Approximation algorithms for forests augmentation ensuring two disjoint paths of bounded length

Victor Chepoi, Bertrand Estellon, Yann Vaxès

Laboratoire d'Informatique Fondamentale de Marseille,
Faculté des Sciences de Luminy, Université de la
Méditerranée, F-13288 Marseille Cedex 9, France,
{*chepoi,estellon,vaxes*}*@lif.univ-mrs.fr*

Abstract. Given a forest $F = (V, E)$ and a positive integer D, we consider the problem of finding a minimum number of new edges E' such that in the augmented graph $H = (V, E \cup E')$ any pair of vertices can be connected by two vertex-disjoint paths of length $\leq D$. We show that this problem and some of its variants are NP-hard, and we present approximation algorithms with worst-case bounds 6 and 4.

Topics. Graph algorithms, approximation algorithms.

1 Introduction and preliminaries

Biconnectivity is a fundamental requirement to the topology of communication networks: a biconnected network survives any single link or node failure (the probability of two or several simultaneous failures in most networks is reasonably small). On the other hand, the communication performances of a network depend of the communication delay between any two nodes of the network. Since the delay of sending a message from one node to another is roughly proportional to the number of nodes (or links) the message has to traverse, it is desirable to route the messages along short paths or paths of bounded length. Therefore a network in which any pair of nodes can be connected by two disjoint paths of bounded length ensure a low communication delay even in case of a single link or node failure. In this paper, we consider the problem of optimal augmentation of networks (more precisely, of their underlying graphs) so that the resulting networks satisfy this connectivity requirement. We show that this augmentation problem is NP-hard even in the case of forests. On the other hand, we provide efficient approximation algorithms for this problem and its variants if the input graph is a forest. Our work continues the research started in [3, 7].

The problem of augmenting a graph to reach biconnectivity by adding a minimum number of new edges is an important graph-algorithmic problem with applications to network reliability and fault-tolerant computing. Eswaran and Tarjan [5] introduced this problem and established that its basic version can be solved efficiently. Subject to additional constraints, the biconnectivity augmentation problem becomes difficult: for example, both the weighted augmentation problem and the optimal augmentation of a planar graph to a biconnected planar graph are NP-hard [5, 8]; for both problems there exist constant factor approximation polynomial algorithms.

The problem of augmenting a graph $G = (V, E)$ to a graph $H = (V, E \cup E')$ of a given diameter D by adding a minimum number of edges is NP-hard for any $D \geq 2$ [3, 9, 10] (and is at least as difficult to approximate as SET COVER). The complexity status of this problem is unknown if the input graph is a forest (or a tree). In this case, [3] presents a factor 2 algorithm for even D and [7] presents a factor 8 algorithm for odd D. Chung and Garey [4] established that if G is a path with n vertices, then the minimum number of added edges is at least $(n-D-1)/(D+1)$ and at most $(n-D+2)/(D-2)$; for some other related bounds see [1]. If, additionally to be of diameter D, the resulting graph H must be biconnected, then the resulting augmentation problem is NP-hard even if the input graph is a tree [3]. For forests, [3] presents a factor 3 approximation algorithm in case of even D and a factor 6 approximation algorithm for odd D. The last result has been improved by Ishii, Yamamoto, and Nagamochi [7] to a factor 4 (plus 2 edges) approximation algorithm. Notice that for trees the performance guarantees of all mentioned algorithms should be much better, however the bottleneck in analyzing them is the difficulty of establishing better lower bounds for the minimum number of added edges; for example, the proof of the above mentioned lower bound for paths [4] is already quite involved.

In this note, we consider three variants of the augmentation problem with additional distance constraints:

Problem A2VDBP (AUGMENTATION with 2 VERTEX-DISJOINT BOUNDED length PATHS): *given a graph $G = (V, E)$ and a positive integer D, add a minimum number of new edges E' such that any pair of vertices can be connected in the augmented graph $H = (V, E \cup E')$ by two vertex-disjoint paths of length $\leq D$.*

Problem A2EDBP (AUGMENTATION with 2 EDGE-DISJOINT BOUNDED length PATHS): *given a graph $G = (V, E)$ and a positive integer D, add a minimum number of new edges E' such that any pair of vertices can be connected in the augmented graph $H = (V, E \cup E')$ by two edge-disjoint paths of length $\leq D$.*

Problem ADCE (AUGMENTATION with DIAMETER CONSTRAINTS in the augmented graph minus an EDGE): *given a graph $G = (V, E)$ and a positive integer D, add a minimum number of new edges E' such that for any edge $e \in E \cup E'$ the diameter of the augmented graph H minus e is at most D.*

The problems A2VDBP, A2EDBP, and ADCE are not equivalent: any feasible augmentation for A2VDBP is a feasible solution to A2EDBP and any feasible solution to A2EDBP is a feasible solution of ADCE but not vice versa, as the following example shows. Let H be a graph consisting of a cycle of length $2R+2$ plus a diagonal cc' connecting two opposite vertices c and c' of this cycle. Removing any edge from H results into a graph of diameter at most $2R-1$, however the neighbors a and b of c different from c' cannot be connected in H by two vertex- or edge-disjoint paths of length $\leq 2R-1$.

In Section 2 we will prove that the problems A2VDBP, A2EDBP, and ADCE are NP-hard already when the input graph is a forest. Based on lower bounds established in Section 3, in Section 4 we present a factor 6 approximation algorithm for all three problems. In Section 5, this algorithm is improved to a factor 4 approximation algorithm for ADCE. Due to space limitations, we will present full proofs and analysis of both algorithms only in the case $D = 2R - 1$.

