Chapter 4
Lower Bounds on Box Counting

Consider a box counting heuristic using radius-based boxes, e.g., *Maximum Excluded Mass Burning*. There is no guarantee that the computed $B_R(r)$ is minimal or even near minimal. However, if a lower bound on $B_R(r)$ is available, we can immediately determine the deviation from optimality for the calculated $B_R(r)$. A method that provides a lower bound $B_R^L(r)$ on $B_R(r)$ is presented in [44]. The lower bound is computed by formulating box counting as an *uncapacitated facility location problem* (**UFLP**), a classic combinatorial optimization problem. This formulation provides, via the dual of the linear programming relaxation of **UFLP**, a lower bound on $B_R(r)$. The method also yields an estimate of $B_R(r)$; this estimate is an upper bound on $B_R(r)$. Under the assumption that $B_R(r) = a(2r+1)^{-d_B}$ holds for some positive constant a and some range of r, a linear program [6], formulated using the upper and lower bounds on $B_R(r)$, provides an upper and lower bound on d_B. In the event that the linear program is infeasible, a quadratic program [18] can be used to estimate d_B.

4.1 Mathematical Formulation

Let the box radius r be fixed. For simplicity we will refer to node j rather than node n_j. Define $\mathbb{N} \equiv \{1, 2, \cdots, N\}$. Let C^r be the symmetric N by N matrix defined by

$$C^r_{ij} = \begin{cases} 0 & \text{if } dist(i, j) \le r, \\ \infty & \text{otherwise.} \end{cases}$$

(As with the matrix M^r_{ij} defined by (3.1), the superscript r in C^r_{ij} does *not* mean the r-th power of the matrix C.) For example, for $r = 1$, the matrix C^r corresponding to the network of Fig. 4.1 is

E. Rosenberg, *A Survey of Fractal Dimensions of Networks*, SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-319-90047-6_4

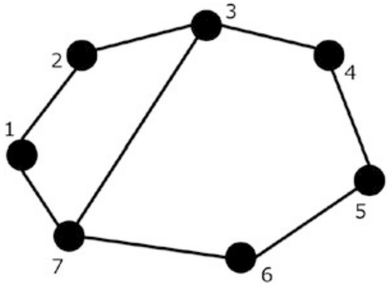

Fig. 4.1 Example network with seven nodes and eight arcs

$$C^1 = \begin{pmatrix} 0 & 0 & - & - & - & - & 0 \\ 0 & 0 & 0 & - & - & - & - \\ - & 0 & 0 & 0 & - & - & 0 \\ - & - & 0 & 0 & 0 & - & - \\ - & - & - & 0 & 0 & 0 & - \\ - & - & - & - & 0 & 0 & 0 \\ 0 & - & 0 & - & - & 0 & 0 \end{pmatrix},$$

where a dash "–" is used to indicate the value ∞.

For $j \in \mathbb{N}$, let

$$y_j = \begin{cases} 1 \text{ if the box centered at } j \text{ is used to cover } \mathbb{G}, \\ 0 \text{ otherwise.} \end{cases}$$

A given node i will, in general, be within distance r of more than one center node j used in the covering of $\mathbb{G}$. However, we will assign each node i to exactly one node j, and the variables x_{ij} specify this assignment. For $i, j \in \mathbb{N}$, let

$$x_{ij} = \begin{cases} 1 \text{ if } i \text{ is assigned to the box centered at } j, \\ 0 \text{ otherwise.} \end{cases}$$

With the understanding that r is fixed, for simplicity we write c_{ij} to denote element (i, j) of the matrix C^r. The minimal network covering problem is

$$\text{minimize} \quad \sum_{j=1}^{N} y_j + \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} x_{ij} \tag{4.1}$$

$$\text{subject to} \quad \sum_{j=1}^{N} x_{ij} = 1 \text{ for } i \in \mathbb{N} \tag{4.2}$$

