

## Mixed Covering of Trees and the Augmentation Problem with Odd Diameter Constraints<sup>1</sup>

Victor Chepoi,<sup>2</sup> Bertrand Estellon,<sup>2</sup> Karim Nouioua,<sup>2</sup> and Yann Vaxès<sup>2</sup>

**Abstract.** In this paper we present a polynomial time algorithm for solving the problem of partial covering of trees with  $n_1$  balls of radius  $R_1$  and  $n_2$  balls of radius  $R_2$  ( $R_1 < R_2$ ) to maximize the total number of covered vertices. The solutions provided by this algorithm in the particular case  $R_1 = R - 1$ ,  $R_2 = R$  can be used to obtain for any integer  $\delta > 0$  a factor  $(2 + 1/\delta)$  approximation algorithm for solving the following augmentation problem with odd diameter constraints  $D = 2R + 1$ : Given a tree  $T$ , add a minimum number of new edges such that the augmented graph has diameter  $\leq D$ . The previous approximation algorithm of Ishii, Yamamoto, and Nagamochi (2003) has factor 8.

**Key Words.** Partial covering, Diameter, Augmentation problem, Dynamical programming, Approximation algorithms.

**1. Introduction.** In this paper we present a polynomial time algorithm for solving the following covering problem on trees:

**PROBLEM PARTIAL MIXED COVERING.** Given a tree  $T = (V, E)$  with  $n$  vertices, the non-negative integers  $R_1, R_2$  ( $R_1 < R_2$ ) and  $n_1, n_2$ , locate  $n_1$  balls of radius  $R_1$  and  $n_2$  balls of radius  $R_2$  to maximize the total number of covered vertices.

This problem generalizes the MAXIMUM COVERAGE problem investigated by Megiddo et al. [14], in which, given a tree  $T$  and the integers  $R_0$  and  $n_0$ , one wished to locate  $n_0$  balls of radius  $R_0$  to maximize the total number of covered vertices. Unlike the  $R$ -DOMINATING problem on trees (asking for covering of a tree with a minimum number of balls of radius  $R$ ), which is easily solvable in linear time, the existence of polynomial time algorithms for PARTIAL MIXED COVERING and MAXIMUM COVERAGE problems is non-trivial. The difficulty resides in the fact that we have to decide which vertices should be covered, which vertices should be chosen as centers, and balls of which radius should be located at those centers.

In [14] the initial motivation for studying the MAXIMUM COVERAGE problem came from the problem of locating a given number of facilities in a transportation network to cover a maximum number of customers. The more general setting of the PARTIAL MIXED COVERING problem allows one to model situations in which two

---

<sup>1</sup> This research was supported in part by the CNRS MathSTIC project “Combinatoire et géométrie des polyèdres: convexité, géométrie discrète et intégralité”.

<sup>2</sup> Laboratoire d’Informatique Fondamentale de Marseille, Faculté des Sciences de Luminy, Université de la Méditerranée, F-13288 Marseille Cedex 9, France. {chepoi,estellon,nouioua,vaxes}@lif.univ-mrs.fr.

kinds of facilities are available, the difference between them being their range, i.e., the maximum distance between a facility and a customer supplied by it. Also, it turns out that the MAXIMUM COVERAGE and the PARTIAL MIXED COVERING problems are (polynomially solvable) special cases of the (NP-hard) general PARTIAL COVERING problem introduced by Kearns [12] and recently revisited in [9]. In the present paper we provide yet another motivation for studying PARTIAL MIXED COVERING by deriving from it an approximation algorithm for the augmentation problem with diameter constraints which we formulate below. Notice also that several related problems can be reduced to PARTIAL MIXED COVERING. For example, running the algorithm for PARTIAL MIXED COVERING for all feasible pairs  $(n'_1, n'_2)$ , we obtain a polynomial time algorithm for the following problem:

**PROBLEM MIXED COVERING.** Given a tree  $T = (V, E)$  with  $n$  vertices, a function  $f$  of two non-negative integer variables, the non-negative integers  $R_1, R_2$  ( $R_1 < R_2$ ) and  $n_1, n_2$ , find a covering (if it exists) of  $T$  with  $n'_1 \leq n_1$  balls of radius  $R_1$  and  $n'_2 \leq n_2$  balls of radius  $R_2$  minimizing the function  $f(n'_1, n'_2)$ .

(If  $f(n'_1, n'_2) = n'_1$ , we obtain the problem of covering  $T$  with  $n_2$  balls of radius  $R_2$  and a minimum number of balls of radius  $R_1$ . In particular, if  $R_1 = 0$ , we get yet another formulation of the MAXIMUM COVERAGE problem.) The MIXED COVERING problem was first formulated in [3] in connection with the following augmentation problem:

**PROBLEM ADC (AUGMENTATION UNDER DIAMETER CONSTRAINTS).** Given a graph  $G = (V, E)$  with  $n$  vertices and a positive integer  $D$ , add a minimum number OPT of new edges  $E'$  such that the augmented graph  $G' = (V, E \cup E')$  has diameter at most  $D$ .

Due to its practical importance for improving the reliability of existing communication networks, the AUGMENTATION under DIAMETER CONSTRAINTS problem has received much attention in the literature [1], [3]–[7], [11], [13], [15]. In particular, it was shown to be NP-hard for any  $D \geq 2$  and at least as difficult to approximate as SET COVER [3], [13], [15]. However, the complexity status of this problem is unknown if the input graph  $G$  is a tree. In case of paths, OPT is determined up to an additive constant error term. Namely, in this case, Chung and Garey [5] established that  $(n - D - 1)/(D + 1) \leq \text{OPT} \leq (n - D + 2)/(D - 2)$ , and Alon et al. [1] refined this bound by establishing that the values of OPT for the  $n$ -cycle (i.e., a path plus one additional edge) satisfy  $\lfloor n/(D - 1) \rfloor - 7 \leq \text{OPT} \leq \lfloor n/(D - 1) \rfloor$  for even  $D$  and  $\lfloor n/(D - 2) \rfloor - 146 \leq \text{OPT} \leq \lfloor n/(D - 2) \rfloor$  for odd  $D$ . Other lower and upper bounds for more general classes of graphs have been considered in [1] and [7]; see also the survey of Chung [4] which contains further references and related problems.

For the problem ADC on trees, Chepoi and Vaxès [3] presented a factor 2 approximation algorithm for even  $D = 2R$  and Ishii et al. [11] presented a factor 8 approximation algorithm for odd  $D = 2R + 1$ . In [3] it was conjectured that the optimal solutions provided by MIXED COVERING may be used to derive approximate feasible solutions

for problem ADC with odd  $D$ . In this paper we prove that indeed any mixed covering of the input tree  $T$  with  $n_1$  balls of radius  $R - 1$  and  $n_2$  balls of radius  $R$  minimizing the function  $f(n_1, n_2) = n_1 + n_2(n_2 - 1)/2$  can be transformed into a feasible solution for problem ADC with  $D = 2R + 1$  containing at most  $(2 + 1/\delta)\text{OPT} + O(\delta^5)$  added edges for any integer  $\delta > 0$ , thus asymptotically matching the approximation ratio for even  $D$ . This augmentation (using at most  $n_1 + n_2(n_2 - 1)/2$  new edges) is obtained by drawing an edge between any pair of centers of  $n_2$  balls of radius  $R$  and between the center of any of  $n_1$  balls of radius  $R - 1$  and the center of some ball of radius  $R$ . Notice that the performance guarantees of all mentioned algorithms for trees 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 [5] is already quite involved.

The paper is organized as follows. In Section 2 we present a few necessary definitions and notations. Section 3 describes a dynamic programming algorithm for solving the PARTIAL MIXED COVERING problem. It also discusses the complexity and the correctness of the algorithm and presents some further problems which are solvable by a similar approach. Finally, in Section 4 we establish and analyze a factor  $2 + 1/\delta$  approximation algorithm for problem ADC with odd diameter constraints.

**2. Preliminaries.** For a graph  $G = (V, E)$ , the *length* of a path between two vertices is the number of edges in this path. The *distance*  $d_G(u, v)$  between two vertices  $u, v$  of  $G$  is the length of the shortest path between these vertices. The *diameter* of  $G$  is the largest distance between two vertices of  $G$ . For an integer  $k \geq 0$  and a vertex  $v \in V$ , let  $B_k(v) = \{x \in V : d_G(v, x) \leq k\}$  denote the *ball* of radius  $k$  centered at  $v$ . Set also  $B_{-1}(v) = \emptyset$ . The *relative radius*  $rr(x, B_k(v))$  of a ball  $B_k(v)$  in a vertex  $x$  of a tree  $T = (V, E)$  equals  $k - d_T(v, x)$ . We say that a ball  $B_k(v)$  is *located* in a subtree  $T'$  of  $T$  if  $v \in T'$ . For two vertices  $x, y$  of a tree  $T$  denote by  $P(x, y)$  the unique path of  $T$  between  $x$  and  $y$ . For a subset  $Q \subset V$  denote by  $T(Q)$  the least subtree of  $T$  containing  $Q$ . For a vertex  $y$  in a rooted tree  $T$  with root  $u$ , any vertex  $x \neq y$  on the path  $P(u, y)$  is called an *ancestor* of  $y$ . If  $x$  is an ancestor of  $y$ , then  $y$  is a *descendant* of  $x$ . Denote by  $T_x$  the subtree of  $T$  rooted at the vertex  $x$  and consisting of  $x$  and all of its descendants.