We conclude this introductory section with a few necessary definitions. A polynomial algorithm is called an *α-factor approximation* algorithm for a minimization problem Π if for each instance I of Π, it returns a solution whose value is at most α times the optimal value $\mathrm{OPT}_{\Pi}(I)$ of Π on I plus a constant not depending of I; see [11]. For a graph $G = (V, E)$, the *length* of a path between two vertices is the number of edges in this path. The *distance* $d(u, v) := d_G(u, v)$ between two vertices u, v of G is the length of the shortest path between these vertices (if u and v are in distinct connected components of G we will set $d(u, v) = \infty$). The *diameter* $\mathrm{diam}(G)$ of G is the largest distance between two vertices of G. For a positive integer k and a vertex $u \in V$ let $B(u, k) = \{v \in V : d(u, v) \leq k\}$ denote the *ball* of radius k centered at u. (For other graph-theoretical notions and notations used but not defined in this text, see [12].) For two vertices u, v of a tree $T = (V, E)$ denote by $P(u, v)$ the unique path of T between u and v. For a vertex x in a rooted tree T with root r, any vertex $y \neq x$ on the path $P(r, x)$ is called an *ancestor* of x. If y is an ancestor of x, then x is a *descendant* of y. For a subset S of vertices of T, the set of *direct descendants* of a vertex $v \in V$ consists of all descendants $u \in S$ of v such that the path $P(u, v)$ does not contain any other vertex from S.

2 NP-completeness

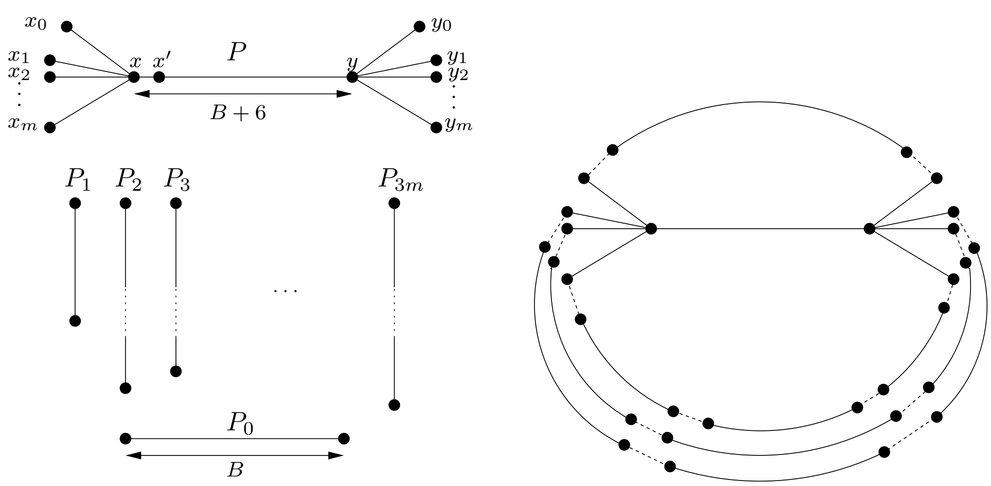
The decision variants of the problems A2VBLP, A2EDBP, and ADCE belong to the class NP. To establish that these problems are NP-complete on forests, we present pseudo-polynomial transformations from the strongly NP-complete problem 3-PARTITION. Then Lemma 4.1 of [6] implies that the augmentation problems are NP-complete as well.

Theorem 2.1 *The problems A2VDBP, A2EDBP, and ADCE are NP-complete on forests.*

PROOF. First we describe a pseudo-polynomial transformation from 3-PARTITION [6] to A2VDBP. Let an instance of 3-PARTITION be given, i.e., a set A of $3m$ elements $a_1, \dots, a_{3m}$, a bound $(8 \leq) B \in \mathbb{Z}^+$, and a size $s(a_i) \in \mathbb{Z}^+$ for each $a_i \in A$ such that $B/4 < s(a_i) < B/2$ and $\sum_{a_i \in A} s(a) = mB$. We construct a forest $F = (V, E)$ as follows: for each $a_i \in A$ introduce a path P_i of length $s(a_i) (> 2)$, additionally consider a path P_0 of length B, and a "bistar" formed by a path P of length $B + 6$ plus $m + 1$ leaves at each end of this path; see Fig. 1. We assert that the set A can be partitioned into m disjoint sets $A_1, \dots, A_m$ such that $\sum_{a_i \in A_j} s(a_i) = B$ for every $1 \leq j \leq m$ if and only if there exists a feasible solution to the problem A2VDBP with $D := 2B + 10$ using at most $4m + 2$ edges. (Our construction is similar to that used in [3] with one difference: we use the additional path P_0 which will force the structure of the augmented graph.)

The forest F has $8m + 4$ leaves. Since a feasible augmentation of F results into a biconnected graph, any leaf of F must be incident to a new edge, therefore this augmentation must contain at least $4m + 2$ edges. If the optimal solution E' of A2BLP uses exactly $4m + 2$ edges, then both ends of added edges are leaves of F. It can be easily seen that the graph $H = (V, E \cup E')$ obtained from F by adding $4m + 2$ edges E' consists of the path P and $m + 1$ ears. An *ear* Ear_i is a path of H between x and y consisting of the edges xx_i, yy_i and 0,1, or several paths P_j ($j \in \{0, 1, \dots, 3m\}$) and some new edges connecting either the end-vertices of two paths or an end-vertex of a path with x_i or y_i.