$$x_{ij} \le y_j \text{ for } i, j \in \mathbb{N} \tag{4.3}$$

$$x_{ij} \geq 0 \text{ for } i, j \in \mathbb{N} \tag{4.4}$$

$$y_j = 0 \text{ or } 1 \text{ for } j \in \mathbb{N}. \tag{4.5}$$

Let **UFLP** denote the optimization problem defined by (4.1)–(4.5). Constraint (4.2) says that each node must be assigned to the box centered at some j. Constraint (4.3) says that node i can be assigned to the box centered at j only if that box is used in the covering, i.e., only if $y_j = 1$. The objective function is the sum of the number of boxes in the covering and the total cost of assigning each node to a box. Problem **UFLP** is feasible since we can always set $y_i = 1$ and $x_{ii} = 1$ for $i \in \mathbb{N}$; i.e., let each node be the center of a box in the covering. Given a set of binary values of y_j for $j \in \mathbb{N}$, since each c_{ij} is either 0 or ∞, if there is a feasible assignment of nodes to boxes then the objective function value is the number of boxes in the covering; if there is no feasible assignment for the given y_j values then the objective function value is ∞. Note that **UFLP** requires only $x_{ij} \geq 0$; it is not necessary to require x_{ij} to be binary. This relaxation is allowed since if (x, y) solves **UFLP** then the objective function value is not increased, and feasibility is maintained, if we assign each i to exactly one k (where k depends on i) such that $y_k = 1$ and $c_{ik} = 0$.

The *primal linear programming relaxation* **PLP** of **UFLP** is obtained by replacing the restriction that each y_j is binary with the constraint $y_j \geq 0$. We associate the dual variable u_i with the constraint $\sum_{j=1}^{N} x_{ij} = 1$, and the dual variable w_{ij} with the constraint $x_{ij} \geq 0$. The *dual linear program* [18] **DLP** corresponding to **PLP** is

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{N} u_i \\ \text{subject to} \quad & \sum_{i=1}^{N} w_{ij} \leq 1 \text{ for } j \in \mathbb{N} \\ & u_i - w_{ij} \leq c_{ij} \text{ for } i, j \in \mathbb{N} \\ & w_{ij} \geq 0 \text{ for } i, j \in \mathbb{N}. \end{aligned}$$

Following [11], we set $w_{ij} = \max\{0, u_i - c_{ij}\}$ and express **DLP** using only the u_i variables:

$$\text{maximize} \quad \sum_{i=1}^{N} u_i \tag{4.6}$$

$$\text{subject to} \quad \sum_{i=1}^{N} \max\{0, u_i - c_{ij}\} \leq 1 \text{ for } j \in \mathbb{N}. \tag{4.7}$$

Let $v(UFLP)$ be the optimal objective function value of **UFLP**. Then $B_R(r) = v(UFLP)$. Let $v(PLP)$ be the optimal objective function value of the linear programming relaxation **PLP**. Then $v(UFLP) \geq v(PLP)$. Let $v(DLP)$ be the optimal objective function value of the dual linear program **DLP**. By linear programming duality theory, $v(PLP) = v(DLP)$. Define $u \equiv (u_1, u_2, \cdots, u_N)$. If u is feasible for **DLP** as defined by (4.6) and (4.7), then the dual objective function $\sum_{i=1}^{N} u_i$ satisfies $\sum_{i=1}^{N} u_i \leq v(DLP)$. Combining these relations, we have

$$B_R(r) = v(UFLP) \geq v(PLP) = v(DLP) \geq \sum_{i=1}^{N} u_i \,.$$

Thus $\sum_{i=1}^{N} u_i$ is a lower bound on $B_R(r)$. As described in [44], to maximize this lower bound subject to (4.7), we use the *Dual Ascent* and *Dual Adjustment* methods of [11]; see also [42].

4.2 Dual Ascent and Dual Adjustment

Call the N variables $u_1, u_2, \cdots, u_N$ the *dual variables*. The *Dual Ascent* method initializes $u = 0$ and increases the dual variables, one at a time, until constraints (4.7) prevent any further increase in any dual variable. For $i \in \mathbb{N}$, let $\mathbb{N}_i = \{j \in \mathbb{N} \,|\, c_{ij} = 0\}$. By definition of c_{ij}, we have $\mathbb{N}_i = \{j \,|\, dist(i, j) \leq r\}$. Note that $i \in \mathbb{N}_i$. From (4.7), we can increase some dual variable u_i from 0 to 1 only if $\sum_{i=1}^{N} \max\{0, u_i - c_{ij}\} = 0$ for $j \in \mathbb{N}_i$. Once we have increased u_i then we cannot increase u_k for any k such that $c_{kj} = 0$ for some $j \in \mathbb{N}_i$. This is illustrated, for $r = 1$, in Fig. 4.2, where $c_{ij_1} = c_{ij_2} = c_{ij_3} = 0$ and $c_{j_1k_1} = c_{j_2k_2} = c_{j_2k_3} = 0$. Once we set $u_i = 1$, we cannot increase the dual variable associated with k_1 or k_2 or k_3.