Define the  $k$ th *power* of a tree  $T = (V, E)$  as the graph  $T^k$  having  $V$  as vertex-set and two vertices  $x, y$  are adjacent in  $T^k$  if and only if  $d_T(x, y) \leq k$ . For a subset of vertices  $Q$ , denote by  $T^k(Q)$  the subgraph of  $T^k$  induced by  $Q$ . It is well known [2] that the  $k$ th power  $T^k$  of a tree is a chordal graph (whence  $T^k(Q)$  are chordal graphs for all  $Q \subseteq V$ ). Therefore  $T^k$  and  $T^k(Q)$  are perfect graphs for all  $k$  and  $Q$  (recall that a graph  $G$  is *perfect* [2] if the minimum number of cliques necessary to cover  $G$  equals the size of the largest stable set of  $G$ ). Notice that  $Y \subseteq V$  is a stable set of  $T^k$  if and only if  $d_T(x, y) > k$  for any  $x, y \in Y$ . On the other hand, a clique  $C$  of  $T^k$  consists of vertices with pairwise distances (in  $T$ ) at most  $k$ . The least subtree  $T(C)$  containing the set  $C$  has diameter at most  $k$ , thus its radius is either at most  $R$  if  $k = 2R$  or at most  $R + 1$  if  $k = 2R + 1$ . In the first case,  $T(C)$  can be covered by a ball of radius  $R$ . In the second case,  $T(C)$  can be covered by an *edge-ball* of radius  $R$ , i.e., by two balls of

radius  $R$  centered at adjacent vertices of  $T(C)$ . As a consequence, a covering of  $T^{2R}$  or of  $T^{2R+1}$  with a minimum number of cliques corresponds to a covering of  $T$  with a minimum number of balls of radius  $R$  or of edge-balls of radius  $R$ , and, due to the perfectness of the graphs  $T^{2R}$  and  $T^{2R+1}$ , this equals the size of a largest stable set of these graphs.

A polynomial time algorithm is called an  $\alpha$ -factor approximation algorithm for a minimization problem  $\Pi$  if for each instance  $I$  of  $\Pi$ , it returns a solution whose value is at most  $\alpha$  times the optimal value  $\text{OPT}_\Pi(I)$  of  $\Pi$  on  $I$  plus a constant not depending of  $I$ ; see [16].

**3. Mixed Covering of Trees.** In this section we describe a dynamic programming algorithm for solving the PARTIAL MIXED COVERING problem on trees. Our algorithm follows the main lines of the algorithm from [14] and works in general in the following way. Root the tree  $T$  at an arbitrary vertex  $u$ . The algorithm proceeds along the tree  $T$  in an upward manner, from leaves to the root, by solving larger and larger subproblems of the following type. Given the current vertex  $s$ , and the integers  $0 \leq n'_1 \leq n_1$ ,  $0 \leq n'_2 \leq n_2$ , the algorithm finds the *maximal* number of covered vertices of  $T_s$  in a partial covering using  $n'_1$  balls of radius  $R_1$  and  $n'_2$  balls of radius  $R_2$  located in  $T_s$ . However, the algorithm must take care of two things: (i) some ball which will be located outside  $T_s$  at some later stage and whose radius and center are yet unknown may have an impact on the covering of  $T_s$ , and (ii) we have to consider the interaction between the subtrees rooted at the neighbors of  $s$ , because some vertices of one or several such subtrees may be covered by a ball located in another subtree. To overcome these difficulties, we introduce two additional parameters  $r$  and  $a$  which take integer values in the ranges  $[-1, R_2 - 1]$  and  $[0, R_2]$ , respectively. For fixed values of  $r$  and  $a$ , the algorithm returns the maximal number of covered vertices of  $T_s$  in a partial covering using  $n'_1$  balls of radius  $R_1$  and  $n'_2$  balls of radius  $R_2$  located in  $T_s$  (permanent balls), given that one additional (temporary) ball of radius  $r$  is located at  $s$  and that the relative radius of at least one of the permanent balls located in  $T_s$  is at least  $a$ . This requires the solution of a resource allocation problem, which optimally distributes the balls of radius  $R_1$  and the balls of radius  $R_2$  among the subtrees rooted at the neighbors of  $s$  in  $T_s$ , using for this the optimal solutions of the previously solved subproblems at each of the sons of  $s$ . Following [14], we introduce the following functions EXT and INT:

1.  $\text{EXT}(T_s; n'_1, n'_2; r)$  is equal to the maximum number of vertices of  $T_s$  which can be covered by  $n'_1$  balls of radius  $R_1$  and  $n'_2$  balls of radius  $R_2$  located in  $T_s$ , given that there is one additional ball of radius  $r$  centered at  $s$ . The algorithm computes  $\text{EXT}(T_s; n'_1, n'_2; r)$  for all  $r \in \{-1, 0, \dots, R_2 - 1\}$ ,  $n'_1 \in \{0, \dots, n_1\}$ ,  $n'_2 \in \{0, \dots, n_2\}$ , and all rooted subtrees  $T_s, s \in V$ , of  $T$ .
2.  $\text{INT}(T_s; n'_1, n'_2; a)$  is equal to the maximum number of vertices of  $T_s$  which can be covered by  $n'_1$  balls of radius  $R_1$  and  $n'_2$  balls of radius  $R_2$  located in  $T_s$ , given that the relative radius  $rr(s, B)$  in  $s$  of one of those balls  $B$  is at least  $a$ . The algorithm computes  $\text{INT}(T_s; n'_1, n'_2; a)$  for all  $a \in \{0, \dots, R_2\}$ ,  $n'_1 \in \{0, \dots, n_1\}$ ,  $n'_2 \in \{0, \dots, n_2\}$ , and all rooted subtrees  $T_s, s \in V$ , of  $T$ .

Note that as  $\text{INT}(T_s; n'_1, n'_2; a)$  is defined only for  $a \geq 0$ , any family of balls attaining  $\text{INT}(T_s; n'_1, n'_2; a)$  always covers the vertex  $s$ , unless  $\text{INT}(T_s; n'_1, n'_2; a) = -\infty$ .

Let  $s_1, \dots, s_l$  be the sons of the current vertex  $s$ . To evaluate the functions INT and EXT on the subtree  $T_s$ , we use the values of these two functions on the subtrees  $T_{s_1}, \dots, T_{s_l}$ . If a temporary ball is located at  $s$ , then the algorithm computes the optimal distribution of remaining balls in the subtrees  $T_{s_1}, \dots, T_{s_l}$ , taking into account the ball centered at  $s$ . If no ball is located at  $s$ , the algorithm distributes the balls to  $T_{s_1}, \dots, T_{s_l}$  to maximize the number of covered vertices. In this case certain vertices of some subtree  $T_{s_i}$  can be covered by a ball located outside  $T_{s_i}$  and, vice versa, a ball centered at a vertex of  $T_{s_i}$  may cover a vertex outside this subtree. The first case is settled by the function EXT which locates a ball of radius  $r$  at  $s$ , the second case is settled by the function INT which forces the location in  $T_{s_i}$  of a ball which covers all vertices of  $T_s$  at distance at most  $a$  from  $s$ . Finally, in order to distribute the permanent balls optimally among the subtrees  $T_{s_1}, \dots, T_{s_l}$ , the algorithm uses the function ALLOC which solves the resource allocation problem with two resources [10]. Namely, given functions  $f_1, \dots, f_l$  of two integer variables,  $\text{ALLOC}(f_1, \dots, f_l; p, q)$  denotes the value of the optimal solution of

$$\begin{aligned} & \text{Maximize} && \sum_{i=1}^l f_i(p_i, q_i) \\ & \text{subject to} && \sum_{i=1}^l p_i = p, \\ & && \sum_{i=1}^l q_i = q, \\ & && p_i, q_i \text{ are non-negative integers.} \end{aligned}$$

The optimal distribution  $\{(p_1, q_1), \dots, (p_l, q_l)\}$  of the resource allocation problem is computed by the function  $\text{BUILD-ALLOC}(f_1, \dots, f_l; p, q)$ .