Let Ear_0 be the ear containing the path P_0. We assert that Ear_0 does not contain other paths of F. If say the path P_j (of length > 2) belongs to Ear_0, then the length of this ear would be at least $B + 2 + 5 = B + 7$. Let x' be the neighbor of x in the path P. Then x' and x_0 cannot be connected by two vertex-disjoint paths of length $\leq D = 2B + 10$, because the shortest path connecting these vertices and not passing via x has length $\geq B + 7 - 1 + B + 5 = 2B + 11 > D$.



(a) Forest F

(b) Augmented graph H

Figure 1: NP-completeness of the problem A2BLP for forests

Thus Ear_0 has length $B+4$. We will show now that each of the ears $\mathrm{Ear}_1, \ldots, \mathrm{Ear}_m$ has length $B+6$. Suppose by way of contradiction that Ear_i has length $\geq B+7$. Due to the structure of H, there exists a unique path between x_i and x' not passing via x : it consists of the subpath of P between x' and y, and the subpath of Ear_i between y and x_i. The length of this path is $\geq B+5+B+6=2B+11>D$, therefore all paths of length $\leq D$ between x_i and x' pass via x, contrary to the admissibility of H. Thus every Ear_i has length $B+6$, i.e., Ear_i is composed of exactly three paths of F of total length B (recall that the length of every path of the forest F is comprised between $B/4$ and $B/2$). The corresponding triplets of A yield a feasible 3-partition.

Conversely, given a feasible solution to 3-PARTITION, we can biconnect F by adding $4m+2$ edges in such a way that in the augmented graph H the ear Ear_0 has length $B+4$ and every other ear Ear_i ($i=1,\ldots,m$) has length $B+6$. We claim that H is a feasible solution of the problem A2BLP. Pick two arbitrary vertices u and v, and denote by C any shortest cycle containing these vertices. If u and v both belong to P or to Ear_0, then C is the union of P and Ear_0, therefore its length is equal to $B+6+B+4=2B+10=D$. If $u \in \mathrm{Ear}_0$ and $v \in \mathrm{Ear}_i$ or $u, v \in \mathrm{Ear}_i$ for some $i>0$, then C is the union of Ear_0 and Ear_i and its length is equal to $B+4+B+6=2B+10=D$. Finally, if $u \in P \cup \mathrm{Ear}_i$ and $v \in \mathrm{Ear}_j$ for distinct $i, j>0$, then C is the union either of P and Ear_j or of Ear_i and Ear_j, and in both cases its length is $2B+12$. We can suppose that $d_F(u,v) \geq 2$ (otherwise we are in one of previous cases), therefore the longest path connecting u and v in C has length $\leq 2B+10$. This establishes the NP-completeness of A2VDBP.

Finally, we briefly describe a pseudo-polynomial transformation from 3-PARTITION to ADCE (or A2EDBP) with $D=3B/2$. The forest F consists of a star with $2m+2$ leaves centered at the vertex x, a path P_0 of length $B+2$, and $3m$ paths P_i of length $s(a_i)$ each. We assert that there exists a 3-partition of A if and only if there exists a feasible solution to ADCE with $4m+2$ edges. Any feasible augmentation H of F using $4m+2$ edges consists of $m+1$ cycles $C_0, C_1, \ldots, C_m$ which pairwise intersects in the central vertex x. Since H is a solution for ADCE, one can show as in previous proof that P_0 is the unique path of F in C_0 and that every other cycle of H contains exactly three paths of the forest F and has length $B+6$. Vice versa, any augmentation H of F obtained from a 3-partition of A using $4m+2$ edges satisfies these two conditions and it can be shown that $\mathrm{diam}(H-\{e\}) \leq D$ for any edge of H. □

3 Lower bounds

For a graph $G=(V,E)$ and a positive integer k, a *vertex k-dominating set* is a set of vertices $C \subseteq V$ such that $\cup_{c\in C} B(c,k) = V$. Analogously, $C \subseteq V$ is an *edge k-dominating set* if the two end-vertices of any edge of G belong to a common ball of radius k centered at a vertex of C. Finding a minimum vertex or edge k-dominating set in a graph is NP-hard, however for trees (and forests) these problems can be solved in linear time [2]. The algorithm can be easily modified to find in linear time a minimum vertex k-dominating set VC_k or a minimum edge k-dominating set EC_k of a forest F with the additional constraint that C contains the set L of all leaves of F. Let $vc_k(F) := |VC_k|$ and $ec_k(F) := |EC_k|$.

Proposition 3.1 *For a forest F,*

(i) $vc_{R-1}(F)/2 \leq OPT_{ADCE}(F) \leq OPT_{A2EDBP}(F) \leq OPT_{A2VDBP}(F)$ *if* $D = 2R-1$.

(ii) $ec_R(F)/2 \leq OPT_{ADCE}(F) \leq OPT_{A2EDBP}(F) \leq OPT_{A2VDBP}(F)$ *if* $D = 2R$.

