Appendix A: Omitted Proofs

Proof of Theorem 1. For a given function $f(\boldsymbol{x})$ that is convex in $\boldsymbol{x}_I$, define $z^1 = \max\{f(\boldsymbol{x})|\boldsymbol{x} \in \bigcup_{j\in J} \mathcal{X}(P_I^j)\}$ and $z^2 = \max\{f(\boldsymbol{x})|\boldsymbol{x} \in \bigcup_{j\in J}\bigcup_{k\in K_j} R_j^k\}$. We first show that $z^1 = z^2$. For each $j \in J$, we write that $\mathcal{X}(P_I^j) \subseteq \mathcal{X}(\bigcup_{k\in K_j} R_j^k) \subseteq \bigcup_{k\in K_j} R_j^k$, where the first inclusion follows from condition (iii), and the second inclusion follows from definition of extreme points. Therefore, $z^1 \le z^2$. The above inclusions also show that $\bigcup_{j\in J}\mathcal{X}(P_I^j)$ is a finite set since $\bigcup_{j\in J}\mathcal{X}(\bigcup_{k\in K_j} R_j^k)$ is finite because of the finiteness of J, K_j and the set of extreme points of hyper-rectangles R_j^k. Let $\boldsymbol{x}^*$ be an optimal solution for z^2. Such optimal solution exists because of the compactness of the feasible region. It follows that $\boldsymbol{x}^* \in \bigcup_{k\in K_{j^*}} R_{j^*}^k$ for some $j^* \in J$. Condition (i) implies that coordinates $i \in N\setminus I$ are fixed in $P_I^{j^*}$, and condition (iii) implies that these coordinates must also be fixed for $\bigcup_{k\in K_{j^*}} R_{j^*}^k$. Therefore, x_i^* is fixed at coordinates $i \in N\setminus I$. Since $f(\boldsymbol{x})$ is convex in the unfixed variables $\boldsymbol{x}_I$, its maximum over $\bigcup_{k\in K_{j^*}} R_{j^*}^k$ occurs at an extreme point $\bar{\boldsymbol{x}}$ of $\text{conv}(\bigcup_{k\in K_{j^*}} R_{j^*}^k)$. Condition (iii) implies that $\bar{\boldsymbol{x}} \in \mathcal{X}(P_I^{j^*})$, proving that $z^1 \ge z^2$. For the second part of the proof, we show that $z^3 = \max\{f(\boldsymbol{x})|\boldsymbol{x}\in\mathcal{P}\} = z^1 = z^2$. It follows from condition (ii) that $z^1 \le z^3 \le z^4$ where $z^4 = \max\{f(\boldsymbol{x})|\boldsymbol{x} \in \bigcup_{j\in J} P_I^j\}$. Using an argument similar to that given above, we conclude that the optimal value z^4 is attained at an extreme point of $P_I^{j^*}$ for some $j^* \in J$, which is also an extreme point of $\bigcup_{k\in K_{j^*}} R_{j^*}^k$. Therefore, using the definition of z^4 and z^2, we write that $z^4 \le z^2 = z^1$, proving the result. $\square$

Proof of Proposition 1. Consider any point $\bar{\boldsymbol{x}}_{N\setminus I} \in \text{proj}_{x_{N\setminus I}}(\mathcal{P})$, and define the indicator function

$$\delta(\boldsymbol{x}|\bar{\boldsymbol{x}}_{N\setminus I}) = \begin{cases} 0, & \text{if } \boldsymbol{x}_{N\setminus I} = \bar{\boldsymbol{x}}_{N\setminus I} \\ -\infty, & \text{else.}\end{cases}$$

We can use $\delta(\boldsymbol{x}|\bar{\boldsymbol{x}}_{N\setminus I})$ as the objective function in the relation $\max\{f(\boldsymbol{x}) \mid \boldsymbol{x}\in\mathcal{P}\} = \max\{f(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{Q}\}$ as it is convex in $\boldsymbol{x}_I$. This follows from the fact that $\delta(\boldsymbol{x}|\bar{\boldsymbol{x}}_{N\setminus I}) = 0$ when $\boldsymbol{x}_{N\setminus I} = \bar{\boldsymbol{x}}_{N\setminus I}$, and it is $-\infty$ otherwise, which is *improper* convex; see Rockafeller (1970). We conclude that $\bar{\boldsymbol{x}}_{N\setminus I} \in \text{proj}_{x_{N\setminus I}}(\mathcal{Q})$. Using a similar argument for the reverse direction, we conclude that $\text{proj}_{x_{N\setminus I}}(\mathcal{P}) = \text{proj}_{x_{N\setminus I}}(\mathcal{Q})$. This also shows that $\text{proj}_{x_{N\setminus I}}(\mathcal{P})$ is finite as $\mathcal{Q}$ is finite by assumption. For any point $\bar{\boldsymbol{x}}_{N\setminus I} \in \text{proj}_{x_{N\setminus I}}(\mathcal{P})$, define $\mathcal{P}(\bar{\boldsymbol{x}}_{N\setminus I}) = \mathcal{P}\cap\{\boldsymbol{x}|\boldsymbol{x}_{N\setminus I} = \bar{\boldsymbol{x}}_{N\setminus I}\}$, and $\mathcal{Q}(\bar{\boldsymbol{x}}_{N\setminus I}) = \mathcal{Q}\cap\{\boldsymbol{x}|\boldsymbol{x}_{N\setminus I} = \bar{\boldsymbol{x}}_{N\setminus I}\}$. For any convex function $f_I(\boldsymbol{x}_I)$ defined in the space of variables $\boldsymbol{x}_I$, and any point $\bar{\boldsymbol{x}}_{N\setminus I} \in \text{proj}_{x_{N\setminus I}}(\mathcal{P})$, we have that