Recalling that δ_j is the node degree of node j, if $c_{ij} = 0$ then the number of dual variables prevented by node j from increasing when we increase u_i is at least $\delta_j - 1$, where we subtract 1 since u_i is being increased from 0. In general, increasing u_i prevents approximately at least $\sum_{j \in \mathbb{N}_i} (\delta_j - 1)$ dual variables from being increased. This is approximate, since there may be arcs connecting the nodes in $\mathbb{N}_i$, e.g., there may be an arc between j_1 and j_2 in Fig. 4.2. However, we can ignore such

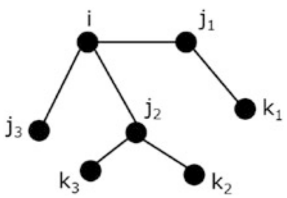

Fig. 4.2 Increasing u_i to 1 block other dual variable increases

considerations since we use $\sum_{j \in \mathbb{N}_i}(\delta_j - 1)$ only as a heuristic metric: we pre-process the data by ordering the dual variables in order of increasing $\sum_{j \in \mathbb{N}_i}(\delta_j - 1)$. We have $\sum_{j \in \mathbb{N}_i}(\delta_j - 1) = 0$ only if $\delta_j = 1$ for $j \in \mathbb{N}_i$, i.e., only if each node in $\mathbb{N}_i$ is a leaf node. This can occur only for the trivial case that $\mathbb{N}_i$ consists of two nodes (one of which is i itself) connected by an arc. For any other topology we have $\sum_{j \in \mathbb{N}_i}(\delta_j - 1) \geq 1$. For $j \in \mathbb{N}$, define $s(j)$ to be the slack in constraint (4.7) for node j, so $s(j) = 1$ if $\sum_{i=1}^{N} \max\{0, u_i - c_{ij}\} = 0$ and $s(j) = 0$ otherwise.

Having pre-processed the data, we run the following *Dual Ascent* procedure. This procedure is initialized by setting $u = 0$ and $s(j) = 1$ for $j \in \mathbb{N}$. We then examine each u_i in the sorted order and compute $\gamma \equiv \min\{s(j) \mid j \in \mathbb{N}_i\}$. If $\gamma = 0$ then u_i cannot be increased. If $\gamma = 1$ then we increase u_i from 0 to 1 and set $s(j) = 0$ for $j \in \mathbb{N}_i$, since there is no longer slack in those constraints.

Figure 4.3 shows the result of applying *Dual Ascent*, with $r = 1$, to *Zachary's Karate Club* network [37] , which has 34 nodes and 77 arcs. In this figure, node 1 is labelled as "v1", etc. The node with the smallest penalty $\sum_{j \in \mathbb{N}_i}(\delta_j - 1)$ is node 17, and the penalty (p in the figure) is 7. Upon setting $u_{17} = 1$ we have $s(17) = s(6) = s(7) = 0$; these nodes are pointed to by arrows in the figure. The node with the next smallest penalty is node 25, and the penalty is 12. Upon setting $u_{25} = 1$ we have $s(25) = s(26) = s(28) = s(32) = 0$. The node with the next smallest penalty is node 26, and the penalty is 13. However, u_{26} cannot be increased, since $s(25) = s(32) = 0$. The node with the next smallest penalty is node 12, and the penalty is 15. Upon setting $u_{12} = 1$ we have $s(12) = s(1) = 0$. The node with the next smallest penalty is node 27, and the penalty is 20. Upon