3.1. *The Algorithm.* Now we present the routines for computing INT and EXT in more details. If  $s$  is a leaf, then  $T_s = \{s\}$  and the valuation of INT and EXT is given by the following formulae:

$$\begin{aligned} (1) \quad \text{INT}(T_s; n'_1, n'_2; a) &= \begin{cases} -\infty & \text{if } n'_1 = n'_2 = 0 \text{ or if } n'_2 = 0 \text{ and } a > R_1, \\ 1 & \text{otherwise;} \end{cases} \\ (2) \quad \text{EXT}(T_s; n'_1, n'_2; r) &= \begin{cases} 0 & \text{if } n'_1 = n'_2 = 0 \text{ and } r = -1, \\ 1 & \text{otherwise.} \end{cases} \end{aligned}$$

Suppose now that  $s$  has at least one descendant. To evaluate the functions INT and EXT on the subtree  $T_s$ , the algorithm uses the values of the functions INT and EXT on the subtrees  $T_{s_1}, \dots, T_{s_l}$ . Namely, the algorithm returns  $\text{INT}(T_s; n'_1, n'_2; a) = \max\{I_1, I_2, I_3\}$  if  $a \leq R_1$ ,  $\text{INT}(T_s; n'_1, n'_2; a) = \max\{I_2, I_3\}$  if  $a > R_1$ , and  $\text{EXT}(T_s; n'_1, n'_2; r) =$

$\max\{E_1, E_2\}$ , where  $I_1, I_2, I_3, E_1$ , and  $E_2$  are defined in the following way:

$$\begin{aligned}
I_1 &= \text{ALLOC}(f_1, \dots, f_l; n'_1 - 1, n'_2) + 1, \\
&\quad \text{where } f_i(p, q) = \text{EXT}(T_{s_i}; p, q; R_1 - 1); \\
I_2 &= \text{ALLOC}(g_1, \dots, g_l; n'_1, n'_2 - 1) + 1, \\
&\quad \text{where } g_i(p, q) = \text{EXT}(T_{s_i}; p, q; R_2 - 1); \\
I_3 &= \max\{\text{ALLOC}(h_1^{j a'}, \dots, h_l^{j a'}; n'_1, n'_2) + 1; j \in \{1, 2, \dots, l\}, a' \in \\
&\quad \{a, a + 1, \dots, R_2\}\}, \quad \text{where } h_i^{j a'}(p, q) = \text{EXT}(T_{s_i}; p, q; a' - 1) \\
&\quad \text{for } i \neq j \text{ and } h_j^{j a'}(p, q) = \text{INT}(T_{s_j}; p, q; a' + 1); \\
E_1 &= \text{INT}(T_s; n'_1, n'_2; r + 1); \\
E_2 &= \text{ALLOC}(f'_1, \dots, f'_l; n'_1, n'_2) + \delta(r), \quad \text{where } \delta(r) = 1 \text{ if } r \geq 0 \text{ and} \\
&\quad \delta(r) = 0 \text{ if } r = -1, \text{ and } f'_i(p, q) = \text{EXT}(T_{s_i}; p, q; \max\{-1, r - 1\}).
\end{aligned}$$

If the INT entry equals  $I_1$  (and  $I_1 > I_3$ ), then a permanent ball  $B'$  of radius  $R_1$  is centered at  $s$  and the remaining  $n'_1 - 1$   $R_1$ -balls and  $n'_2$   $R$ -balls are optimally distributed among the subtrees  $T_{s_1}, \dots, T_{s_l}$ . Notice that in this case  $a \leq R_1$  and that the relative radius in  $s$  of all permanent balls located in  $T_s - \{s\}$  is less than  $R_1$ , otherwise the ball  $B'$  is useless, yielding  $I_1 \leq I_3$ . Therefore, there are no interactions among subtrees, from which we infer that the problems associated with the subtrees are independent, i.e., a covered vertex of  $T_{s_i}$  is either covered by  $B'$  or by a ball located in  $T_{s_i}$ . This explains why in order to compute  $I_1$  we make a call of ALLOC with parameters  $f_i(p, q) = \text{EXT}(T_{s_i}; p, q; R_1 - 1)$ ,  $i = 1, \dots, l$ . The analysis of  $I_2$  is similar. Now suppose that the INT entry equals  $I_3$ . In this case the partial covering of  $T_s$  is done by permanent balls located in  $T_s - \{s\}$ . Let  $T_{s_j}$  be the subtree which hosts a permanent ball  $B'$  maximizing  $rr(s, B') =: a'$ . By definition of INT, we must have  $a' \geq a$ . Notice that every covered vertex of some subtree  $T_{s_i}$  is necessarily covered either by the ball  $B'$  or by a permanent ball located in  $T_{s_i}$ . Since we do not know a priori neither the subtree  $T_{s_j}$  providing the maximum nor  $a'$ , we should test all possibilities. Now, for given  $a' \in \{a, a + 1, \dots, R_2 - 1\}$  and  $j \in \{1, \dots, l\}$ , since the ball  $B'$  with relative radius  $a'$  is located in  $T_{s_j}$ , in order to distribute the permanent balls among the subtrees  $T_{s_1}, \dots, T_{s_l}$  we make a call of ALLOC with parameters  $h_i^{j a'}(p, q) = \text{EXT}(T_{s_i}; p, q; a' - 1)$  for each  $i = 1, \dots, l$ ,  $i \neq j$ , and  $h_j^{j a'}(p, q) = \text{INT}(T_{s_j}; p, q; a' + 1)$ .

Analogously, if the EXT entry equals  $E_1$ , then there exists a permanent ball  $B'$  located in  $T_s$  such that  $rr(s, B') > r$ . As a consequence, the ball of radius  $r$  centered at  $s$  is useless, whence we can use the result for INT, thus explaining why  $E_1 = \text{INT}(T_s; n'_1, n'_2; r + 1)$ . Finally, if  $E_2 > E_1$ , then we search for an optimal distribution of  $n'_1 + n'_2$  permanent balls in  $T_s$  such that  $rr(s, B) \leq r$  holds for each permanent ball  $B$ . Then all covered vertices of any subtree  $T_{s_i}$  are necessarily covered either by a permanent ball located in  $T_{s_i}$  or by the ball of radius  $r$  centered at  $s$ . Therefore, the problems associated with the subtrees are independent and we make a call of ALLOC with parameters  $f'_i(p, q) = \text{EXT}(T_{s_i}; p, q; \max\{-1, r - 1\})$ . Notice that if  $r = -1$ , then the vertex  $s$  is not covered by a permanent ball nor by the (empty) ball of radius  $-1$  centered at  $s$ . In all other cases,  $s$  is covered by the ball of radius  $r \geq 0$  centered at  $s$ . We conclude this subsection with a formal description of the algorithm.



To restore an optimal partial covering, the algorithm keeps in the tables S-INT and S-EXT the parameters of the distributions yielding the optimal value for the functions INT and EXT, and perhaps the radius of the permanent ball centered in the current vertex (if the optimal solution requires its location). The total space for these tables is equal to  $n_1 n_2 R_2 O(\sum_{v \in V} \deg(s)) = O(n_1 n_2 R_2 n)$ . Using the tables S-INT and S-EXT, an optimal location is computed by recursive functions BUILD-INT and BUILD-EXT in a downward manner. Each of these functions takes as input a vertex  $s$  and a list of parameters identifying a respective INT- or EXT-problem for  $s$ , and, using the information stored in the tables S-INT and S-EXT, decides if a permanent ball (and of what radius) should be centered at  $s$ , and specifies the parameters for its recursive call at each son  $s_i$  of  $s$ . After processing the root  $u$  of the tree  $T$ , the algorithm returns BUILD-EXT( $T_u, n_1, n_2, -1$ ).

**BUILD-INT**( $T_s, n'_1, n'_2, a$ )

**if**  $s$  is a leaf  
**then return** The set of  $n'_1$  balls of radius  $R_1$  and  $n'_2$  balls of radius  $R_2$  centered in  $s$   
**else**  $(A, a^*, j^*) \leftarrow$  S-INT( $T_s; n'_1, n'_2; a$ )  
 Let  $\{(p_i, q_i): i = 1, \dots, l\}$  be the allocation  $A$ .  
**if**  $j^* = \infty$   
**then**  $\mathcal{B} \leftarrow \{B(s, a^*)\}$   
**for**  $i \leftarrow 0$  **to**  $l$  **do**  $\mathcal{B} \leftarrow \mathcal{B} \cup$  BUILD-EXT( $T_{s_i}, p_i, q_i, a^* - 1$ )  
**else**  $\mathcal{B} \leftarrow \emptyset$   
**for**  $j \leftarrow 0$  **to**  $l$   
**do if**  $j = j^*$  **then**  $\mathcal{B} \leftarrow \mathcal{B} \cup$  BUILD-INT( $T_{s_j}, p_j, q_j, a^* + 1$ )  
**else**  $\mathcal{B} \leftarrow \mathcal{B} \cup$  BUILD-EXT( $T_{s_j}, p_j, q_j, a^* - 1$ )  
**return**  $\mathcal{B}$

**BUILD-EXT**( $T_s, n'_1, n'_2, r$ )

**if**  $s$  is a leaf  
**then return** The set of  $n'_1$  balls of radius  $R_1$  and  $n'_2$  balls of radius  $R_2$  centered in  $s$   
**else**  $(A, c) \leftarrow$  S-EXT( $T_s; n'_1, n'_2; r$ )  
 Let  $\{(p_i, q_i): i = 1, \dots, l\}$  be the allocation  $A$ .  
**if**  $c = 1$  **then**  $\mathcal{B} \leftarrow$  BUILD-INT( $T_s, n'_1, n'_2, r + 1$ )  
**else** (\*  $c = 2$  \*)  
 $\mathcal{B} \leftarrow \emptyset$   
**for**  $j \leftarrow 0$  **to**  $l$   
**do**  $\mathcal{B} \leftarrow \mathcal{B} \cup$  BUILD-EXT( $T_{s_j}, p_j, q_j, \max\{-1, r - 1\}$ )  
**return**  $\mathcal{B}$

3.2. *Correctness and Complexity.* In this subsection, we establish the correctness and the complexity of the algorithm presented in Section 3.1.