Proof. In both cases it suffices to establish only the leftmost inequality, because the inequalities $\mathrm{OPT}_{ADCE}(F) \leq \mathrm{OPT}_{A2EDBP}(F) \leq \mathrm{OPT}_{A2VDBP}(F)$ trivially hold for all graphs. Let E' be an optimal augmentation for the problem ADCE, and let C denote the set of end-vertices of the edges from E'. Notice that any leaf x of the forest F belongs to C, otherwise the paths in the graph $H=(V,E\cup E')$ issued from x will use the unique edge e of H incident to x, contrary to the fact that H is a solution of ADCE. Thus $L \subseteq C$. We assert that if $D=2R-1$, then every vertex of F is covered by a ball of radius $R-1$ centered at C, and if $D=2R$, then the end-vertices of every edge of F are at distance $\leq R$ from a vertex of C. Suppose by way of contradiction that in the case $D=2R-1$ there exists a vertex $u \notin \cup\{B_{R-1}(c) : c \in C\}$. Obviously, u is not a leaf of F. Pick a neighbor v of u in F and let $e=uv$. From the choice of u one conclude that the closest to v vertex of C is at distance $\geq R-1$ in F. Since the closest to u vertex of C is at distance $\geq R$ and any path between u and v of the graph $H-e$ uses at least one added edge, we deduce that the distance between u and v in the graph $H-e$ is at least $R+1+(R-1)=2R$, contrary to the assumption that E' is a feasible augmentation for ADCE. Hence $\cup\{B_{R-1}(c) : c \in C\} = V$, yielding $vc_{R-1}(F) \leq |C|$ (analogously one can show that $ec_R(F) \leq |C|$ if $D=2R$). Since $|C| \leq 2|E'|$ (the worst case occurs when E' is a matching on C), we obtain the required inequality. □

An immediate consequence of Proposition 3.1 is that any feasible augmentation for A2VDBP using at most $3vc_{R-1}(F)$ edges for $D=2R-1$ and at most $3ec_R(F)$ edges for $D=2R$ would provide a factor 6 approximation algorithm for each of the problems A2VDBP, A2EDBP, and ADCE. Next section is devoted to the description and analysis of such an algorithm.

4 A factor 6 approximation algorithm

We describe and justify the augmentation algorithm for $D = 2R - 1$ and only outline the changes for the case $D = 2R$. Assume without loss of generality that the input forest $F = (V, E)$ contains at least one edge, otherwise we simply run the algorithm on the forest obtained from F by adding an arbitrary edge. Let L be the set of leaves of F and let $L_0 \subset L$ be the set of leaves constituting one-vertex trees of F. For the rest of this paper, we denote by $d(u, v)$ *the distance in the forest* F between two vertices $u, v \in V$. Suppose that every tree of the forest F containing at least two vertices is rooted at some leaf. Let S be the set of all such roots. The algorithm picks an arbitrary root $r \in S$ and the neighbor r' of r. At the next stage, a minimum vertex $(R-1)$-dominating set C of the forest F containing r' and the set L of leaves is computed. The algorithm proceeds each rooted tree level-by-level starting from its root, and for current vertex $c \in (C - L) \cup S$ it computes the list D_c of its direct descendants in C sorted in increasing order with respect to the distances to c. For each vertex $c' \in D_c$, the algorithm selects 0,1, or 2 vertices (this number depends of the distance $d(c', c)$) on the path $P(c, c')$ between c and c'. The set consisting of C and all selected vertices is grouped into two classes A and A' such that every vertex of the forest F can be connected by two vertex-disjoint paths of length $\leq R - 2$, one going to a vertex of A and another to a vertex of A' (the one-vertex components of F are included in both A and A', while S is included only in A). The algorithm returns the augmentation consisting of all edges of the form ra for $a \in A$ and $r'a'$ for $a' \in A'$. In the case $D = 2R$, the algorithm is completely analogous, except that C is the minimum edge R-dominating set of F containing $L \cup \{r'\}$.

Algorithm A2VDBP(F,D)
Input: A forest $F = (V, E)$ with $E \neq \emptyset$ and an odd integer $D = 2R - 1 \geq 9$.
Output: A set E' of new edges, such that any $u, v \in V$ can be connected in $H = (V, E \cup E')$ by two vertex-disjoint paths of length $\leq D$ and $|E'| \leq 6 \cdot OPT_{A2VDBP}(F)$.

Phase 0: Root every tree with at least two vertices at some leaf and denote by S the set of such roots. Set $A := L_0 \cup S$ and $A' := L_0$. Pick $r \in S$ and the neighbor r' of r.

Phase 1: Compute a minimum vertex $(R-1)$-dominating set C with $L \cup \{r'\} \subseteq C$.

Phase 2: For each $c \in (C - L) \cup S$ compute the list D_c of direct descendants of c in C and sort D_c in increasing order of the distances to c.

Phase 3: Proceed every rooted tree level-by-level. For current vertex $c \in (C - L) \cup S$ traverse the list D_c and to each vertex $c' \in D_c$ apply one of the following rules:

- **case $\mathbf{d(c, c') \leq R - 2}$**: if $c \in A$, then insert c' in A', else insert c' in A.
- **case $\mathbf{R - 1 \leq d(c, c') \leq 2R - 4}$**: pick the vertex $c_1 \in P(c, c')$ at distance $R - 2$ to c'. If $c \in A$, then insert c_1 in A' and c' in A, else insert c_1 in A and c' in A'.
- **case $\mathbf{d(c, c') \geq 2R - 3}$**: pick the vertices $c_1, c_2 \in P(c, c')$, c_1 at distance $R - 2$ and c_2 at distance $2R - 4$ to c'. Find a closest to c_2 vertex c'' of C which is not a descendant of c_2. If $c'' \in A$, then insert c_2, c' in A' and insert c_1 in A, else insert c_2, c' in A and c_1 in A'.

Phase 4: Return $E' = \{ra : a \in A - \{r\}\} \cup \{r'a' : a' \in A' - \{r'\}\} \cup \{rr'\}$.

Next we establish that any pair u, v of vertices can be connected in the augmented graph $H = (V, E \cup E')$ by two vertex-disjoint paths of length $\leq D$.