$$\max\{f_I(\boldsymbol{x}_I)|\boldsymbol{x}\in\mathcal{P}(\bar{\boldsymbol{x}}_{N\setminus I})\} \tag{8a}$$

$$= \max\{\delta(\boldsymbol{x}|\bar{\boldsymbol{x}}_{N\setminus I}) + f_I(\boldsymbol{x}_I)|\boldsymbol{x}\in\mathcal{P}\} \tag{8b}$$

$$= \max\{\delta(\boldsymbol{x}|\bar{\boldsymbol{x}}_{N\setminus I}) + f_I(\boldsymbol{x}_I)|\boldsymbol{x}\in\mathcal{Q}\} \tag{8c}$$

$$= \max\{f_I(\boldsymbol{x}_I)|\boldsymbol{x}\in\mathcal{Q}(\bar{\boldsymbol{x}}_{N\setminus I})\}, \tag{8d}$$

where (8a) and (8d) follow from the definition of the indicator function above, and (8b) follows from the assumption. Since $f_I(\boldsymbol{x}_I)$ is convex, the optimal value of (8a) is attained at an extreme point $\hat{\boldsymbol{x}}$ of $\mathcal{P}(\bar{\boldsymbol{x}}_{N\setminus I})$. For each such extreme point, there are infinitely many convex functions $\hat{f}_I(\boldsymbol{x}_I)$ with $\hat{\boldsymbol{x}}$ as their unique maximizer over $\mathcal{P}(\bar{\boldsymbol{x}}_{N\setminus I})$. For every one of these functions, according to the chain equalities (8a)–(8d), $\mathcal{Q}$ has a point that matches the optimal value. Since $\mathcal{Q}$ is finite, it must be that $\hat{\boldsymbol{x}} \in \mathcal{Q}(\bar{\boldsymbol{x}}_{N\setminus I})$. This shows that the set of extreme points of $\mathcal{P}(\bar{\boldsymbol{x}}_{N\setminus I})$ is finite. Similarly, it can be shown that for any extreme point $\dot{\boldsymbol{x}}$ of $\text{conv}(\mathcal{Q}(\bar{\boldsymbol{x}}_{N\setminus I}))$, we have that $\dot{\boldsymbol{x}} \in \mathcal{P}(\bar{\boldsymbol{x}}_{N\setminus I})$. As a result, $\text{conv}(\mathcal{P}(\bar{\boldsymbol{x}}_{N\setminus I})) = \text{conv}(\mathcal{Q}(\bar{\boldsymbol{x}}_{N\setminus I}))$ for every $\bar{\boldsymbol{x}}_{N\setminus I} \in \text{proj}_{x_{N\setminus I}}(\mathcal{P})$. Now, construct sets P_I^j in Theorem 1 as $\text{conv}(\mathcal{P}(\bar{\boldsymbol{x}}_{N\setminus I}))$ for every $\bar{\boldsymbol{x}}_{N\setminus I} \in \text{proj}_{x_{N\setminus I}}(\mathcal{P})$, which yields a finite collection. Condition (i) is satisfied as each set P_I^j is restricted at $\{\boldsymbol{x} | \boldsymbol{x}_{N\setminus I} = \bar{\boldsymbol{x}}_{N\setminus I}\}$. Condition (ii) holds since for any extreme point of P_I^j, there is a point of $\mathcal{P}$ by construction, and since any point $\bar{\boldsymbol{x}} \in \mathcal{P}$ satisfies $\bar{\boldsymbol{x}} \in \mathcal{P}(\bar{\boldsymbol{x}}_{N\setminus I}) \subseteq \text{conv}(\mathcal{P}(\bar{\boldsymbol{x}}_{N\setminus I}))$. For condition (iii), rectangles R_j^k can be considered as the set of extreme points of P_I^j, which has been shown to be finite. □

Proof of Lemma 1. The result is obtained as a special case of Theorem 1, where sets P_I^j and R_j^k coincide, i.e., $P_I^j = R_j^1$ for all $j \in J$. The conditions and the result follow immediately. □

Proof of Proposition 2. For a given DD $\mathcal{D}$, define a node-sequence as an ordered set of connected nodes from the root to the terminal, i.e., $\boldsymbol{u} = (u_1, u_2, \cdots, u_{n+1})$ where $u_i \in \mathcal{U}_i$ for $i \in N \cup \{n+1\}$. Let U be the collection of all node-sequences of $\mathcal{D}$. For $\boldsymbol{u} \in U$, define

$$S_I(\boldsymbol{u}) = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{ll} l^{\min}_{(u_i,u_{i+1})} \leq x_i \leq l^{\max}_{(u_i,u_{i+1})}, & \forall i \in I \\ x_i = l_{(u_i,u_{i+1})}, & \forall i \in N \setminus I \end{array} \right\}.$$

