new dynamic local search techniques to overcome local optima solutions, adapted from the VNS/VND techniques from [9]. Second, we propose an exact subgradient tree (SGT) method that uses a combinatorial relaxation of the problem obtained by dropping the degree constraints and incorporating them into the objective function of the relaxed problem by means of Lagrange multipliers. The relaxed subproblems in the SGT are solved by using a subgradient optimization algorithm [6]. As the solution of a given subproblem is integral (a spanning tree possibly violating the degree constraints of the original problem), we propose a new and novel branching technique for partitioning the space of feasible (for the subproblem) solutions in disjoint regions of spanning trees while separating any relaxed node solution. The reader should notice that a classic branching by dichotomy for this problem means fixing some edge of the relaxed solution (a spanning tree) in zero or in one to create two new partitions, which is known to be very inefficient. The idea we use here is quite general and can be extended to any divide-and-conquer approach that uses a combinatorial relaxation.

We show that the exact SGT approach is competitive with the one in [4] and that the VNS-Lagrangian heuristic outperforms all the best known heuristics for the DCMST problem.

[1] Email: rca@lia.ufc.br
[2] Email: adrianoblue@gmail.com

2 Problem formulation

In model (P) below let $E(S) \subseteq E$, for $S \subseteq V$, represent the set of edges with both extremities in S and $\delta(v) \subseteq E$, for $v \in V$, represent the set of nodes adjacent to v in G. Let x_e, for all $e \in E$, be binary variables representing if an edge e belongs ($x_e = 1$) or not ($x_e = 0$) to a minimum degree constrained spanning tree of G. The problem formulation [1] is

$$(1) \qquad (P) \quad \min \sum_{e \in E} c_e x_e$$

$$(2) \qquad s.t. \sum_{e \in E} x_e = |V| - 1,$$

$$(3) \qquad \sum_{e \in E(S)} x_e \leq |S| - 1, \;\; S \subset V,$$

$$(4) \qquad \sum_{e \in \delta(i)} x_e \leq d_i, \;\; \forall\, i \in V.$$

$$(5) \qquad x_e \in \{0,1\} \;\; \forall\, e \in E.$$

Associating Lagrange multipliers $\lambda \in \mathbb{R}_+^{|V|}$ with (4) we obtain the relaxed problem

$$(6) \qquad (R) \quad \min \sum_{e=(i,j) \in E} (c_e + \lambda_i + \lambda_j) x_e - \sum_{i \in V} \lambda_i d_i$$
$$s.t. \; (2), (3), (5).$$

Problem (R) gives lower bounds on the solution of (P) for a given $\lambda \geq 0$. In [1] the reader can find a subgradient optimization procedure to obtain lower bounds on the solution of (P).

Now consider the Lagrangian relaxation (R_F) of a non empty restriction (P_F) of (P) below, where we fix some variables $x_e = 1$ for $e \in F_1$ and $x_e = 0$ for $e \in F_0$, with $F_1 \cap F_0 =$, $F_1, F_0 \subset E$, $|F_1| < |V| - 1$. Define primal multipliers $\beta_v = 0$ if $\sum_{u \in \delta(v) \cap F_1} x_{uv} < d_v$ and $\beta_v = \infty$, otherwise.

$$(7) \qquad (R_F) \quad \min \sum_{e=(i,j) \in E} (c_e + \lambda_i + \lambda_j + \beta_i + \beta_j) x_e - \sum_{i \in V} \lambda_i d_i$$
$$s.t. \;\; x_e = 1, \forall\, e \in F_1, \quad x_e = 0, \forall\, e \in F_0$$
$$(2), (3), (5),$$

Proposition 2.1 *For a given fixing of variables as above, the solution value of (R_F) is a lower bound for the restricted version (P_F) of (P). Moreover, it is at least equal to the value obtained by solving (R_F) without considering the* *primal multipliers β.*

We explore this proposition, whose proof is omitted here, in the exact SGT scheme presented hereafter to avoid allocating an excessive quantity of memory to represent the list of prohibited edges in any solution associated to a given partition of the space of feasible spanning trees associated to the sets F_0 and F_1. Note that exploring this proposition does not mean that some node $u \in V$, not saturated by the edges in F_1, could not violate its degree constraint. In fact, edges adjacent to u with perturbed Lagrangian costs presenting smaller costs among all the other non fixed edges, can belong to the Lagrangian relaxed solution and they can occur in larger number than the upper limit d_u. In such situations, we can divide the partition represented by F_0 and F_1 into smaller ones in an exact algorithm.