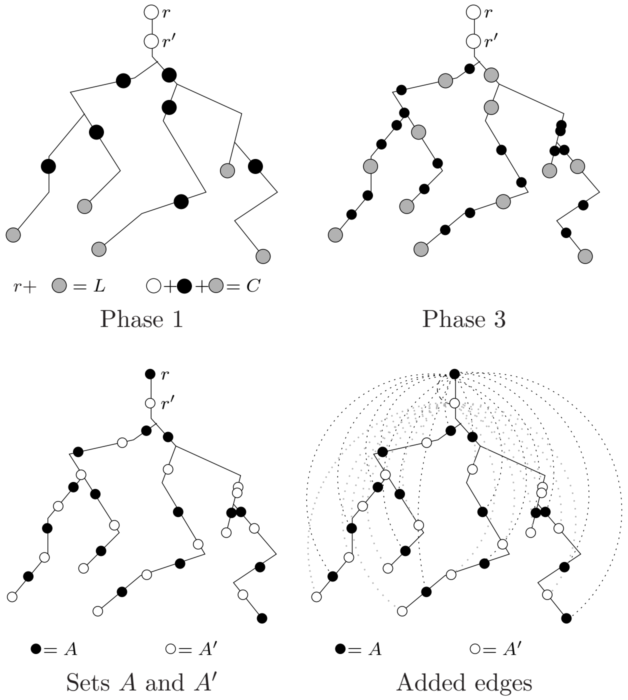


Figure 2: Illustration of Algorithm A2VDBP(F,D)

Lemma 4.1 *For any vertex $x \in V$ there exists two vertices $a \in A$ and $a' \in A'$ such that $d(x,a) \leq R-2$, $d(x,a') \leq R-2$, $P(a,a') \cap C \subseteq \{a,a'\}$, and x belongs to the (possibly degenerated) path $P(a,a')$.*

Proof. The result trivially holds if $x \in L_0$, because $L_0 \subseteq A \cap A'$ and we can set $a := x =: a'$. If $x \in S$, then $x \in A$ and we can set $a := x$. Let c' be the closest to x vertex of C. According to the algorithm, we can set $a' := c'$ if $d(x,c') \leq R-2$ and $a' := c_1$ if $R-1 \leq d(x,c') \leq 2R-4$. In the remaining case $d(c,c') \geq 2R-3$, the role of a' is played by the vertex c_2 described in the algorithm. If $x \in C-S$, then x is a direct descendant of some vertex $c \in C$, and, according to the algorithm, we can define $\{a,a'\} := \{x,c\}$ if $d(x,c) \leq R-2$ and $\{a,a'\} := \{x,c_1\}$ otherwise.

So, assume that $x \notin C \cup L_0$. Let c be the closest to x ancestor from C and let c' be the closest to x descendant from C. Clearly c' is a direct descendant of c. If $d(c,c') \leq R-2$, then we can set $\{a,a'\} := \{c,c'\}$ because c and c' belong to distinct sets A, A'. On the other hand, if $R-1 \leq d(c,c') \leq 2R-4$, then set $\{a,a'\} := \{c,c_1\}$ if $x \in P(c,c_1)$ and set $\{a,a'\} := \{c_1,c'\}$ if $x \in P(c_1,c')$. From the algorithm we infer that the vertices c, c' belong to one set A, A' and c_1 belongs to another set, establishing the assertion.

Finally suppose that $d(c,c') \geq 2R-3$, i.e., we are in the third case of Phase 3. Recall that the algorithm picks two vertices $c_1, c_2 \in P(c,c')$, c_1 at distance $R-2$ and c_2 at distance $2R-4$ to c' and considers a closest in C non-descendant c'' of c_2. The vertices c', c_2 belong to one set A, A', while c_1 and c_2 (as well as c'' and c_2) belong to distinct sets. Therefore we can assume that $x \in P(c_2,c)$, otherwise the proof is immediate by setting $\{a,a'\} = \{c',c_1\}$ if $x \in P(c',c_1)$ and $\{a,a'\} = \{c_1,c_2\}$ if $x \in P(c_1,c_2)$. Let z be the vertex of $P(x,c')$ at distance

R to c' ($z \in P(c_1, c_2)$ because $R \geq 5$, thus $z \neq x$). Denote by $\hat{c} \in C$ any closest to z center of an $(R-1)$-ball covering z. Since $d(z, \hat{c}) \leq R-1 < R = d(z, c')$, the choice of c' implies that $\hat{c}$ cannot be a descendant of z. For the same reason, $\hat{c}$ cannot be a descendant of x either. Hence $d(z, \hat{c}) = d(z, c_2) + d(c_2, \hat{c})$, $d(z, c'') = d(z, c_2) + d(c_2, c'')$, and $d(c_2, c'') \leq d(c_2, \hat{c})$ by the choice of c'', yielding $d(z, c'') \leq d(z, \hat{c}) \leq R-1$. The choice of $\hat{c}$ implies $d(z, c'') = d(z, \hat{c})$, therefore c'' can play the role of $\hat{c}$. In particular, this implies that c'' is not a descendant of x. Thus $R-1 \geq d(z, c'') = d(z, c_2) + d(c_2, x) + d(x, c'') \geq R-4+d(c_2, x)$ because $d(z, c_2) = R-4$, yielding $d(c_2, x) \leq 3 \leq R-2$. Notice also that $d(x, c'') < d(z, c'') \leq R-1$ because $x \in P(z, c'') - \{z\}$, i.e., $d(x, c'') \leq R-2$. Finally, we assert that $d(c, c'') < d(c, c')$. This is obviously true if $c'' \in P(c, x) \subset P(c, c')$ (in fact, $c'' = c$.) Otherwise, let y be the nearest common ancestor of c'' and x. Then $d(y, c'') \leq R-3$ while $d(y, c') = d(y, x) + d(x, c_2) + d(c_2, c') \geq 1 + 2R - 4 = 2R-3 > R-3$. Thus $d(c, c'') < d(c, c')$, and at the moment when the algorithm analyzes the pair c, c', the vertex c'' is already affected to A or A'. Since c_2 and c'' belong to distinct sets A, A', we can set $\{a, a'\} := \{c'', c_2\}$. Indeed $d(x, c_2) \leq R-2, d(x, c'') \leq R-2$, and $x \in P(c_2, c'')$, thus completing the proof. $\square$