Viewing $\text{Sol}(\mathcal{D})$ as a compact set $\mathcal{P}$ in Lemma 1, it is straightforward to verify that sets $S_I(\boldsymbol{u})$ satisfy the conditions for hyper-rectangles R_I^j. It also follows from the definition of (virtual) DDs that $\text{Sol}(\hat{\mathcal{D}}) = \bigcup_{\boldsymbol{u}\in U} S_I(\boldsymbol{u})$ and $\text{Sol}(\bar{\mathcal{D}}) = \bigcup_{\boldsymbol{u}\in U} \mathcal{X}(S_I(\boldsymbol{u}))$. The result follows from Lemma 1. □

Proof of Corollary 1. For the direct implication, assume that a given compact set $\mathcal{P}$ admits a rectangular decomposition w.r.t. I through sets P_I^j for $j \in J$. Then, the DD that encodes the finite collection of points $\bigcup_{j\in J} \mathcal{X}(P_I^j)$ provides the desired DD representation because of Theorem 1. For the reverse implication, assume that a given compact set $\mathcal{P}$ is DD-representable w.r.t. I through a DD $\mathcal{D}$. Since $\text{Sol}(\mathcal{D})$ is finite, it follows from Proposition 1 that $\mathcal{P}$ admits a rectangular decomposition w.r.t. I. □

Proof of Corollary 2. Since $\mathcal{Q}$ is bounded, there are finitely many points $\bar{\boldsymbol{x}} \in \text{proj}_x(\mathcal{P})$. For each such point, it follows from the definition of projection and the boundedness of $\mathcal{Q}$ that there exists an interval $[l_{\bar{x}}, u_{\bar{x}}]$ such that $(\bar{\boldsymbol{x}}; \bar{y}) \in \mathcal{P}$ for every $\bar{y} \in [l_{\bar{x}}, u_{\bar{x}}]$. As a result, we can write that

$\mathcal{P} = \bigcup_{\bar{\boldsymbol{x}} \in \text{proj}_x(\mathcal{P})} R_y^{\bar{x}}$ where $R_y^{\bar{x}} = \{(\boldsymbol{x}; y) \in \mathbb{R}^{n+1} \,|\, \boldsymbol{x} = \bar{\boldsymbol{x}},\, y \in [l_{\bar{x}}, u_{\bar{x}}]\}$. Hyper-rectangles $R_y^{\bar{x}}$ satisfy the conditions of Lemma 1, and hence admits a rectangular decomposition w.r.t. the index of variable y which is $n+1$. The result follows from Corollary 1. $\square$

Proof of Theorem 2. First, we show that the Algorithm 1 terminates after a finite number of iterations. To this end, we argue that each loop in the algorithm is repeated for a finite number of iterations. The outer *while* loop is executed for each member of the partial assignment set $\hat{\mathcal{X}}$. These partial assignments are generated based on longest r-u paths associated with nodes u in the exact cut set of relaxed DDs $\overline{\mathcal{D}}$, which contain a finite number of paths by definition; see line 22 of the algorithm. Each resulting partial assignment $\hat{\boldsymbol{x}}$ is different due to the structure of DDs that do not admit multiple paths with similar encoding values. As a result, an upper bound for the number of partial assignments that can be included in $\hat{\mathcal{X}}$ is all possible partial assignments of the solution set of $\mathcal{M}$ for $\boldsymbol{x}$ variables. Because the discrete set $\mathcal{P}$ in the description of $\mathcal{M}$ is bounded by assumption, we conclude that $\hat{\mathcal{X}}$ contains a finite number of elements. For the *repeat* loop in lines 6–10, the goal is to obtain the optimal value of the solution set represented by $\underline{\mathcal{D}}$ subject to the constraints of the subproblem $\mathcal{S}(.)$. It follows from the property of BD method that the resulting optimality and feasibility cuts correspond to the extreme points and extreme rays of the dual of the subproblem, which is independent of the choice of the fixed point $\underline{\boldsymbol{x}}$. Since the feasible region of this dual problem is a polyhedron, the set of its extreme points and rays are finite, which yield a finite set of optimality and feasibility cuts that can be added through the execution of this loop. It is also known in the BD literature (Conforti et al. (2014a)) that the same optimality/feasibility cuts cannot be generated repeatedly through different iterations, since the added cuts remain in the description of the master DD due to refinement. A similar argument can be used to show that the number of iterations of the *repeat* loop in lines 18–21 is finite, thereby yielding the result.