Proposition 2.2 *Let v be a node violating the degree constraint d_v in a Lagrangian relaxed solution (a tree $T_{\mathcal{P}}$) related to a given partition $\mathcal{P}$ associated to F_0 and F_1. Consider the p edges incident to v in $T_{\mathcal{P}}$ in the order they appear in the list $L_v = \langle e_1, e_2, ..., e_p \rangle$. The partitions $\mathcal{P} \cap \{x \,|\, x_{e_1} = 0\}$ and $\mathcal{P} \cap \mathcal{P}_j$, for each $j \in \{2, ..., p\}$, with $\mathcal{P}_j = \{x \,|\, x_{e_i} = 1, i = 1, ..., j-1, x_{e_j} = 0\}$, are disjoint. Moreover, the partitions $\mathcal{P} \cap \mathcal{P}_j$, for $j \geq d_v + 1$ are infeasible for the original problem and can be discarded by an exact algorithm for the DCMST problem.*

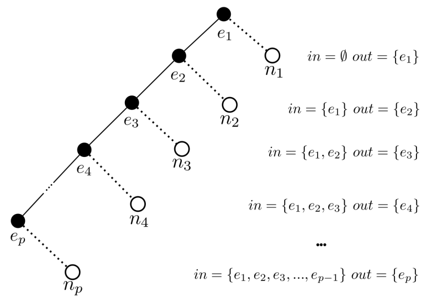
Figure 1 illustrates the partitioning process of Proposition 2.2. Dotted lines represent that an edge is fixed at zero. Dashed lines represent that an edge is fixed at one. In this figure we assume initially $F_0 = F_1 = \emptyset$ for the father partition (F_0 and F_1, for each child partition, are represented by the lists *out* and *in* of edges fixed at zero and one, respectively). All possible partitions $n_1, \cdots, n_p$ are represented and some of them can be infeasible for the original problem (the ones with $|in| > d_v$).

Note that this partitioning scheme is quite general and can be applied to exact algorithms where the employed relaxation is a combinatorial problem (i.e. always has an integer relaxed solution).

3 VNS-Lagrangian heuristic

The idea of our heuristic approach is as usual. In the VNS-Lagrangian heuristic, that is an improved version of the heuristic in [1] to overcome local optima solutions, first we relax the degree constraints of the problem and incorporate them to the objective function by using Lagrange multipliers. A subgradient

Fig. 1. Disjunctive combinatorial branch: the new disjoint partitions are the leaves in this tree representation.



optimization procedure is used to obtain lower bounds on the problem solution. The Lagrange multipliers are used to perturb the original edge costs in order to obtain feasible degree constrained trees in the graph with modified edge costs. We use an adapted version of the Kruskal algorithm to deal with the degree constraints and a dynamic VNS-VND procedure based on [9] to improve feasible solutions. The VNS has a shaking phase, where a solution T' is obtained from the incumbent solution T from a neighborhood structure $N_1^{VNS}, ..., N_{k_{max}}^{VNS}$, where $N_k^{VNS}(T)$ is the set of feasible trees which are neighbors to T having exactly $k \leq k_{max}$ different edges, with k_{max} given. In a second phase, we try to improve T' with a Dynamic VND procedure based on three neighborhood structures $N_{VND}^{(1)}$, $N_{VND}^{(2)}$ and $N_{VND}^{(3)}$ of a feasible solution T. For this, we classify the edges which are not in T into three groups: $E0$ of edges with no saturated extremity; $E1$ of edges with exactly one saturated extremity and $E2$ of edges with both saturated extremities, with $E(G) = E(T) \cup E_0 \cup E_1 \cup E_2$. An edge exchange is defined by a pair of edges $(e, \bar{e})$, where $e \in E(T)$ and $\bar{e} \in E(G) \setminus E(T)$.

Neighborhood $N_{VND}^{(1)}(T)$ is the set of feasible trees which can be obtained from T by applying one edge exchange $(e, \bar{e})$. Basically, an edge $\bar{e} \in (E0 \cup E1)$ is inserted in T and a cycle C is generated. We remove and edge $e \in E(T)$ from C in order to reestablish feasibility of the degree constraints.

In neighborhood $N_{VND}^{(2)}(T)$ we use two edge exchanges: $(e_1, \bar{e}_1)$ and $(e_2, \bar{e}_2)$. The insertion of an edge $\bar{e}_1 \in (E1 \cup E2)$ generates a cycle. We remove the edge $e_1 \in E(T)$ from this cycle. The second exchange is used if we need to arrange the degree of a node v which remains violated after the first edge exchange.