**THEOREM 3.1.** *The described algorithm correctly solves the PARTIAL MIXED COVERING problem in  $O(n_1^3 n_2^3 R_2^2 n^2)$  time.*

PROOF. To prove the correctness of the algorithm, it suffices to show that all values of INT and EXT are correctly computed. This is shown by the following claims.

CLAIM 1. *If INT and EXT are correctly evaluated on each of the subtrees  $T_{s_1}, \dots, T_{s_l}$ , then INT is correctly evaluated on  $T_s$ .*

PROOF. Let  $n'_1 \in \{0, \dots, n_1\}$ ,  $n'_2 \in \{0, \dots, n_2\}$ , and  $a \in \{0, \dots, R_2\}$ . Consider an optimal partial covering  $\mathcal{C}^*$  of  $T_s$  with at most  $n'_1$  balls of radius  $R_1$  and at most  $n'_2$  balls of radius  $R_2$  located in  $T_s$ , given that the relative radius in  $s$  of some ball of  $\mathcal{C}^*$  is at least  $a$ . Suppose additionally that all balls  $B$  of  $\mathcal{C}^*$  are necessary, i.e.,  $\mathcal{C}^*$  minus  $B$  is no longer an optimal covering for the parameters  $n'_1, n'_2$ , and  $a$ . Let  $B'$  be a ball of  $\mathcal{C}^*$  with a maximal relative radius in  $s$  and set  $a' = rr(s, B')$ . Notice that if  $\mathcal{C}^*$  contains a ball  $B$  centered at  $s$ , then necessarily  $B' = B$ . Indeed, since  $B$  cannot be removed from  $\mathcal{C}^*$  without violating the optimality of this partial covering, the relative radius in  $s$  of any ball of  $\mathcal{C}^*$  different from  $B$  is less than the radius of  $B$ . Since  $rr(s, B)$  equals the radius of  $B$ , we conclude that  $B' = B$ .

First suppose that  $B'$  is centered at  $s$ . From what has been shown above, we deduce that  $rr(s, B) < a'$  for any ball  $B \in \mathcal{C}^*$  different from  $B'$ . Thus every vertex of  $T_{s_i}$ ,  $i = 1, \dots, l$ , which is covered by  $\mathcal{C}^*$  is covered either by a ball located in  $T_{s_i}$  or by  $B'$ . Since EXT is correctly evaluated on each of the subtrees  $T_{s_i}$ ,  $i \in \{1, \dots, l\}$ , the number of vertices of  $T_{s_i}$  covered by  $\mathcal{C}^*$  is at most  $\text{EXT}(T_{s_i}; p_i, q_i; a' - 1)$ . Since the function ALLOC optimally distributes the remaining  $n'_1 + n'_2 - 1$  balls, the value of  $I_1$  (if  $B'$  has radius  $R_1$ ) or of  $I_2$  (if  $B'$  has radius  $R_2$ ) is at least  $|T_s \cap (\bigcup \{B \in \mathcal{C}^*\})|$ .

Now suppose that no ball of  $\mathcal{C}^*$  is centered at  $s$  and assume that the ball  $B'$  is located in the subtree  $T_{s_j}$ . Then  $rr(s, B') = a' \geq a$  by definition of INT. For any  $i = 1, \dots, l$ , every vertex of  $T_{s_i}$  covered by  $\mathcal{C}^*$  is covered either by a ball centered at  $T_{s_i}$  or by  $B'$ . Hence the number of vertices of  $T_{s_i}$  covered by  $\mathcal{C}^*$  is at most  $\text{EXT}(T_{s_i}; p_i, q_i; a' - 1)$  for  $i \neq j$  (because  $rr(s_i, B') = a' - 1$ ) and the number of vertices of  $T_{s_j}$  covered by  $\mathcal{C}^*$  is at most  $\text{INT}(T_{s_j}; p_j, q_j; a' + 1)$  (because  $rr(s_j, B') = a' + 1$ ). From the optimality of ALLOC we infer that  $I_3$  is at least  $|T_s \cap (\bigcup \{B \in \mathcal{C}^*\})|$ .  $\square$

CLAIM 2. *If INT and EXT are correctly evaluated on each of the subtrees  $T_{s_1}, \dots, T_{s_l}$  and INT is correctly evaluated on  $T_s$ , then EXT is correctly evaluated on  $T_s$ .*

PROOF. Let  $n'_1 \in \{0, \dots, n_1\}$ ,  $n'_2 \in \{0, \dots, n_2\}$ , and  $r \in \{-1, \dots, R_2 - 1\}$ . Consider an optimal partial cover  $\mathcal{C}^*$  of  $T_s$  with  $n'_1$  balls of radius  $R_1$ ,  $n'_2$  balls of radius  $R_2$  located in  $T_s$ , and an additional ball  $B'$  of radius  $r$  centered at  $s$ . Suppose additionally that all balls of  $\mathcal{C}^*$  are necessary.

First suppose that there is a ball  $B \in \mathcal{C}^*$  different from  $B'$  obeying  $rr(s, B) > r$ . Then no covered vertex of  $T_s$  is covered solely by  $B'$ , therefore  $B'$  is useless. As a consequence, we conclude that the number of vertices covered by  $\mathcal{C}^*$  equals  $\text{INT}(T_s; n'_1, n'_2; r + 1)$ .

So, suppose that  $rr(s, B) \leq r$  for every ball  $B$  of  $\mathcal{C}^*$  different from  $B'$ . Then every vertex covered by  $\mathcal{C}^*$  of each subtree  $T_{s_i}$ ,  $i = 1, \dots, l$ , is covered either by a ball located in  $T_{s_i}$  or by  $B'$ . If one denote by  $p_i$  and  $q_i$  the number of balls of radius  $R_1$  and  $R_2$  of  $\mathcal{C}^*$  located in the subtree  $T_{s_i}$ , then the number of vertices of  $T_{s_i}$  covered by  $\mathcal{C}^*$  is at most

$\text{EXT}(T_i; p_i, q_i; \max\{-1, r - 1\})$ . From the optimality of ALLOC we deduce that  $E_2$  is at least  $|T_s \cap (\bigcup\{B \in \mathcal{C}^*\})|$ , establishing the result.  $\square$

To find the complexity of the algorithm, we estimate the number of operations necessary to evaluate the functions INT and EXT in some vertex  $s$ . One computation of INT requires one call of ALLOC to evaluate  $I_1$  and  $I_2$  and at most  $R_2 \deg(s)$  to evaluate  $I_3$ . Analogously, one computation of EXT requires one call of the function ALLOC. There is a dynamic programming algorithm with complexity  $O(lp^2q^2)$  for solving the resource allocation problem  $\text{ALLOC}(f_1, \dots, f_i; p, q)$  with two resources (see, for example, pages 207–208 of [10]). Therefore, if INT and EXT have been evaluated on the subtrees  $T_{s_1}, \dots, T_{s_t}$ , then the  $2n_1n_2(R_2 + 1)$  values of INT and EXT for  $T_s$  can be found in  $O(n_1^3n_2^3R_2^2\deg^2(s))$  time. Since  $\sum_{s \in V} \deg(s) = 2n - 2$ , the total complexity of the algorithm is  $O(n_1^3n_2^3R_2^2n^2)$ .  $\square$

**3.3. Related Problems.** In this subsection we present several further problems, related to PARTIAL MIXED COVERING, which can be solved by adapting the algorithm described above.