Next, we show that the outputs $(\boldsymbol{x}^*, z^*)$ and w^* of Algorithm 1 give an optimal solution and optimal value of $\mathcal{H}$, respectively. To this end, we first prove that $(\boldsymbol{x}^*, z^*)$ is a feasible solution to $\mathcal{H}$ and w^* is a lower bound for its optimal value. This solution is updated at line 12 of Algorithm 1 as the point encoding a longest r-t path of $\underline{\mathcal{D}}$, after being refined with respect to optimality and feasibility cuts generated from subproblems $\mathcal{S}(.)$. We write that $(\boldsymbol{x}^*, z^*) \in \text{Sol}(\underline{\mathcal{D}}) \subseteq \mathcal{M}^C(\hat{\boldsymbol{x}}) \subseteq \mathcal{M}$, where the second inclusion holds because $\underline{\mathcal{D}}$ is a restricted DD associated with $\mathcal{M}^C(\hat{\boldsymbol{x}})$ for some C and $\hat{\boldsymbol{x}}$, and the third inclusion follows from the fact that a feasible solution to $\mathcal{M}^C(\hat{\boldsymbol{x}})$ is feasible to $\mathcal{M}$ by definition. Further, the BD structure implies that the point encoding a longest path obtained at the termination of the loop in line 6–10 satisfies the constraints of the subproblem $\mathcal{S}(.)$. As a result, $(\boldsymbol{x}^*, z^*)$ satisfies the constraints of both the master and the subproblem, hence being feasible to $\mathcal{H}$. Using a similar argument, we obtain that w^* is a lower bound for the optimal value of $\mathcal{M}$ subject to constraints in $\mathcal{S}(.)$, since w^* is the length of the longest path in the restricted DD representing a restriction of $\mathcal{M}$, after valid optimality and feasibility cuts are added based on the subproblem. Now, we show that $(\boldsymbol{x}^*, z^*)$ is indeed an optimal solution to $\mathcal{H}$. Assume by contradiction that there exists an optimal solution $(\tilde{\boldsymbol{x}}, \tilde{z})$ of $\mathcal{H}$ with optimal value $\tilde{w}$ such that $\tilde{w} > w^*$. There are three cases. (i) Assume that $\tilde{\boldsymbol{x}}$ is added as a partial assignment to set $\hat{\mathcal{X}}$ at some iteration of the algorithm. For the while loop where $\tilde{\boldsymbol{x}}$ is selected at line 3, all $\boldsymbol{x}$ variables are fixed for the restricted DD $\underline{\mathcal{D}}$. Therefore, $\tilde{\boldsymbol{x}}$ is used as the input for the subproblem, i.e., we solve $\mathcal{S}(\tilde{\boldsymbol{x}})$, which yields the optimal value $\tilde{z}$ since $(\tilde{\boldsymbol{x}}, \tilde{z})$ is an optimal solution of $\mathcal{H}$. Since the weights on the arcs of the restricted DD $\underline{\mathcal{D}}$ are set as the coefficients of variables in the linear objective function of $\mathcal{H}$, the length of the r-t path of $\underline{\mathcal{D}}$ encoded by $(\tilde{\boldsymbol{x}}, \tilde{z})$ is $\tilde{w}$. If follows from the contradiction assumption that $\tilde{w} > w^*$ in line 11 of Algorithm 1, which leads to updating the optimal solution and optimal value to $(\tilde{\boldsymbol{x}}, \tilde{z})$ and $\tilde{w}$. As a result, $(\boldsymbol{x}^*, z^*)$ and w^* cannot be returned by the algorithm as the output, a contradiction. (ii) Assume that $\tilde{\boldsymbol{x}}$ is not added as a partial assignment to set $\hat{\mathcal{X}}$ because the algorithm terminates before such a partial assignment is reached in line 22. This implies that there must be a partial assignment $\hat{\boldsymbol{x}} \in \hat{\mathcal{X}}$ with $\hat{x}_i = \tilde{x}_i$ for $i = 1, \dots, j-1$ for some $j \in N$ such that the relaxed DD $\overline{\mathcal{D}}$ associated with $\mathcal{M}^C(\hat{\boldsymbol{x}})$ for some C is pruned without reaching line 22. The only possibility for this event is that the length $\bar{w}$ of the longest r-t path of $\overline{\mathcal{D}}$ must satisfy $\bar{w} \le w^*$, violating the condition in line 17. However, because of the facts that $\overline{\mathcal{D}}$ is a relaxed DD associated with $\mathcal{M}^C(\hat{\boldsymbol{x}})$, and that $(\tilde{\boldsymbol{x}}, \tilde{z})$ must be a feasible solution to $\mathcal{M}^C(\hat{\boldsymbol{x}})$, we conclude that $\bar{w} \ge \tilde{w}$. Combining this inequality with that of the line above, we obtain $\tilde{w} \le w^*$, a contradiction to the initial assumption on the optimal value of the problem. (iii) Assume that $\tilde{\boldsymbol{x}}$ is not added as a partial assignment to set $\hat{\mathcal{X}}$ because the longest r-u path chosen in line 22 deviates from $\tilde{\boldsymbol{x}}$ for some node u of the the path associated with $\tilde{\boldsymbol{x}}$. This implies that there must be a partial assignment $\hat{\boldsymbol{x}} \in \hat{\mathcal{X}}$ with $\hat{x}_i = \tilde{x}_i$ for $i = 1, \dots, j-1$ for some $j \in N$ such that the relaxed DD $\overline{\mathcal{D}}$ associated with $\mathcal{M}^C(\hat{\boldsymbol{x}})$ for some C has an exact cut set that contains node u in some layer $k \ge j$. Let $\dot{\boldsymbol{x}}$ be the solution encoding the longest r-u path of $\overline{\mathcal{D}}$ that is chosen in line 22 in place of $\tilde{\boldsymbol{x}}$. It follows from definition of exact cut set that an extension of $\dot{\boldsymbol{x}}$ with components $\dot{x}_i = \tilde{x}_i$ for $i = k, \dots, n$ and $\dot{z} = \tilde{z}$ is a feasible solution to $\mathcal{H}$. However, the above assumption implies that the length $\dot{w}$ of the path encoded by $(\dot{\boldsymbol{x}}, \dot{z})$ is no less than $\tilde{w}$. If $\dot{w} > \tilde{w}$, then we have reached a contradiction to the assumption that $\tilde{w}$ is the optimal value of $\mathcal{H}$. If $\dot{w} = \tilde{w}$, we may repeat the contradiction arguments for the new point $(\dot{\boldsymbol{x}}, \dot{z})$ until we exhaust all possible replacements for the assumed optimal solution. The last such solution must either fall in case (i) or case (ii) above, reaching contradiction. □