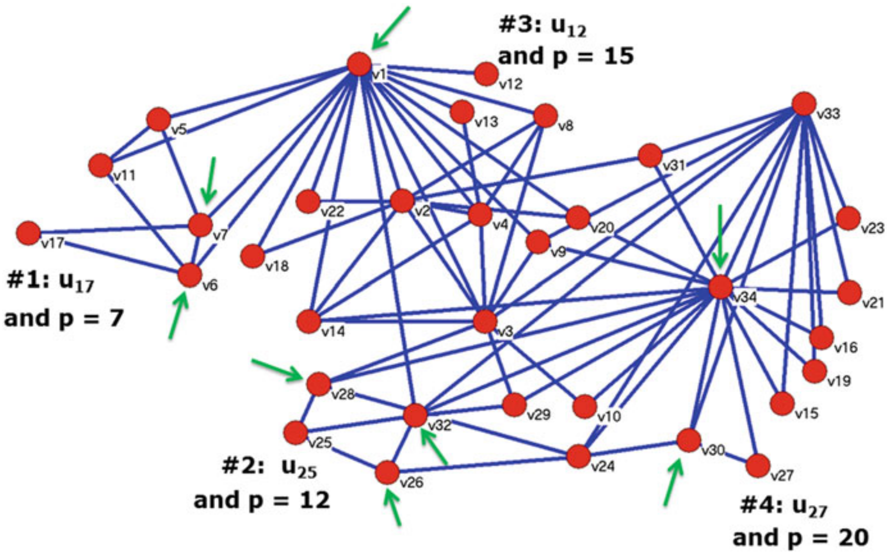

Fig. 4.3 Results of applying *Dual Ascent* to *Zachary's Karate Club* network

setting $u_{27} = 1$ we have $s(27) = s(30) = s(34) = 0$. No other dual variable can be increased, and *Dual Ascent* halts, yielding a dual objective function value of 4, which is the lower bound $B_R^L(1)$ on $B_R(1)$.

We can now calculate the upper bound $B_R^U(1)$. For $j \in \mathbb{N}$, set $y_j = 1$ if $s(j) = 0$ and $y_j = 0$ otherwise. Setting $y_j = 1$ means that the box of radius r centered at node j will be used in the covering of $\mathbb{G}$. For *Zachary's Karate Club* network, at the conclusion of *Dual Ascent* with $r = 1$ there are 12 values of j such that $s(j) = 0$; for each of these values we set $y_j = 1$.

We have shown that if u satisfies (4.7) then

$$\sum_{i=1}^{N} u_i = B_R^L(r) \le B_R(r) \le B_R^U(r) = \sum_{j=1}^{N} y_j \,.$$

If $\sum_{i=1}^{N} u_i = \sum_{j=1}^{N} y_j$ then we have found a minimal covering. If $\sum_{i=1}^{N} u_i < \sum_{j=1}^{N} y_j$ then we use a *Dual Adjustment* procedure [11] to attempt to close the gap $\sum_{j=1}^{N} y_j - \sum_{i=1}^{N} u_i$. For *Zachary's Karate Club* network, for $r = 1$ we have $\sum_{j=1}^{N} y_j - \sum_{i=1}^{N} u_i = 8$.

The *Dual Adjustment* procedure is motivated by the complementary slackness optimality conditions of linear programming. Let (x, y) be feasible for **PLP** and let (u, w) be feasible for **DLP**, where $w_{ij} = \max\{0, u_i - c_{ij}\}$. The complementary slackness conditions state that (x, y) is optimal for **PLP** and (u, w) is optimal for **DLP** if

$$y_j \left(\sum_{i=1}^{N} \max\{0, u_i - c_{ij}\} - 1 \right) = 0 \text{ for } j \in \mathbb{N} \tag{4.8}$$

$$(y_j - x_{ij}) \max\{0, u_i - c_{ij}\} = 0 \text{ for } i, j \in \mathbb{N} \,. \tag{4.9}$$

We can assume that x is binary, since as mentioned above, we can assign each i to a single k (where k depends on i) such that $y_k = 1$ and $c_{ik} = 0$. We say that a node $j \in \mathbb{N}$ is "open" (i.e., the box centered at node j is used in the covering of $\mathbb{G}$) if $y_j = 1$; otherwise, j is "closed." When (x, y) and u are feasible for **PLP** and **DLP**, respectively, and x is binary, constraints (4.9) have a simple interpretation: if for some i we have $u_i = 1$ then there can be at most one open node j such that $dist(i, j) \le r$. For suppose to the contrary that $u_i = 1$ and there are two open nodes j_1 and j_2 such that $dist(i, j_1) \le r$ and $dist(i, j_2) \le r$. Then $c_{ij_1} = c_{ij_2} = 0$. Since x is binary, by (4.2), either $x_{ij_1} = 1$ or $x_{ij_2} = 1$. Suppose without loss of generality that $x_{ij_1} = 1$ and $x_{ij_2} = 0$. Then

$$(y_{j_1} - x_{ij_1}) \max\{0, u_i - c_{ij_1}\} = (y_{j_1} - x_{ij_1}) u_i = 0$$

but

$$(y_{j_2} - x_{ij_2}) \max\{0, u_i - c_{ij_2}\} = y_{j_2} u_i = 1\,,$$

so complementary slackness fails to hold. This argument is easily extended to the case where there are more than two open nodes such that $dist(i, j) \leq r$. The conditions (4.9) can also be visualized using Fig. 4.2, where $c_{ij_1} = c_{ij_2} = c_{ij_3} = 0$. If $u_i = 1$ then at most one node in the set $\{i, j_1, j_2, j_3\}$ can be open.