(i) A natural generalization of the PARTIAL MIXED COVERING problem is that of maximizing the number of covered vertices by  $n_1$  balls of radius  $R_1$ ,  $n_2$  balls of radius  $R_2$ ,  $\dots$ ,  $n_k$  balls of radius  $R_k$ , where  $R_1 < R_2 < \dots < R_k$ . We call the resulting problem GENERALIZED PARTIAL MIXED COVERING. Our algorithm can be modified to solve it in  $O(R_k^2n^2\Pi_{i=1}^kn_i^3)$  time (which is less than  $O(n^{3k+4})$ ). For this, in the computation of INT we distinguish  $k + 1$  cases: one for each kind of permanent ball centered at  $s$  plus one dealing with the case when no permanent ball is centered at  $s$ . Then the INT entry equals the maximum of these  $k + 1$  values  $I_1, \dots, I_k, I_{k+1}$ . Additionally, instead of solving a resource allocation problem with two resources each time, we solve via dynamic programming a resource allocation problem with  $k$  resources (requiring  $O(\deg(s)\Pi_{i=1}^kn_i^2)$  time per instance).

(ii) Problem (i) can be further generalized in the following way: given the integers  $0 \leq R_1 < \dots < R_k$  and the positive integers  $n_2, \dots, n_k$ , locate  $n_i$  balls of radius  $R_i$ ,  $i = 2, \dots, k$ , so that the remaining part of the tree can be covered with a minimum number  $n_1$  of balls of radius  $R_1$ . To solve this problem we can solve a sequence of GENERALIZED PARTIAL MIXED COVERING problems, one for each value of  $n_1$  varying between 0 and  $n$ , and return the smallest value of  $n_1$  for which the whole tree is covered. However, we can solve this problem more efficiently by modifying the algorithm presented above. For this, we modify the definition of the functions INT and EXT, and, instead of taking maxima and performing maximization in the resource allocation problem, we take minima and solve a minimization resource allocation problem. For example,  $\text{EXT}(T_s; n'_2, \dots, n'_k; r)$  is defined to be the minimum number of balls of radius  $R_1$  which, together with  $n'_2$  balls of radius  $R_2$ ,  $\dots$ ,  $n'_k$  balls of radius  $R_k$ , and an additional ball of radius  $r$  centered at  $s$ , cover the subtree  $T_s$  ( $\text{EXT}(T_s; n'_2, \dots, n'_k; a)$  is defined accordingly). The complexity of this algorithm is  $O(R_k^2n^2\Pi_{i=2}^kn_i^3)$ .

In the particular case  $k = 2$ , we obtain the problem of covering a tree  $T$  with  $n_2$  balls of radius  $R_2$  and a minimum number of balls of radius  $R_1$ , which can be solved in  $O(n_2^3n^2R_2^2)$  time. Now, if we want to solve the MIXED COVERING problem with  $R_2 = R \leq 2$ ,  $R_1 = R - 1 > 0$ , and  $f(n_1, n_2) = n_1 + n_2(n_2 - 1)/2$ , then we may

suppose that  $n_2 \leq \sqrt{n}$ . Since the dynamic programming algorithm keeps the solutions of subproblems for all vertices (in particular for  $u$ ) and all  $n_2 \leq \sqrt{n}$ , it suffices to select the solution minimizing  $f$ . The algorithm in this case has complexity  $O(n^{3.5}R_2^2)$ .

(iii) Another natural generalization of the partial mixed covering problem is the following partial mixed list covering problem. Namely, additionally to the input data of (i), for each vertex  $s$  of the tree  $T$ , a list  $L_s \subseteq \{1, \dots, k\}$  is given, which defines what kinds of permanent balls are allowed to be centered at  $s$  (the vertices  $s$  with  $L_s = \emptyset$  are forbidden for establishing permanent balls). Then each INT entry is again the maximum of  $I_1, \dots, I_k, I_{k+1}$  defined as in (i), but, in this case,  $I_j = -\infty$  for any  $j \notin L_s$ .

(iv) Several weighted versions of the PARTIAL MIXED COVERING problem and its variations (i)–(iii) can be solved using the same approach. For instance, the tree  $T = (V, E)$  can be endowed with a length function  $l: E \rightarrow \mathbb{R}_+$ . In this case the balls will be defined with respect to the distance induced by this length function. Also, we may want to consider that covering some vertex  $u$  induces a gain  $\pi_u$  and we wish to find a partial mixed covering maximizing the total gain of covered vertices. In this case, each time the current vertex  $s$  is covered, we add  $\pi_s$  (instead of 1) to the value of INT or EXT.

(v) The GENERALIZED PARTIAL MIXED COVERING problem in which the balls of certain radii are replaced by edge-balls can be easily formulated in the form of (iii) or (iv). For this, we subdivide every edge  $e$  of the tree  $T = (V, E)$  by introducing a new vertex  $s_e$ , thus obtaining a new tree  $T'$ . Then  $d_{T'}(u, v) = 2d_T(u, v)$  for any two vertices  $u, v$  of  $T$ . Instead of  $n_i$  balls of a radius  $R_i$  we consider  $n_i$  balls of radius  $2R_i$  of  $T'$  and require that they can be centered only at the vertices of  $T$ . Instead of  $n_i$  edge-balls of a radius  $R_i$  we consider  $n_i$  balls of radius  $2R_i + 1$  of  $T'$  which can be centered only at the new vertices of  $T'$ . Define the gain of covering any old vertex to be 1 and the gain of covering any new vertex to be 0. Then solving the resulting partial mixed covering problem on the tree  $T'$  is equivalent to solving the initial problem with balls and edge-balls on the tree  $T$ .

**4. The Augmentation Problem with Odd Diameter Constraints.** In this section we apply MIXED COVERING to derive a factor  $2 + 1/\delta$  (for any integer  $\delta > 0$ ) approximation algorithm for the augmentation problem with odd diameter constraints  $D = 2R + 1$  on trees. For this, we compute in  $O(n^{3.5}R^2)$  time a mixed covering of  $T$  with  $n'_1$  balls of radius  $R - 1$  and  $n'_2$  balls of radius  $R$  which minimizes the function  $f(n'_1, n'_2) = n'_1 + n'_2(n'_2 - 1)/2$ . The augmentation algorithm returns the set  $F$  of new edges running between the centers of any pair of balls of radius  $R$  and between the center of any ball of radius  $R - 1$  and the center of some ball of radius  $R$ . Set  $\text{ALG} := n'_1 + n'_2(n'_2 - 1)/2$  and let  $\text{OPT}$  denote the number of edges of an optimal solution of problem ADC for  $T$ .

**THEOREM 4.1.** *For any integer  $\delta > 0$ , we have  $\text{ALG} \leq (2 + 1/\delta)\text{OPT} + O(\delta^5)$ .*

**PROOF.** First we show that the augmented graph  $H = (V, E \cup F)$  has diameter at most  $2R + 1$ . Pick two arbitrary vertices  $u, v \in V$ . Suppose that in the mixed covering,  $u$

belongs to a ball centered at the vertex  $p$  and  $v$  belongs to a ball centered at the vertex  $q$ . If both these balls have radius  $R$ , then  $p$  and  $q$  are connected by a new edge, therefore  $d_H(u, v) \leq R + 1 + R = 2R + 1$ . If one ball has radius  $R$  and another one has radius  $R - 1$ , then  $d_H(p, q) \leq 2$ , whence  $d_H(u, v) \leq (R - 1) + 2 + R = 2R + 1$ . Finally, if both balls centered at  $p$  and  $q$  have radius  $R - 1$ , then  $d_H(p, q) \leq 3$  according to the algorithm, therefore  $d_H(u, v) \leq (R - 1) + 3 + (R - 1) = 2R + 1$ . This shows that  $F$  is a feasible augmentation of  $T$ .

Let  $E'$  be an optimal solution for ADC and let  $G' = (V, E \cup E')$  be the augmented graph. Denote by  $C_1$  the set of end-vertices of edges from  $E'$  and by  $\mathcal{B}_1$  the set of balls of radius  $R - 1$  centered at vertices of  $C_1$ . Set  $n_1 := |C_1|$ . Let  $Q := V - \bigcup\{B_{R-1}(c) : c \in C_1\}$ . Consider the graph  $T^{2R+1}(Q)$ . As we noticed in Section 2,  $T^{2R+1}(Q)$  is a perfect graph, therefore the size of a maximum stable set  $Y \subseteq Q$  of  $T^{2R+1}(Q)$  equals the minimum number  $n_2$  of edge-balls of radius  $R$  covering the set  $Q$ . Denote by  $\mathcal{B}_2$  the family of edge-balls in this covering. The *cluster* of a vertex  $x \in Q$  is the set  $C_x = \{c \in C_1 : d_T(x, c) = R\}$ . All vertices of  $C_1 \setminus C_x$  are at distance  $> R$  from  $x$  because  $x \in Q$ , therefore, if the cluster  $C_x$  is empty, then  $x$  must be at distance  $\leq 2R + 1$  in  $T$  from all vertices of  $Q$ . Notice also that the clusters  $C_x, C_y$  of two vertices  $x, y$  of  $Q$  are disjoint provided  $d_T(x, y) \geq 2R + 1$ .