Neighborhood $N_{VND}^{(3)}(T)$ uses three edge exchanges: $(e_1, \bar{e}_1)$, $(e_2, \bar{e}_2)$ and $(e_3, \bar{e}_3)$. First, we insert an edge $\bar{e}_1 \in E2$ and a cycle is generated. An edge $e_1 \in E(T)$ is removed from this cycle. After the edge exchange $(e_1, \bar{e}_1)$, two nodes v_1 and v_2 may remain violated. The remaining two edge exchanges are used to arrange the degrees of v_1 and v_2.

Note that for DCMST instances with degree constraints $d_v = 2$, for all $v \in V$, the three neighborhoods above have no effect for improving feasible solutions. To overcome such situations, we propose redefining the sets $E0$, $E1$ and $E2$ dynamically based on the current tree after each edge exchange.

Finally, if a better feasible solution is found then we actualize the incumbent solution for the problem and use it as a cut-off value in the SGT exact method.

4 SGT scheme

The SGT is a branch-and-bound search tree algorithm composed of three operations: evaluation, pruning and branching of SGT nodes. In the evaluation process we have two types of lower bounds on each node solution - one given by the value of the Lagrangian relaxed solution (with edge costs perturbed by Lagrange multipliers) and the other by the original edge costs associated to that solution. We determine sequences of Lagrange multipliers in a tree structure following the generation of SGT nodes. These sequences are updated from a father to a child node by taking into account the fixing of the edges and the direction of the gradient obtained at the node relaxed solution. Internal to each SGT subproblem, we iterate the subgradient procedure a certain number of iterations to improve its relaxed bound. SGT nodes are organized in the search tree according to the best bound strategy. The pruning operation allows deleting SGT nodes from the search tree when the bound on the node solution is greater than the value of the best incumbent solution for the problem. In the branching process a given partition (father node) is divided into new smaller and disjointed ones (children nodes). The node we choose to perform the branch is the one with smaller violation of its degree constraint. The novelty here is how we separate a feasible tree from the domain of spanning trees to obtain new disjoint subproblems partitions according to the Proposition 2.2. Observe that we do not use linear relaxation in our work (a combinatorial one is used instead). Moreover, when some edges are fixed at one (thus they must be in the node solution), it is possible that some node v becomes saturated (the number of edges connected to v is equal to d_v). In this case, all remaining unfixed edges incident to v can be fixed at zero. However, saving the information of the edges fixed at zero for each SGT node is not practical since the required memory increases very fast. To overcome this problem, we introduce the concept of primal multipliers in the Proposition 2.1 that are also used to perturb the Lagrange edge costs in order to prevent a known saturated node of having additional edges being incident to it in the relaxed SGT node solution.

5 Computational Results

The C++ algorithms are linked with g++-4.4.3 for Linux Ubuntu in a Core 2 Duo PC with 2.4 GHz / 3GB RAM. In this short paper we report only results for two classes of benchmark instances and compare our results with the ones in [9] and [4]. A more extensive set of experiments including Hamiltonian [1], DE and DR [4] instances will be reported in a complete version of this paper to show the efficiency of our solution approach. The legend in the next tables is self-explained. CPU is in $(min : sec)$ and $gap = 100(UB - LB)/LB$.

Table 1
Computational results for Euclidean instances from [1]