Proof of Theorem 3. First, we give a useful observation from Algorithm 2. It follows from this algorithm that state values (s_u^+, s_u^-) at each node $u \in \mathcal{U}_j$ records the number of time periods passed since the last start-up and shut-down of the unit at time j, as these values reset to 1 whenever there is change in the unit status over two consecutive periods, and are incremented by one if the unit status remains the same. These state values imply the status of the unit at the beginning of each period, i.e., the unit is down when $s_u^+ \geq s_u^-$, and it is up otherwise.

For the direct implication, assume that $(\dot{\boldsymbol{x}}, \dot{z})$ encodes an r-t path of length $\dot{c}$ in the equivalence class formed by $\mathcal{D}$. We construct an extended solution $(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}}, \dot{\bar{\boldsymbol{y}}}, \dot{\boldsymbol{q}}, \dot{z})$ of (2) with objective value $\dot{c}$ as follows. First, we show that $\dot{x}_j \in \{0,1\}$ for all $j \in \mathcal{T}$ and $\dot{z} \in [-\Gamma, \Gamma]$, thereby satisfying domain constraints in (2). The definition of equivalence class implies that, for each $j \in \mathcal{T}$, there exists a node pair $(u,v) \in \mathcal{A}_j \times \mathcal{A}_{j+1}$ such that $\dot{x}_j \in [l^{\min}_{(u,v)}, l^{\max}_{(u,v)}]$, i.e., the variable value belongs to the interval defined by the min and max label values of the arcs connecting nodes u and v; see Section 2.3. It follows from the construction of $\mathcal{D}$ in Algorithm 2 that two consecutive nodes in layers $1, \dots, T$ cannot be connected with multiple arcs with different label values, since different arc labels lead to different state values for the head node. As a result, we must have $l^{\min}_{(u,v)} = l^{\max}_{(u,v)} \in \{0,1\}$. The argument for $\dot{z} \in [-\Gamma, \Gamma]$ follows directly from the construction and label values assigned to the arcs at the last layer of $\mathcal{D}$. To construct the extended solution, for each $j \in \mathcal{T}$, define $\dot{y}_j = 1$ if $\dot{x}_{j-1} = 0$ and $\dot{x}_j = 1$, and $\dot{y}_j = 0$ otherwise. Similarly, define $\dot{\bar{y}}_j = 1$ if $\dot{x}_{j-1} = 1$ and $\dot{x}_j = 0$, and $\dot{\bar{y}}_j = 0$ otherwise. These definitions guarantee the satisfaction of (1c). Further, let $u \in \mathcal{U}_j$ be the node at layer j of the path encoding $(\dot{\boldsymbol{x}}, \dot{z})$, and define $\dot{q}_j = K_{s_u^-}$ if $\dot{x}_j = 1$ and $\dot{x}_{j-1} = 0$, and $\dot{q}_j = 0$ otherwise. These value assignments satisfy (1b) because this constraint implies that when a unit changes status from down to up, i.e., $x_j = 1$ and $x_{j-1} = 0$, then $q_j \geq \max_{k=1,\dots,s_u^-} K_k$, as s_u^- represents the number of periods that the unit has been down consecutively before going up. Since the start-up cost function is assumed to be logarithmic, we have that $\max_{k=1,\dots,s_u^-} K_k = K_{s_u^-}$. Further, since $\dot{q}_j$ takes the maximum value at equality, it yields a non-redundant solution. It remains to show that the constructed point satisfies constraints (1d) and (1e). Assume by contradiction that there exists $j^* \in \mathcal{T}$ for which constraint (1d) is violated by $(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}}, \dot{\bar{\boldsymbol{y}}}, \dot{\boldsymbol{q}}, \dot{z})$. Note that (1d) models two requirements at time j^*: (i) if the unit is down, then it could not have started up in the last L time periods; and (ii) if the unit is up, then it could not have started up more than once in the last L time periods. For the contradiction, assume first that condition (i) is violated, i.e., $\dot{x}_{j^*} = 0$ and there exists $\bar{j} \in \{j^* - L + 1, \dots, j^*\}$ such that $\dot{y}_{\bar{j}} = 1$. It follows from construction that $\dot{x}_{\bar{j}} = 1$ and $\dot{x}_{\bar{j}-1} = 0$. Let $(u,v) \in \mathcal{U}_{\bar{j}} \times \mathcal{U}_{\bar{j}+1}$ be the nodes at layers $\bar{j}$ and $\bar{j}+1$ of the r-t path encoding $(\dot{\boldsymbol{x}}, \dot{z})$. We have $s_u^+ \geq s_u^-$ because of the earlier argument on the state values. Since the arc connecting u to v on the r-t path is a 1-arc $(\dot{x}_{\bar{j}} = 1)$, we have $s_v^+ = 1$; see line 6 of Algorithm 2. It follows from the algorithm steps that the only way for the unit to be able to shut down after $\bar{j}$ is to satisfy the condition of line 9, i.e., $s_h^+ \geq L$ for some node h in layers $\bar{j}+L,\ldots,T$. Since $\bar{j} \geq j^* - L + 1$ by definition, we obtain that $j^* < \bar{j} + L$, and hence $\dot{x}_{j^*}$ cannot be equal to zero, a contradiction. Next, assume that condition (ii) above is violated for the contradiction assumption, i.e., the unit starts up more than once in periods $j^* - L + 1, \ldots, j^*$. It is easy to verify that there exists a layer $\tilde{j} \in \{j^* - L + 1, \ldots, j^*\}$ for which condition (i) is violated. Therefore, an argument similar to that of condition (i) yields the desired contradiction. The contradiction for violating constraint (1e) is obtained similarly due to the symmetry in the problem structure. We conclude that $(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}}, \dot{\bar{\boldsymbol{y}}}, \dot{\boldsymbol{q}}, \dot{z})$ is feasible to (2). We next show that the length of the r-t path encoding $(\dot{\boldsymbol{x}}, \dot{z})$ matches the objective value of $(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}}, \dot{\bar{\boldsymbol{y}}}, \dot{\boldsymbol{q}}, \dot{z})$. The proof follows from considering the contribution of three terms in the objective function of (2). First, the contribution of each variable assignment $\dot{x}_j = 1$ to the objective function is c_f, which is captured in the weight of the associated 1-arcs of the r-t path through Algorithm 2 in lines 6 and 8. Second, the contribution of the start-up status of the unit to the objective function is $\dot{q}_j = K^1_{s_u^-}$ where u is the node at layer j of the path encoding $(\dot{\boldsymbol{x}}, \dot{z})$, which is captured in line 6 of Algorithm 2. Third, the contribution of $\dot{z}$ in the objective function is directly considered in the arc weight in the last layer of the equivalence class of $\mathcal{D}$.