Lemma 4.2 *Any pair u, v of distinct vertices can be connected in the augmented graph $H = (V, E \cup E')$ by two vertex-disjoint paths P_1 and P_2 of length at most $D = 2R-1$.*

Proof. By Lemma 4.1, there exist four vertices $a, b \in A$ and $a', b' \in A'$ such that $\max\{d(a, u), d(a', u), d(b, v), d(b', v)\} \leq R-2$ and $u \in P(a, a'), v \in P(b, b')$. Moreover, $P(a, a') \cap C \subseteq \{a, a'\}$ and $P(b, b') \cap C \subseteq \{b, b'\}$. Then the vertices r and r' may occur in the paths $P(a, a')$ and $P(b, b')$ only as their end-vertices, namely, $r' \in P(a, a')$ implies that $r' = a'$, while $r \in P(a, a')$ implies that $r = a$ and $u \in \{r, r'\}$.

If $\{u, v\} = \{r, r'\}$, then P_1 and P_2 are the two parallel edges between r and r'. On the other hand, if $u \in \{r, r'\}$, $v \notin \{r, r'\}$, then as P_1 and P_2 we take the two paths between u and v in the simple cycle of length at most $2R-4+3 = 2R-1$ formed by the path $P(b, b')$ and the new edges br, rr', and $r'b'$ (if $r' \neq b'$). So, assume that $r, r' \notin \{u, v\}$. If the paths $P(a, a')$ and $P(b, b')$ are disjoint, then as P_1 we take the path formed by $P(u, a), P(b, v)$, and the new edges ar, rb, while as P_2 we take the path formed by $P(u, a'), P(b', v)$, and the new edges $a'r'$ (if $r' \neq a'$), $r'b'$ (if $r' \neq b'$). These paths have length at most $2R-4+2 = 2R-2 < D$. They are disjoint because they may intersect only in r and r' and it is easy to see that $r \notin P_2$ and $r' \notin P_1$. Now, consider the case when $P(a, a') \cap P(b, b') \neq \emptyset$. Then the length of the path $P(u, v)$ is at most D because $d(u, t) + d(t, v) \leq R-2+R-2 < D$ for any vertex $t \in P(a, a') \cap P(b, b')$. Set $P_1 := P(u, v)$. It remains to specify the second path P_2. First suppose that $r' \in P(u, v)$. Then $a' = r' = b'$ because r' may appear only as an end-vertex on the paths $P(a, a')$ and $P(b, b')$. Therefore $(P(u, a) \cup P(v, b)) \cap P(u, v) = \{u, v\}$. Since $r \notin P_1$, we take as P_2 the path of H consisting of $P(u, a)$, the new edges ar and rb, and the path $P(b, v)$. Its length is at most $R-2+2+R-2 = 2R-2 < D$. Hence, let $r' \notin P(u, v)$. If $P(u, v) \cap P(u, a) = \{u\}$, then set $Q_1 := P(u, a), \alpha_1 := a, \beta_1 := r$. Otherwise, we have $P(u, v) \cap P(u, a') = \{u\}$, and set $Q_1 := P(u, a'), \alpha_1 := a', \beta_1 := r'$. In both cases, $Q_1 \cap P(u, v) = \{u\}$ and the length of Q_1 is at most $R-2$. Analogously, if $P(u, v) \cap P(v, b) = \{v\}$, then set $Q_2 := P(v, b), \alpha_2 := b, \beta_2 := r$ otherwise set $Q_2 := P(v, b'), \alpha_2 := b', \beta_2 := r'$. Again, in both cases $Q_2 \cap P(u, v) = \{v\}$ and the length of Q_2 is at most $R-2$. The paths Q_1 and Q_2 are disjoint, because all vertices of $Q_1 \cap Q_2$ would lie on the unique path of F connecting u and v and we know that $P(u, v)$ intersects $Q_1 \cup Q_2$ only in the vertices u, v. We take as P_2 the path of H consisting of Q_1, the edges

$\alpha_1\beta_1, \beta_1\beta_2$ (if $\beta_1 \neq \beta_2$), $\alpha_2\beta_2$, and the path Q_2. Its length is at most $2(R-2)+3 = 2R-1 = D$ and $P_1 \cap P_2 = \{u, v\}$. This establishes that indeed H is a feasible solution to the problem A2VDBP. $\square$

Lemma 4.3 $|A| + |A'| \leq 3|C| - 4$.