Instance	Results from [9]		VNS-Lagr-Heur				+SGT for gap > 0				Results from [4]			
$\|V\|$_(id)	UB	cpu	LB	UB	cpu	gap	LB	UB	cpu	gap	LB	UB	cpu	gap
100_1	3790	0:00	3790	3790	0:00	0.000					3790	3790	0:00	0.000
100_2	3829	0:00	3829	3829	0:00	0.000					3829	3829	0:00	0.000
100_3	3916	0:00	3916	3916	0:01	0.000					3916	3916	0:00	0.000
200_1	5316	0:06	5316	5316	0:00	0.000					5316	5316	0:00	0.000
200_2	5651	0:08	5647	5647	0:01	0.000					5647	5647	0:02	0.000
200_3	5699	0:06	5698	5698	0:01	0.000					5698	5698	0:01	0.000
300_1	6477	0:50	6475	6477	0:11	0.030	6477	6477	0:12	0.000	6477	6477	0:02	0.000
300_2	6807	0:55	6803	6809	0:48	0.088	6807	6807	20:49	0.000	6805	6807	0:31	0.029
300_3	6430	0:40	6430	6430	0:01	0.000					6430	6430	0:04	0.000
400_1	7414	3:01	7414	7414	0:01	0.000					7414	7414	0:02	0.000
400_2	7783	2:33	7776	7783	1:43	0.090	7782	7782	26:55	0.000	7779	7782	0:32	0.038
400_3	7604	3:19	7602	7604	1:24	0.026	7604	7604	1:33	0.000	7603	7604	0:38	0.013
500_1	8272	9:12	8272	8272	0:03	0.000					8272	8272	0:06	0.000
500_2	8396	10:56	8392	8392	2:01	0.000					8392	8392	0:18	0.000
500_3	8502	8:24	8502	8502	2:07	0.000					8502	8502	0:26	0.000
600_1	9037	10:19	9037	9037	0:01	0.000					9037	9037	0:09	0.000
700_1	9786	16:37	9786	9786	0:02	0.000					9786	9786	0:13	0.000
800_1	10333	29:21	10333	10333	0:07	0.000					10333	10333	0:25	0.000
900_1	10918	36:00	10918	10918	0:07	0.000					10918	10918	0:29	0.000
1000_1	11408	36:05	11407	11408	2:58	0.008	11407	11408	*	0.008	11407	11408	2:48	0.008
2000_1	15670	180:50	15670	15670	12:23	0.000					15670	15670	4:32	0.000
2000_2	16255	208:06	16239	16243	59:03	0.024	16239	16242	*	0.018	16242	16242	45:12	0.000
2000_3	16687	225:29	16668	16670	54:18	0.011	16668	16670	*	0.011	16669	16670	68:56	0.005
2000_4	16445	34:36	16368	16370	64:12	0.012	16368	16370	*	0.012	16369	16370	70:32	0.006
2000_5	16532	193:27	16520	16523	37:54	0.018	16520	16521	220:12	0.006	16520	16521	47:57	0.006

(*) Time limit exceeded (1 day)

In Table 1 we see that the VNS-Lagrangian Heuristic module outperforms the one in [9]. The SGT module acts when our heuristic gap is not null. It obtains the same upper bounds of [4] and proves optimality for three new instances. The SGT lower bounds are competitive with the ones of the non delayed relax and cut exact algorithm of [4].

6 Conclusions

We propose a new VNS-Lagrangian Heuristic and an exact SGT algorithm for the DCMST problem that are competitive with the best known algorithms for this problem. The SGT uses a novel partitioning technique that can be extended to other exact methods exploring combinatorial relaxations for the subproblems. We show how to strength the subproblem relaxed solution with the use of the primal multipliers. On the best of our knowledge, it is the first time these techniques are proposed for solving a NP-hard problem and they showed to be very efficient.

References

[1] Andrade, R., A. Lucena and N. Maculan, *Using Lagrangian dual information to generate degree constrained spanning trees*, Discrete Applied Mathematics **154** (2006), pp. 703–717.

[2] Caccetta, L. and S. Hill, *A branch and cut method for the degree-constrained minimum spanning tree problem*, Networks **37** (2001), pp. 74–83.

[3] Cunha, A. S. and A. Lucena, *Lower and upper bounds for the degree-constrained minimum spanning tree problem*, Networks **50** (2007), pp. 55–66.

[4] Cunha, A. S. and A. Lucena, *A hybrid relax-and-cut/branch and cut algorithm for the degree-constrained minimum spanning tree problem*, Technical report, Departamento de Administração, Universidade Federal do Rio de Janeiro (2008).

[5] Garey, M. and D. Johnson, "Computers and Intractability, A Guide to the Theory of NP-Completeness," Freeman, 1979.

[6] Held, M., P. Wolfe and H. Crowder, *Validation of subgradient optimization*, Math. Programming **6** (1974), pp. 62–88.

[7] Krishnamoorthy, M., A. Ernst and Y. Sharaiha, *Comparison of algorithms for the degree constrained minimum spanning tree*, Journal of Heuristics **7** (2001), pp. 587–611.

[8] Ribeiro, C. and M. Souza, *Variable neighborhood search for the degree-constrained minimum spanning tree problem*, Discrete Appl. Math. **118** (2002), pp. 43–54.

[9] Souza, M. and P. Martins, *Skewed vns enclosing second order algorithm for the degree constrained minimum spanning tree problem*, European Journal of Oper. Res. **191** (2008), pp. 677–690.