If $B_R^U(r) > B_R^L(r)$, we run the following *Dual Adjustment* procedure to close some nodes, and construct x, to attempt to satisfy constraints (4.9). Define

$$Y = \{j \in \mathbb{N} \mid y_j = 1\}\,,$$

so Y is the set of open nodes. The *Dual Adjustment* procedure, which follows *Dual Ascent*, has two steps.

Step 1 For $i \in \mathbb{N}$, let $\alpha(i)$ be the "smallest" node in Y such that $c_{i,\alpha(i)} = 0$. By "smallest" node we mean the node with the smallest node index, or the alphabetically lowest node name; any similar tie-breaking rule can be used. If for some $j \in Y$ we have $j \neq \alpha(i)$ for $i \in \mathbb{N}$, then j can be closed, so we set $Y = Y - \{j\}$. In words, if the chosen method of assigning each node to a box in the covering results in the box centered at j never being used, then j can be closed.

Applying Step 1 to *Zachary's Karate Club* network with $r = 1$, using the tie-breaking rule of the smallest node index, we have, for example, $\alpha(25) = 25$, $\alpha(26) = 25$, $\alpha(27) = 27$, and $\alpha(30) = 27$. After computing each $\alpha(i)$, we can close nodes 7, 12, 17, and 28, as indicated by the bold **X** next to these nodes in Fig. 4.4. After this step, we have $Y = \{1, 6, 25, 26, 27, 30, 32, 34\}$. This step lowered the primal objective function from 12 (since originally $|Y| = 12$) to 8.

Step 2 Suppose we consider closing j, where $j \in Y$. We consider the impact of closing j on i, for $i \in \mathbb{N}$. If $j \neq \alpha(i)$ then closing j has no impact on i, since i is not assigned to the box centered at j. If $j = \alpha(i)$ then closing j is possible only if there is another open node $\beta(i) \in Y$ such that $\beta(i) \neq \alpha(i)$ and $c_{i,\beta(i)} = 0$ (i.e., if there is another open node, distinct from $\alpha(i)$, whose distance from i does not exceed r). Thus we have the rule: close j if for $i \in \mathbb{N}$ either

$$j \neq \alpha(i)$$

or

$$j = \alpha(i) \text{ and } \beta(i) \text{ exists.}$$

Once we close j and set $Y = Y - \{j\}$ we must recalculate $\alpha(i)$ and $\beta(i)$ (if it exists) for $i \in \mathbb{N}$.