**CLAIM 1.** *If two vertices  $x, y \in Q$  are not adjacent in  $T^{2R+1}(Q)$ , then there exists at least one added edge running between the clusters  $C_x$  and  $C_y$ . In particular,  $\text{OPT} = |E'| \geq n_2(n_2 - 1)/2$ .*

**PROOF.** Consider a path  $P$  of length  $\leq 2R + 1$  connecting the vertices  $x$  and  $y$  in the augmented graph  $G'$ . Since  $d_T(x, y) > 2R + 1$ , this path will necessarily use one or several new edges. Denote by  $x'$  and  $y'$  the vertices of  $P \cap C_1$  closest to  $x$  and  $y$ , respectively. Since  $x, y \in Q$ ,  $d_T(x, x') \geq R$ ,  $d_T(y, y') \geq R$ , and  $C_x \cap C_y = \emptyset$ , and we conclude that  $x' \in C_x, y' \in C_y$ , and  $x'$  and  $y'$  must be connected by an edge of  $E'$ .

Since any two vertices  $x, y$  of  $Y$  are not adjacent in  $T^{2R+1}(Q)$ , there exists at least one new edge connecting the clusters  $C_x$  and  $C_y$ . Since the clusters of vertices of  $Y$  are pairwise disjoint and  $|Y| = n_2$ , these new edges are pairwise distinct and therefore there exist at least  $n_2(n_2 - 1)/2$  such edges.  $\square$

**CLAIM 2.**  $\text{OPT} \geq n_1/2$ .

**PROOF.** By definition,  $n_1 = |C_1|$ , where  $C_1$  is the set of end-vertices of edges of  $E'$ . Obviously  $|E'| \geq |C_1|/2$ , the worst case occurring when  $E'$  is a perfect matching on  $C_1$ .  $\square$

Next we refine the lower bounds for OPT provided by Claims 1 and 2. Also from the covering  $\mathcal{B}_1 \cup \mathcal{B}_2$  of  $T$  with  $n_1$  balls of radius  $R - 1$  and  $n_2$  edge-balls of radius  $R$ , we will derive a feasible solution for the mixed covering problem, thus establishing an upper bound on the number ALG of edges added by the algorithm. Let  $n_0$  be the minimum number of balls of radius  $R$  of  $T$  covering the set  $Q$  and let  $\mathcal{B}_0$  denote the set of balls

from this covering. Since every edge-ball of radius  $R$  can be covered by two balls of radius  $R$ , we obtain the following inequality:

CLAIM 3.  $n_0 \leq 2n_2$ .

For a fixed integer  $\Delta \geq \delta + 2$ , we perform the following operations with the balls of the collection  $\mathcal{B}_0$ . Initialize  $\mathcal{A} := \emptyset$ , and  $\mathcal{C} := \mathcal{B}_0$ . Test each ball  $B$  of the current collection  $\mathcal{C}$ , and if the set  $(B \cap Q) \setminus ((\bigcup\{B' \in \mathcal{C}: B' \neq B\}) \cup (\bigcup\{B' \in \mathcal{A}\}))$  can be covered by at most  $\Delta$  balls of radius  $R - 1$ , then add these balls to  $\mathcal{A}$  and remove  $B$  from  $\mathcal{C}$ . Repeat this operation until no ball of the current collection  $\mathcal{C}$  can be further replaced by  $\Delta$  balls of radius  $R - 1$ . Set  $\alpha := |\mathcal{A}|$  and  $\gamma := |\mathcal{C}|$ . Notice that  $\mathcal{B}_1 \cup \mathcal{A}$  and  $\mathcal{C}$  constitute a mixed covering of the tree  $T$  with  $n_1 + \alpha$  balls of radius  $R - 1$  and  $\gamma$  balls of radius  $R$ .

CLAIM 4. *Every ball  $B = B_R(v)$  of the resulting collection  $\mathcal{C}$  contains  $\Delta$  vertices*

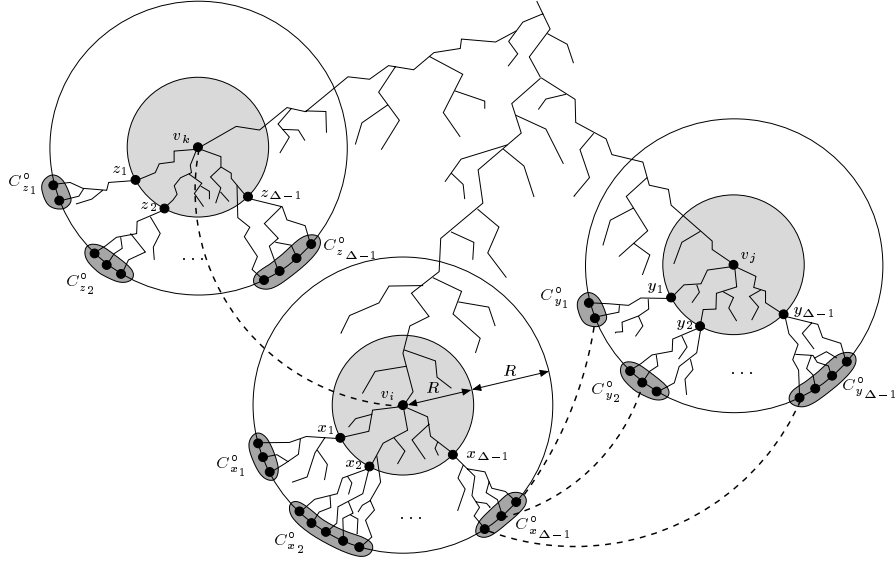
$$x_1, \dots, x_\Delta \in B^+ := (B \cap Q) \setminus ((\bigcup\{B' \in \mathcal{C}: B' \neq B\}) \cup (\bigcup\{B' \in \mathcal{A}\}))$$

*such that  $d_T(x_i, v) = R$  for all  $i = 1, \dots, \Delta$  and  $d_T(x_i, x_j) = 2R$  for all distinct  $i, j \in \{1, \dots, \Delta\}$ .*

PROOF. Since the ball  $B$  survived the last test, the minimum number of balls of radius  $R - 1$  necessary to cover the set  $B^+$  is at least  $\Delta + 1$ . Due to the duality between packing and covering with balls in trees (applied to the perfect graph  $T^{2R-2}(B^+)$ ),  $B^+$  contains  $\Delta + 1$  vertices  $S = \{x_1, \dots, x_\Delta, x_{\Delta+1}\}$  forming a stable set of  $T^{2R-2}(B^+)$ . Since  $2R - 1 \leq d_T(x_i, x_j) \leq d_T(x_i, v) + d_T(v, x_j) \leq R + R = 2R$ , we conclude that at least  $\Delta$  vertices, say  $x_1, \dots, x_\Delta$ , must be located at distance  $R$  from the center  $v$  of  $B$ , moreover, the paths connecting  $v$  with these vertices pairwise intersect only in  $v$ , otherwise we will find two vertices of  $S$  having distance  $\leq 2R - 2$ . Thus  $d_T(x_i, x_j) = 2R$  for arbitrary  $i, j \in \{1, \dots, \Delta\}$ ,  $i \neq j$ .  $\square$

Suppose that the balls of the collection  $\mathcal{C}$  are ordered  $B_1, \dots, B_\gamma$  in the following way: root the tree  $T(Q)$  at the center of an arbitrary ball of  $\mathcal{C}$  and for two balls  $B, B' \in \mathcal{C}$  set  $B = B_i, B' = B_j$ , where  $i < j$ , provided the center of the ball  $B$  is closer or at the same distance to the root than the center of the ball  $B'$ , breaking ties arbitrarily. Denote by  $v_i$  the center of the ball  $B_i$ ,  $i = 1, \dots, \gamma$ . Consider the set of  $\Delta$  vertices  $x_1, \dots, x_\Delta$  of the ball  $B_i$  described in Claim 4. Since the paths  $P(v_i, x_1), \dots, P(v_i, x_\Delta)$  pairwise intersect solely in the vertex  $v_i$ , at most one such path, say  $P(v_i, x_\Delta)$ , may pass via the father of  $v_i$ . Therefore the set  $F_i := \{x_1, \dots, x_{\Delta-1}\}$  consists solely of descendants of the vertex  $v_i$ . On the other hand, if  $y \in F_j, j < i$ , then  $d_T(v_i, y) > R = d_T(v_j, y)$ , whence  $y$  cannot be a descendant of  $v_i$ . For each vertex  $x \in F_i$ , set  $C_x^\circ := C_x - \{v_i\}$ . For an illustration of this and other notions introduced above, see Figure 1.