Proof. In Phase 3, for vertex $c \in (C - L) \cup S$ and each its direct descendant c' in C, the algorithm insert in $A \cup A'$ at most two new vertices (if $c = r$ and $c' = r'$, then no new vertex is added). Any vertex $c' \in C - S$ either belongs to L_0 or is a direct descendant of a unique vertex of $(C - L) \cup S$. Hence the number of new vertices is at most $2(|C| - |S| - |L_0|) - 2$. Since the vertices of L_0 are included in both sets A and A' and the remaining vertices of C in only one such set, we conclude that $|A| + |A'| \leq 2(|C| - |S| - |L_0|) - 2 + 2|L_0| + (|C| - |L_0|) = 3|C| - 2|S| - |L_0| - 2 \leq 3|C| - 4$. $\square$

Hence the number of edges added by the algorithm is $|A_1| + |A_2| + 1 \leq 3|C| - 3$. From Proposition 3.1 and Lemma 4.2 we obtain the following result:

Theorem 4.4 *Algorithm A2VDBP is a factor 6 approximation algorithm for the problems A2VDBP, A2EDBP, and ADCE on forests $F = (V, E)$ for any $D \geq 9$.*

5 A factor 4 approximation algorithm for ADCE

In this section, we modify (and simplify) the algorithm A2VDBP in order to return feasible solutions of smaller size for the problem ADCE (we assume $D = 2R - 1$). We use the same notations and conventions as in the algorithm A2VDBP, except that we do not need the vertex r' and instead of sets A, A' we will use one multiset A (A contains two copies of each vertex from $L_0 \cup S - \{r\}$). Namely, given a minimum vertex $(R-1)$-dominating set C of F containing the set of leaves L, we proceed each rooted tree of F and complete C to a set A with the property that for every vertex x there exist two vertices $a, a' \in A$ such that $x \in P(a, a')$ and $d(x, a) \leq R - 2, d(x, a') \leq R - 1$. The algorithm returns the augmentation $E' = \{ar : a \in A\}$. Every element of A gives rise to an added edge, therefore r will be connected with every vertex of $L_0 \cup (S - \{r\})$ by two parallel edges.

Algorithm ADCE(F,D)
Input: A forest $F = (V, E)$ and an odd integer $D = 2R - 1 \geq 5$.
Output: A set E' of new edges, such that $\text{diam}(H - e) \leq D$ for any $e \in E \cup E'$ where $H = (V, E \cup E')$ and $|E'| \leq 4 \cdot OPT_{ADCE}(F)$.
Phase 0: Root every tree containing at least two vertices at some leaf. Denote by S the list of such roots and pick $r \in S$.
Phase 1: Compute a minimum vertex $(R-1)$-dominating set C with $L \subseteq C$. Set $A := (C - \{r\}) \cup (L_0 \cup (S - \{r\}))$.
Phase 2: For each vertex $c \in C - (L_0 \cup S)$, find the closest in C ancestor c' of c. If $d(c, c') \geq R$, then find the vertex $c_1 \in P(c, c')$ at distance $R - 1$ to c and insert c_1 in A.
Phase 3: Return $E' = \{ra : a \in A\}$.

We present several auxiliary results establishing the feasibility of the augmentation E'.

Lemma 5.1 *For any vertex $x \in V - \{r\}$, there exists two different elements a, a' of the multiset A such that $x \in P(a, a')$ and $d(x, a) \leq d(x, a') \leq R - 1$.*

Proof. First, let $x \in V - S$. The result straightforwardly follows from the algorithm if x is a vertex of C or if x has been inserted in A. So, let $x \notin A$, and let c be a closest to x vertex of C (clearly, $d(c, x) \leq R - 1$). If x is an ancestor of c, then c and one of the vertices c' or c_1 described in Phase 2 form the required pair $\{a, a'\}$. Now, assume that c is not a descendant of x. Suppose by way of contradiction that all descendants of x in A are located at distance $\geq R$ from x. Consider an arbitrary direct descendant a of x in A. Since $d(x, a) \geq R$, from the algorithm we infer that $a \in A - C$. Hence a is at distance $R - 1$ from some descendant $b \in C$, i.e. $d(x, b) \geq R + (R - 1) = 2R - 1$. From this we can deduce that x is at distance $\geq 2R - 1$ from all its descendants from C. Pick a descendant x' of x satisfying $d(x, x') = R - 1$. Since any descendant of x' in C is at distance $\geq (2R - 1) - (R - 1) = R$ from x' and $d(x, x') = R - 1$, the center $c'' \in C$ of any $(R-1)$-ball covering x' is a descendant of x but not a descendant of x'. Let x'' be the nearest common ancestor of x' and c''. Then $d(c'', x'') \leq R - 2$ and $d(x, x'') \leq R - 2$, yielding $d(x, c'') \leq 2R - 4 < 2R - 1$, contrary to the fact that x is at distance $\geq 2R - 1$ from any of its descendants from C. This contradiction completes the proof of the case $x \in V - S$. If $x \in S - \{r\} \cup L_0$, we can take as a and a' the two occurrences of x in A. $\square$

Lemma 5.2 *For any vertex $x \in V - \{r\}$, there exists $a \in A$ such that $d(x, a) \leq R - 2$.*

Proof. Let $x \notin L_0 \cup S$, otherwise we can set $a = x$ by definition of A. Suppose by way of contradiction that all vertices of A are at distance $\geq R - 1$ from x and pick a vertex $c' \in C \subseteq A$ at distance $R - 1$ to x. If c' is a descendant of x, then x will be inserted in A when c' will be analyzed in Phase 2 of the algorithm, a contradiction. So, all descendants of x from C are at distance $\geq R$ to x. Let x' be the son of x and let c'' be any closest to x' vertex of C. Since $d(x, c'') \geq R - 1$, c'' must be a descendant of x'. Thus $d(x', c'') = R - 1$ by what has been shown about x. As $P(c'', x) \cap C = \{c''\}$, the vertex x' will be inserted in A when c'' will be analyzed. Since $R \geq 3$, we get a contradiction with the choice of x. $\square$