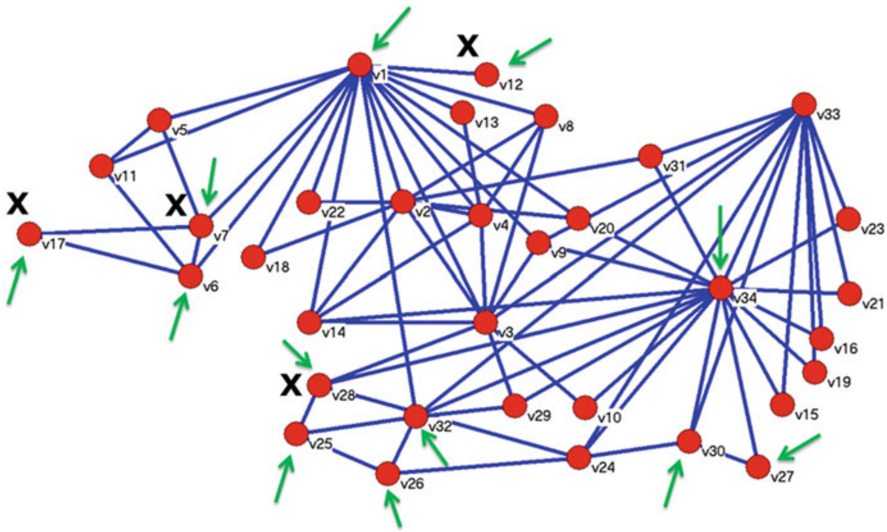

Fig. 4.4 Closing nodes in *Zachary's Karate Club* network

Applying Step 2 to *Zachary's Karate Club* network with $r = 1$, we find that, for example, we cannot close node 1, since $1 = \alpha(5)$ and $\beta(5)$ does not exist. Similarly, we cannot close node 6, since $6 = \alpha(17)$ and $\beta(17)$ does not exist. We can close node 25, since $25 = \alpha(25)$ but $\beta(25) = 26$ (i.e., we can reassign node 25 from the box centered at 25 to the box centered at 26), $25 = \alpha(26)$ but $\beta(26) = 26$, $25 = \alpha(28)$ but $\beta(28) = 34$, and $25 = \alpha(32)$ but $\beta(32) = 26$. After recomputing $\alpha(i)$ and $\beta(i)$ for $i \in \mathbb{N}$, we determine that node 26 can be closed. Continuing in this manner, we determine that nodes 27 and 30 can be closed, yielding $Y = \{1, 6, 32, 34\}$. Since now the primal objective function value and the dual objective function value are both 4, we have computed a minimal covering. When we execute *Dual Ascent* and *Dual Adjustment* for *Zachary's Karate Club* network with $r = 2$ we obtain primal and dual objective function values of 2, so again a minimal covering has been found.

4.3 Bounding the Fractal Dimension

Assume that for some positive constant a we have

$$B_R(r) = a(2r + 1)^{-d_B} . \tag{4.10}$$

Suppose we have computed $B_R^L(r)$ and $B_R^U(r)$ for $r = 1, 2, \cdots, K$. From

$$B_R^L(r) \leq B_R(r) \leq B_R^U(r)$$

we obtain, for $r = 1, 2, \cdots, K$,

$$\log B_R^{\,L}(r) \le \log a - d_B \log(2r+1) \le \log B_R^{\,U}(r)\,. \tag{4.11}$$

The system (4.11) of $2K$ inequalities may be infeasible, i.e., it may have no solution a and d_B. If the system (4.11) is feasible, we can formulate a linear program to determine the maximal and minimal values of d_B [44]. For simplicity of notation, let the K values $\log(2r+1)$ for $r = 1, 2, \cdots, K$ be denoted by x_k for $k = 1, 2, \cdots, K$, so $x_1 = \log(3)$, $x_2 = \log(5)$, $x_3 = \log(7)$, etc. For $k = 1, 2, \cdots, K$, let the K values of $\log B_R^{\,L}(r)$ and $\log B_R^{\,U}(r)$ be denoted by y_k^L and y_k^U, respectively. Let $b = \log a$. The inequalities (4.11) can now be expressed as

$$y_k^L \le b - d_B\, x_k \le y_k^U\,.$$

The minimal value of d_B is the optimal objective function value of **BCLP** (Box Counting Linear Program):

$$\begin{aligned} \text{minimize} \quad & d_B \\ \text{subject to} \quad & b - d_B\, x_k \ge y_k^L \text{ for } 1 \le k \le K \\ & b - d_B\, x_k \le y_k^U \text{ for } 1 \le k \le K. \end{aligned}$$

This linear program has only two variables, b and d_B. Let $d_B^{\min}$ and $b^{\min}$ be the optimal values of d_B and b, respectively. Now we change the objective function of **BCLP** from *minimize* to *maximize*, and let $d_B^{\max}$ and $b^{\max}$ be the optimal values of d_B and b, respectively, for the *maximize* linear program. The box counting dimension d_B, assumed to exist by (4.10), satisfies

$$d_B^{\min} \le d_B \le d_B^{\max}\,.$$

For example [44], for the much-studied *jazz* network [19], the linear program **BCLP** is feasible, and solving the *minimize* and *maximize* linear programs yields $2.11 \le d_B \le 2.59$.

Feasibility of **BCLP** does not imply that the box counting relationship (4.10) holds, since the upper and lower bounds might be so far apart that alternative relationships could be posited. If the linear program is infeasible, we can assert that the network does *not* satisfy the box counting relationship (4.10). Yet even if **BCLP** is infeasible, it might be so "close" to feasible that we nonetheless want to calculate d_B. When **BCLP** is infeasible, we can compute d_B using the solution of **BCQP** (box counting quadratic program), which minimizes the sum of the squared distances to the $2K$ bounds [44]:

$$\begin{aligned} \text{minimize} \quad & \sum_{k=1}^{K} (u_k^2 + v_k^2) \\ \text{subject to} \quad & u_k = (b - d_B\, x_k) - y_k^L \text{ for } 1 \le k \le K \\ & v_k = y_k^U - (b - d_B\, x_k) \text{ for } 1 \le k \le K. \end{aligned}$$