CLAIM 5. *For all  $i, j \in \{1, \dots, \gamma\}$ ,  $j < i$ , if  $x \in F_i$  and  $y \in F_j$ , then  $d_T(x, y) \geq 2R + 1$ . If  $d_T(x, y) = 2R + 1$ , then  $v_j$  is the closest to  $v_i$  center of a ball of  $\mathcal{C}$  which is an ancestor of  $v_i$ .*



**Fig. 1.** Illustration of some notions used in the proof of Theorem 4.1.

**PROOF.** Let  $z$  be the nearest common ancestor of  $v_i$  and  $v_j$ . If  $z \notin \{v_i, v_j\}$ , then  $z$  is also the nearest common ancestor of  $x$  and  $y$ , because  $x$  is a descendant of  $v_i$  and  $y$  is a descendant of  $v_j$ . Since  $x \in F_i$  and  $y \in F_j$ ,  $d(x, v_i) = R$  and  $d(y, v_j) = R$  hold, and we deduce that

$$d_T(x, y) = d_T(x, v_i) + d_T(v_i, z) + d(z, v_j) + d_T(v_j, y) \geq R + 1 + 1 + R > 2R + 1.$$

On the other hand, if  $z \in \{v_i, v_j\}$ , then  $z = v_j$ , because  $j < i$ . Since  $y$  is not a descendant of  $v_i$ , we conclude that  $v_i \in P(x, y)$ , and

$$d_T(x, y) = d_T(x, v_i) + d_T(v_i, y) \geq R + (R + 1) = 2R + 1.$$

Hence, if  $d_T(x, y) = 2R + 1$ , then  $v_i$  is a descendant of  $v_j$  and it remains to show that no other center  $v_k$  of a ball of  $\mathcal{C}$  can be located on the path between  $v_i$  and  $v_j$ . Suppose the contrary: then, since  $y$  is not a descendant of  $v_k$ , the vertices  $v_i$  and  $v_k$  belong to the path  $P(x, y)$ , hence

$$d_T(x, v_i) + d_T(v_i, v_k) + d_T(v_k, y) \geq R + 1 + (R + 1) = 2R + 2,$$

yielding a contradiction.  $\square$

For each  $i = 2, \dots, \gamma$ , denote by  $v_{i'}$  (if it exists) the closest to  $v_i$  center of a ball of  $\mathcal{C}$  which is an ancestor of  $v_i$ . The following assertion is an immediate consequence of Claim 5.

CLAIM 6. For any  $i = 2, \dots, \gamma$ , if  $j, k \in \{1, \dots, i-1\} - \{i'\}$ ,  $j \neq k$ , and  $x \in F_i$ ,  $y \in F_j$ , and  $z \in F_k$ , then  $d_T(x, y) > 2R + 1$ ,  $d_T(x, z) > 2R + 1$ , and  $d_T(y, z) \geq 2R + 1$ . In particular, the clusters  $C_x$ ,  $C_y$ , and  $C_z$  are pairwise disjoint.

We apply this claim to provide new lower bounds for OPT. For  $i = 2, \dots, \gamma$ , set  $\Gamma_i = \bigcup_{j < i} \bigcup_{y \in F_j} C_y$ . Let  $\beta_i$  be the number of edges of the optimal solution  $E'$  running between the vertex  $v_i$  and the clusters from  $\Gamma_i$ . For a vertex  $x \in F_i$ , denote by  $\kappa(x)$  the number of edges of  $E'$  running between the cluster  $C_x$  and the clusters from  $\Gamma_i$ . Let  $\kappa_i = \min\{\kappa(x) : x \in F_i\}$ . Notice that  $\kappa_i \geq \beta_i$ , because  $v_i$  belongs to all clusters  $C_x$ ,  $x \in F_i$ . On the other hand,  $\kappa_i \geq i - 2$ , because there is an added edge between  $C_x$  and every cluster  $C_y$  such that  $d_T(x, y) > 2R + 1$  (see Figure 1 for an illustration), and, by Claim 6,  $\Gamma_i$  contains at least  $i - 2$  pairwise disjoint clusters  $C_y$ , obeying  $d_T(x, y) > 2R + 1$ .

CLAIM 7.  $OPT \geq (\Delta - 1) \sum_{i=2}^{\gamma} \kappa_i - (\Delta - 2) \sum_{i=2}^{\gamma} \beta_i$ .

PROOF. For each  $i = 2, \dots, \gamma$ , there are  $\beta_i$  edges of  $E'$  between  $v_i$  and vertices of  $\Gamma_i$ , therefore, for any vertex  $x \in F_i$  at least  $\kappa_i - \beta_i$  edges of  $E'$  run between  $C_x^\circ$  and  $\Gamma_i$ . Since the clusters of the  $\Delta - 1$  vertices from  $F_i$  pairwise intersect solely in  $v_i$ , at least  $(\Delta - 1)(\kappa_i - \beta_i)$  distinct edges of  $E'$  run between  $\bigcup_{x \in F_i} C_x^\circ$  and  $\Gamma_i$ . Moreover, since the clusters of any two vertices  $y \in F_j$  and  $z \in F_k$ ,  $j \neq k$ , are disjoint, we obtain

$$OPT \geq \sum_{i=2}^{\gamma} [\beta_i + (\Delta - 1)(\kappa_i - \beta_i)] = (\Delta - 1) \sum_{i=2}^{\gamma} \kappa_i - (\Delta - 2) \sum_{i=2}^{\gamma} \beta_i. \quad \square$$

CLAIM 8.  $OPT \geq (n_1 - \beta + \sum_{i=2}^{\gamma} \beta_i)/2$ , where  $\beta$  is the number of vertices  $v_i$ ,  $i = 2, \dots, \gamma$ , such that  $\beta_i > 0$ .

PROOF. First notice that for any vertex  $v_i$  with  $\beta_i > 0$ , the  $\beta_i$  edges of  $E'$  incident to  $v_i$  form a star with  $\beta_i + 1$  vertices. Each of the  $n_1 - \sum_{i=2}^{\gamma} \beta_i - \beta$  remaining vertices of  $C_1$  is incident to an added edge, yielding at least  $(n_1 - \sum_{i=2}^{\gamma} \beta_i - \beta)/2$  other edges of  $E'$ , the worst case being a perfect matching. This shows that

$$OPT \geq \sum_{i=2}^{\gamma} \beta_i + \frac{n_1 - \sum_{i=2}^{\gamma} \beta_i - \beta}{2} = \frac{n_1 + \sum_{i=2}^{\gamma} \beta_i - \beta}{2}. \quad \square$$

CLAIM 9.  $ALG \leq n_1 - \beta + \Delta(n_0 - \gamma) + \gamma(\gamma - 1)/2$ .

PROOF. Notice that every  $v_i$  with  $\beta_i > 0$  is a center of a ball of radius  $R$  of  $\mathcal{C}$  and a center of a ball of radius  $R - 1$  of  $\mathcal{B}_1$ . Remove those  $\beta$  balls from  $\mathcal{B}_1$ . Together with  $\mathcal{A}$  and  $\mathcal{C}$ , the resulting collection  $\mathcal{B}_1$  form a mixed covering of  $T$  with  $\gamma$  balls of radius  $R$  and at most  $n_1 - \beta + \Delta(n_0 - \gamma)$  balls of radius  $R - 1$ . This covering gives rise to a feasible solution of the augmentation problem using at most  $n_1 - \beta + \Delta(n_0 - \gamma) + \gamma(\gamma - 1)/2$  new edges.  $\square$

First assume that  $\gamma \geq 2$ . From Claims 7 and 8 we obtain

$$\begin{aligned} \left(2 + \frac{1}{\delta}\right) \text{OPT} &\geq n_1 - \beta + \sum_{i=2}^{\gamma} \beta_i + \frac{\Delta-1}{\delta} \sum_{i=2}^{\gamma} \kappa_i - \frac{\Delta-2}{\delta} \sum_{i=2}^{\gamma} \beta_i \\ &= n_1 - \beta + \left(\frac{\Delta-2}{\delta} - 1\right) \sum_{i=2}^{\gamma} (\kappa_i - \beta_i) + \left(1 + \frac{1}{\delta}\right) \sum_{i=2}^{\gamma} \kappa_i \\ &\geq n_1 - \beta + \left(1 + \frac{1}{\delta}\right) \sum_{i=2}^{\gamma} \kappa_i, \end{aligned}$$

where the last inequality follows from  $\kappa_i \geq \beta_i$  for  $i = 2, \dots, \gamma$  and  $\Delta \geq \delta + 2$ . Since  $\kappa_i \geq i - 2$  for  $i = 2, \dots, \gamma$ , we conclude that

$$\left(2 + \frac{1}{\delta}\right) \text{OPT} \geq n_1 - \beta + \left(1 + \frac{1}{\delta}\right) \frac{(\gamma-1)(\gamma-2)}{2}.$$

In order to ensure  $(2 + 1/\delta)\text{OPT} \geq \text{ALG}$ , in view of last inequality and Claim 9 it suffices to show that

$$n_1 - \beta + \left(1 + \frac{1}{\delta}\right) \frac{(\gamma-1)(\gamma-2)}{2} \geq n_1 - \beta + \Delta(n_0 - \gamma) + \frac{\gamma(\gamma-1)}{2}.$$

Taking  $\Delta = \delta + 2$ , after some elementary transformations this inequality can be rewritten as  $f(\gamma) = \gamma^2 + b\gamma - c \geq 0$ , where  $b = 2\delta^2 + 2\delta - 3$  and  $c = (2\delta^2 + 4\delta)n_0 - 2\delta - 2$ . Since  $b > 0$  because  $\delta$  is a positive integer, the inequality  $f(\gamma) > 0$  holds if  $c \leq 0$ . Now, if  $c > 0$ , the inequality holds for any  $\gamma \geq \gamma_0$ , where  $\gamma_0$  is the largest solution of the quadratic equation  $f(\gamma) = 0$  (the exact value of  $\gamma_0$  will be given below). Therefore, if  $\gamma \geq \gamma_0$ , then  $\text{ALG} \leq (2 + 1/\delta)\text{OPT}$ .