For the reverse implication, assume that $(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}}, \dot{\bar{\boldsymbol{y}}}, \dot{\boldsymbol{q}}, \dot{z})$ is a non-redundant solution of (2) with objective value $\dot{c}$. We show that the projected point $(\dot{\boldsymbol{x}}, \dot{z})$ encodes an r-t path of length $\dot{c}$ in the equivalence class formed by $\mathcal{D}$. Assume by contradiction that no such path exists in the equivalence class of $\mathcal{D}$. Therefore, there exists an r-t path $\mathtt{P}$ of $\mathcal{D}$ with associated point $(\bar{\boldsymbol{x}}, \bar{z})$ and a layer number $j^* \in \mathcal{T} \setminus \{T\}$ such that $\bar{x}_j = \dot{x}_j$ for $j \in \{1, \ldots, j^*\}$ and the node u at layer j^*+1 of $\mathtt{P}$ does not have an outgoing arc with label value $\dot{x}_{j^*+1}$. There are four cases. For the first case, assume that $\dot{x}_{j^*} = 0$ and $\dot{x}_{j^*+1} = 0$. It follows from the state value definitions that $s_u^+ \geq s_u^-$ as $\dot{x}_{j^*} = \bar{x}_{j^*} = 0$. Line 4 of Algorithm 2 implies that u has an outgoing arc with label value $\dot{x}_{j^*+1} = 0$, a contradiction. For the second case, assume that $\dot{x}_{j^*} = 1$ and $\dot{x}_{j^*+1} = 0$. Because of constraints (1c) and (1d), we must have $\dot{x}_j = \bar{x}_j = 1$ for $j = j^* - L + 1, \ldots, j^*$. Let v be the node at layer $j^* - L + 1$ of $\mathtt{P}$. Since $\bar{x}_j = 1$ for $j = j^* - L + 1, \ldots, j^*$, it follows from line 8 of Algorithm 2 that $s_u^+ = s_v^+ + L - 1 \geq L$ as $s_v^+ \geq 1$. As a result, the condition of line 9 of Algorithm 2 is satisfied at node u, which leads to creating a 0-arc as an outgoing arc of u, a contradiction. For the third case, where $\dot{x}_{j^*} = 1$ and $\dot{x}_{j^*+1} = 1$, the contradiction is obtained similarly to the first case due to the symmetry of the problem. For the fourth case, where $\dot{x}_{j^*} = 0$ and $\dot{x}_{j^*+1} = 1$, the contradiction is achieved similarly to the second case due to the symmetry of the problem. Finally, since $\dot{z} \in [-\Gamma, \Gamma]$, it must belong to the equivalence class of $\mathcal{D}$ as $-\Gamma$ and Γ are used as label values of the pair of arcs connecting every node of layer $T+1$ to the terminal node in $\mathcal{D}$. We conclude that $(\dot{\boldsymbol{x}}, \dot{z})$ encodes an r-t path of length $\dot{c}$ in the equivalence class formed by $\mathcal{D}$. To show that the objective value $\dot{c}$ is equal to the length of the r-t path encoding $(\dot{\boldsymbol{x}}, \dot{z})$, we note that $\dot{q}^1_j = K^1_{s_h^-}$ where h is the node at layer j of the r-t path, since otherwise $(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}}, \dot{\bar{\boldsymbol{y}}}, \dot{\boldsymbol{q}}, \dot{z})$ would be a redundant solution of (2). The rest of the proof follows from an argument similar to that of the direct implication case given above. $\square$