Lemma 5.3 *For any edge $e \in E \cup E'$, we have $diam(H - e) \leq D$.*

Proof. Pick two arbitrary vertices $u, v \in V$. Combining Lemmata 5.1 and 5.2 we can conclude that there exist four elements $a, a', b, b' \in A$ such that $a \neq a'$, $b \neq b'$, $u \in P(a, a')$, $v \in P(b, b')$, $d(u, a) \leq R - 2, d(u, a') \leq R - 1$, and $d(v, b) \leq R - 2, d(v, b') \leq R - 1$. If $r \in \{u, v\}$ then obviously we have two disjoint paths of length at most R between u or v in H. Now, suppose that $r \notin \{u, v\}$. For any edge e, the distances in $H - e$ between u and r and between v and r are at most R. If one of these two distances is at most $R - 1$, we are done. Otherwise, we must have either $a = b$ and $e = ar \in E'$ or e belong to $P(u, a) \cap P(v, b)$. In the first case, e does not belong to the path $P(u, v)$ and the length of this path is at most D, because $d(u, v) \leq d(u, a) + d(v, b) \leq 2R - 4 \leq D$. Now, suppose that e belong to $P(u, a) \cap P(v, b)$. Then u and v belong to the same tree component T of F. If e does not belong to $P(u, v)$, then $d(u, v) \leq d(u, a) + d(v, b) \leq 2R - 4$. Otherwise, if $e \in P(u, v) \cap P(u, a) \cap P(v, b)$ then the pairs u, b and v, a lie in different connected components of $T - e$. Therefore $d(v, a) + d(u, b) \leq d(u, a) + d(v, b) \leq 2R - 4$ and the smallest of the distances $d(v, a)$ and $d(u, b)$, say the first, is at most $R-2$. Thus $d_{H-e}(u,v) \leq d_{H-e}(u,r)+d_{H-e}(r,a)+d_{H-e}(a,v) \leq R+1+R-2 = 2R-1$, yielding $\mathrm{diam}(H-e) \leq 2R-1 = D$ for every edge $e \in E \cup E'$. $\square$

The next lemma follows immediately from the analysis of the algorithm:

Lemma 5.4 $|A| \leq 2|C|$.

From Proposition 3.1 and previous lemmata, we obtain the following result (the algorithm for the case $D = 2R$ is similar, the correctness proof follows the same lines, however is technically more involved, and is not given here due to space limitations):

Theorem 5.5 *Algorithm ADCE is a factor 4 approximation algorithm for the problem ADCE on forests $F = (V,E)$ for any $D \geq 5$.*

The algorithms A2VDBP and ADCE can be implemented in $O(|V|log|V|)$ time. Phase 1 of both algorithms takes linear time, while Phase 2 of A2VDBP can be easily performed in $O(|V|log|V|)$ time. In order to find the vertices of type c_1 and c_2, one can use mergeable heaps (supporting each of the operations Insert, Minimum, Extract-Min, and Union in $O(log|V|)$ time), i.e. in overall $O(|V|log|V|)$ time. Finally, the vertices c'' occurring in third case of Phase 3 of A2VDBP can be computed in total linear time.

We do not have tight examples for our algorithms; in fact, we conjecture that their approximation performances are better than 6 and 4. For this, however, better lower bounds than those established in Proposition 3.1 are needed. The worst case example which we found is a forest consisting of n isolated vertices and $D = n/2$. Both algorithms return a solution consisting of $2n-2$ edges. The first algorithm add two parallel edges between r and r', and two edges xr, xr' for any vertex $x \neq r, r'$, while the second algorithm add two parallel edges between r and any vertex $x \neq r$. For all three problems, the optimal solution is the cycle C_n obtained by adding only n edges.

In most of the cases occurring in the proof of Lemma 5.3, the algorithm ADCE returns two edge-disjoint paths of length $\leq D$. Nevertheless, the solutions returned by this algorithm are not always feasible augmentations for the problem A2EDBP. This leads to the question of finding a factor 4 approximation algorithm for the problem A2EDBP.

References

[1] N. Alon, A. Gyarfas, M. Ruszinko, Decreasing the diameter of bounded degree graphs, *J. Graph Theory*, **35** (2000), 161–172.

[2] G. J. Chang, Labeling algorithms for domination problems in sun-free chordal graphs, *Discrete Appl. Math.*, **22** (1988/1989), 21–34.

[3] V. Chepoi, Y. Vaxès, Augmenting trees to meet biconnectivity and diameter constraints, *Algorithmica*, **33** (2002), 243–262.

[4] F. R. K. Chung, M. R. Garey, Diameter Bounds for Altered Graphs, *J. Graph Theory*, **8** (1984), 511–534.

[5] K. P. Eswaran, R. E. Tarjan, Augmentation problems, *SIAM J. Computing*, **5** (1976), 653–665.

[6] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* Freeman, San Francisco, CA, 1979.

[7] T. Ishii, S. Yamamoto, H. Nagamochi, Augmenting forests to meet odd diameter requirements, *International Symposium on Algorithms and Computation* (ISAAC'03), Lecture Notes in Computer Science, **2906** (2003), pp. 434–443.

[8] G. Kant, H. L. Bodlaender, Planar graph augmentation problems, *Workshop on Algorithms and Data Structures* (WADS'91), Lecture Notes in Computer Science **519** (1991), pp. 286–298.

[9] Ch.-L. Li, S.Th. McCormick, and D. Simchi–Levi, On the minimum-cardinality-bounded-diameter and the bounded-cardinality-minimum-diameter edge addition problems, *Operations Research Letters* **11** (1992), 303–308.

[10] A.A. Schoone, H.L. Bodlaender, and J. van Leeuwen, Diameter increase caused by edge deletion, *J. Graph Theory* **11** (1987), 409–427.

[11] V.V. Vazirani, *Approximation Algorithms*, Springer-Verlag, Berlin, 2001.

[12] D. B. West, *Introduction to Graph Theory,* Prentice Hall, London, 2001.