Now, suppose that  $\gamma \leq \gamma_0$ . Using the lower bounds for  $\text{OPT}$  established in Claims 1 and 2, we obtain  $(2 + 1/\delta)\text{OPT} \geq n_1 + n_2^2/2\delta - n_2/2\delta$ . On the other hand, from Claim 9 we deduce that  $\text{ALG} \leq n_1 + n_0\Delta + \gamma^2/2$ . Since  $n_0 \leq 2n_2$  by Claim 3,  $\Delta = \delta + 2$ , and  $\gamma \leq \gamma_0$ , we have  $\text{ALG} \leq n_1 + 2(\delta + 2)n_2 + \gamma_0^2/2$ . Now, by definition of  $\gamma_0$ ,

$$\begin{aligned} \frac{\gamma_0^2}{2} &= \frac{[\sqrt{(2\delta^2 + 2\delta - 3)^2 + 8((\delta^2 + 2\delta)n_0 - \delta - 1)} - (2\delta^2 + 2\delta - 3)]^2}{8} \\ &< \frac{(2\delta^2 + 2\delta - 3)^2 + 8((\delta^2 + 2\delta)n_0 - \delta - 1) + (2\delta^2 + 2\delta - 3)^2}{8} \\ &< \delta^2(\delta + 1)^2 + 2\delta(\delta + 2)n_2 - \delta - 1, \end{aligned}$$

because  $(p - q)^2 \leq p^2 + q^2$  if  $p, q \geq 0$ ,  $2\delta^2 + 2\delta - 3 < 2\delta(\delta + 1)$ , and  $n_0 \leq 2n_2$ . Comparing the lower bound for  $(2 + 1/\delta)\text{OPT}$  with the upper bounds for  $\text{ALG}$  and  $\gamma_0^2/2$ , the desired inequality  $\text{ALG} \leq (2 + 1/\delta)\text{OPT}$  holds if

$$n_2^2 - n_2 \geq 2\delta^3(\delta + 1)^2 + 4\delta^2(\delta + 2)n_2 + 4\delta(\delta + 2)n_2 - 2\delta^2 - 2\delta,$$

i.e., provided

$$g(n_2) = n_2^2 - n_2(4\delta^3 + 12\delta^2 + 8\delta + 1) - (2\delta^5 + 4\delta^4 + 2\delta^3 - 2\delta^2 - 2\delta) \geq 0.$$

As a result, we conclude that  $\text{ALG} \leq (2 + 1/\delta)\text{OPT}$  holds for all  $n_2$  larger than or equal to the largest solution  $n_2^+$  of the quadratic equation  $g(n_2) = 0$ , otherwise, if  $n_2 < n_2^+$ , we have  $\text{ALG} - (2 + 1/\delta)\text{OPT} \leq -g(n_2)/2\delta \leq -g(2\delta^3 + 6\delta^2 + 4\delta + \frac{1}{2})/2\delta = O(\delta^5)$ . The case  $\gamma = 0$  or 1 can be settled in a similar way, using the inequalities  $(2 + 1/\delta)\text{OPT} \geq n_1 + n_2^2/2\delta - n_2/2\delta$  and  $\text{ALG} \leq n_1 + 2n_2(\delta + 2)$ . Therefore, in all cases we obtain that  $\text{ALG} \leq (2 + 1/\delta)\text{OPT} + O(\delta^5)$ , concluding the proof of the theorem.  $\square$

REMARK 1. In the particular cases  $\delta = 1, 2$ , and 3, we obtain the following values for the additive error between  $\text{ALG}$  and  $(2 + 1/\delta)\text{OPT}$ : 80, 621, 2560, respectively.

REMARK 2. Using the factor  $2 + 1/\delta$  augmentation algorithm for ADC analyzed in Theorem 4.1 and the biconnectivity augmentation algorithm of Eswaran and Tarjan [8], we obtain a factor  $3 + 1/\delta$  approximation algorithm for problem ADC on trees and odd diameters  $D$  with an additional requirement that the resulting augmented graph is biconnected (this problem is known to be NP-hard on trees [3]). A polynomial time algorithm with performance guarantee 4 is presented in [11]; see also [3] for related results.

REMARK 3. Notice that the optimal solutions of problem ADC with  $D = 2R + 1$  may contain cliques of arbitrary size  $k$ . For this, consider the tree  $T$  consisting of a star  $S$  in which the center  $v$  is adjacent to its tips  $v_1, \dots, v_k$ , plus  $k$  other pairwise disjoint stars  $S_1, \dots, S_k$ , where the star  $S_i$  consists of  $K \gg k$  paths of length  $R$  pairwise intersecting only in the central vertex  $v_i$ . The diameter of  $T$  is  $2R + 2$ , and, in order to decrease it to  $2R + 1$ , the best way is to add an edge between any pair of centers of the stars.

REMARK 4. In the case of  $D = 2R$ , the factor 2 approximation algorithm for problem ADC on trees given in [3] computes an optimal mixed covering with one ball of radius  $R$  and a minimum number of balls of radius  $R - 1$ , and draws an edge between the center of each  $(R - 1)$ -ball and the center of the  $R$ -ball. We conjecture that this algorithm for even  $D$  as well as the algorithm for odd  $D$  analyzed in this paper are actually optimal up to an additive constant error term.

**Acknowledgment.** We are grateful to the referee for several useful suggestions.

## References

- [1] N. Alon, A. Gyarfas, M. Ruzinko, Decreasing the diameter of bounded degree graphs, *J. Graph Theory*, **35** (2000), 161–172.
- [2] A. Brandstadt, V.B. Le, J.P. Spinrad, *Graph Classes*, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, PA, 1999.
- [3] V. Chepoi, Y. Vaxes, Augmenting trees to meet biconnectivity and diameter constraints, *Algorithmica*, **33** (2002), 243–262.
- [4] F.R.K. Chung, Diameters of graphs: old problems and new results, *Congr. Numer.*, **60** (1987), 295–317.
- [5] F.R.K. Chung, M.R. Garey, Diameter bounds for altered graphs, *J. Graph Theory*, **8** (1984), 511–534.

- [6] Y. Dodis, S. Khanna, Designing networks with bounded pairwise distance, *Proc. Annual ACM Symposium on the Theory of Computing*, 1999, pp. 750–759.
- [7] P. Erdős, A. Gyárfás, M. Ruzsinkó, How to decrease the diameter of triangle-free graphs, *Combinatorica*, **18** (1998), 493–501.
- [8] K.P. Eswaran, R.E. Tarjan, Augmentation problems, *SIAM J. Comput.*, **5** (1976), 653–665.
- [9] R. Gandhi, S. Khuller, A. Srinivasan, Approximation algorithms for partial covering problems, *J. Algorithms*, **53** (2004), 54–84.
- [10] T. Ibaraki, N. Katoh, *Resource Allocation Problems. Algorithmic Approaches*, MIT Press, Cambridge, MA, 1988.
- [11] T. Ishii, S. Yamamoto, H. Nagamochi, Augmenting forests to meet odd diameter requirements, *Proc. International Symposium on Algorithms and Computation (ISAAC '03)*, Lecture Notes in Computer Science, vol. 2906, Springer-Verlag, Berlin, 2003, pp. 434–443.
- [12] M. Kearns, *The Computational Complexity of Machine Learning*, MIT Press, Cambridge, MA, 1990.
- [13] Ch.-L. Li, S.Th. McCormick, D. Simchi-Levi, On the minimum-cardinality-bounded-diameter and the bounded-cardinality-minimum-diameter edge addition problems, *Oper. Res. Lett.*, **11** (1992), 303–308.
- [14] N. Megiddo, E. Zemel, S.L. Hakimi, The maximum coverage location problem, *SIAM J. Algebraic Discrete Methods*, **4** (1983), 253–261.
- [15] A.A. Schoone, H.L. Bodlaender, J. van Leeuwen, Diameter increase caused by edge deletion, *J. Graph Theory*, **11** (1987), 409–427.
- [16] V.V. Vazirani, *Approximation Algorithms*, Springer-Verlag, Berlin, 2001.