Proof of Proposition 3. We need to show that for each path $\bar{\mathtt{P}}$ of $\bar{\mathcal{D}}$ with encoding point $(\bar{\boldsymbol{x}}, \bar{z})$, there exists a path $\tilde{\mathtt{P}}$ of $\tilde{\mathcal{D}}$ with encoding point $(\tilde{\boldsymbol{x}}, \tilde{z})$ such that $\bar{\boldsymbol{x}} = \tilde{\boldsymbol{x}}$, $\bar{z} = \tilde{z}$, and $w(\bar{\mathtt{P}}) \geq w(\tilde{\mathtt{P}})$, where $w(.)$ represent the length of the path. Define $\bar{\mathtt{P}}^k$ to be the path composed of the first k arcs of $\bar{\mathtt{P}}$ for $k \in \mathcal{T}$. We prove the result by induction on the size of k, i.e., we show that for each $\bar{\mathtt{P}}^k$ with encoding point $\bar{\boldsymbol{x}}^k$, there exists a path $\tilde{\mathtt{P}}^k$ of $\tilde{\mathcal{D}}$ with encoding point $\tilde{\boldsymbol{x}}^k$ such that $\bar{\boldsymbol{x}}^k = \tilde{\boldsymbol{x}}^k$ and $w(\bar{\mathtt{P}}^k) \geq w(\tilde{\mathtt{P}}^k)$. Note that it suffices to show the above result for k up to T, since each node at node layer $T+1$ is connected to the terminal node by two arcs with labels $-\Gamma$ and Γ in both $\bar{\mathcal{D}}$ and $\tilde{\mathcal{D}}$, and therefore a matching arc can always be found for the desired path. For the base of induction, i.e., $k = 1$, it follows from Algorithm 2 that $\bar{\mathcal{D}}$ has two arcs with labels 0 and 1 going out of the root node. These two arcs remain in $\tilde{\mathcal{D}}$ with the same weights according to Definition 2. For the induction hypothesis, assume that, for some $k \geq 2$, there exists a path $\tilde{\mathtt{P}}^{k-1}$ of $\tilde{\mathcal{D}}$ with encoding point $\tilde{\boldsymbol{x}}^{k-1}$ such that $\bar{\boldsymbol{x}}^{k-1} = \tilde{\boldsymbol{x}}^{k-1}$ and $w(\bar{\mathtt{P}}^{k-1}) \geq w(\tilde{\mathtt{P}}^{k-1})$. We show that there exists a path $\tilde{\mathtt{P}}^k$ of $\tilde{\mathcal{D}}$ with encoding point $\tilde{\boldsymbol{x}}^k$ such that $\bar{\boldsymbol{x}}^k = \tilde{\boldsymbol{x}}^k$ and $w(\bar{\mathtt{P}}^k) \geq w(\tilde{\mathtt{P}}^k)$. There are four cases. (i) Assume that $\bar{x}^k_{k-1} = 0$ and $\bar{x}^k_k = 0$. Note that $\bar{x}^k_{k-1} = \bar{x}^{k-1}_{k-1} = 0$ as they are defined over the same path $\bar{\mathtt{P}}$. It follows from the induction hypothesis that $\bar{x}^{k-1}_{k-1} = \tilde{x}^{k-1}_{k-1} = 0$. Let u be the node at node layer k of the path $\tilde{\mathtt{P}}^{k-1}$. We consider two cases for this node. For the first case, assume that u is not a merged node. As a result, it has been created directly from the modified algorithm of Definition 2, which implies that $s^+_u \geq s^-_u$. Following line 4 of Algorithm 2, there is a 0-arc going out of u in $\tilde{\mathcal{D}}$. Adding this arc to $\tilde{\mathtt{P}}^{k-1}$ yields the desired path $\tilde{\mathtt{P}}^k$, as the contribution to the objective function is 0 for this variable assignment in both paths $\bar{\mathtt{P}}^k$ and $\tilde{\mathtt{P}}^k$. For the second case, assume that u is a merged node. Let $\{v_1, \dots, v_r\}$ be the set of nodes that have been merged into u using the merging operation of Definition 3. Since u has an incoming arc with label 0, it means that one of the nodes v_i in the above set must have an incoming arc with label 0. The modified algorithm implies that $s^+_{v_i} \geq s^-_{v_i}$. The merging condition, then, dictates that $s^+_{v_i} \geq s^-_{v_i}$ for all $i \in \{1, \dots, r\}$. It follows from the merging equations that $s^+_u = \max_i s^+_{v_i} \geq \max_i s^-_{v_i} = s^-_u$. Therefore, an argument similar to that of the first case yields the desired result. (ii) Assume that $\bar{x}^k_{k-1} = 0$ and $\bar{x}^k_k = 1$. Let h be the node at node layer k of $\bar{\mathtt{P}}^k$. It follows from line 5 of Algorithm 2 that $s^-_h \geq \ell$ and that the arc weight at layer k is $c_f + K_{s^-_h}$. It is clear from the definition of s^-_h that $\bar{x}^k_{k-s^-_h-1} = 1$ and $\bar{x}^k_{k-s^-_h-1+i} = 0$ for all $i = 1, \dots, s^-_h$. It follows from the induction hypothesis that $\tilde{x}^{k-1}_{k-s^-_h-1} = 1$ and $\tilde{x}^{k-1}_{k-s^-_h-1+i} = 0$ for all $i = 1, \dots, s^-_h$. Let u and v be the nodes at node layers $k - s^-_h + 1$ and k of $\tilde{\mathtt{P}}^{k-1}$, respectively. We consider two cases for node u. For the first case, assume that u is not a merged node. As a result, it has been created directly from the modified algorithm of Definition 2, which implies that

$s^-_u = s^=_u = 1$. For the second case, assume that u is a merged node. Let $\{u_1, \dots, u_r\}$ be the set of nodes that have been merged into u using the merging operation of Definition 3. Since u has an incoming arc with label 0 right after an arc with label 1, it implies that one of the nodes u_i in the above set must have $s^-_{u_i} = s^=_{u_i} = 1$. It follows from the modified algorithm that $s^-_u \geq 1$ and $s^=_u \leq 1$, which holds for the first case as well. Using an argument similar to that given above for the nodes at layers $k - s^-_h + 1$ to k of $\tilde{\mathsf{P}}^{k-1}$, we conclude that $s^-_v \geq s^-_u + s^-_h - 1 \geq s^-_h$ and $s^=_v \leq s^=_u + s^-_h - 1 \leq s^=_h$, where the first inequalities follow from the state definitions through merging operation applied on successive node layers, and the second inequalities hold because of the relations above on state values at node u. As a result, the condition in line 5 of Algorithm 2 is satisfied at node v as $s^-_v \geq s^-_h \geq \ell$, meaning that it has an outgoing arc with label 1 and weight $c_f + K_{s^=_v} \leq c_f + K_{s^=_h}$. Adding this arc to $\tilde{\mathsf{P}}^{k-1}$ yields the desired path $\tilde{\mathsf{P}}^k$. (iii) Assume that $\bar{x}^k_{k-1} = 1$ and $\bar{x}^k_k = 1$. The result is obtained by using an argument similar to that of case (i) due to symmetry. (iv) Assume that $\bar{x}^k_{k-1} = 1$ and $\bar{x}^k_k = 0$. An argument similar to that of case (ii) proves the result due to symmetry. □

Proof of Proposition 4. We prove the equivalence by showing that for each feasible variable assignment, both sets of constraints impose the same restrictions. There are four cases for each $i \in N$ and $j \in \mathcal{T}$. (i) Assume that $x^i_{j-1} = x^i_j = 0$. It follows from constraints (1c) and (1d) that $y^i_j = \bar{y}^i_j = 0$. In this case, the right-hand-side (RHS) of both inequalities (1f) and (4a), as well as (1g) and (4b) reduce to zero. (ii) Assume that $x^i_{j-1} = 0$ and $x^i_j = 1$. It follows from constraint (1c) that $y^i_j = 1$ and $\bar{y}^i_j = 0$. In this case, the RHS of both constraints (1f) and (4a) reduce to SU^i. Further, inequality (1h) implies that $p^i_{j-1} = 0$ as $x^i_{j-1} = 0$. As a result, both constraints (1g) and (4b) become redundant in the description of $\mathcal{E}$ and $\mathcal{G}$, since their LHS is a non-positive quantity $-p^i_j \leq 0$ and their RHS are non-negative values as $RD^i \geq 0$ and $RD^i - SD^i \geq 0$ by assumption. (iii) Assume that $x^i_{j-1} = 1$ and $x^i_j = 0$. It follows from constraint (1c) that $y^i_j = 0$ and $\bar{y}^i_j = 1$. In this case, the RHS of both constraints (1g) and (4b) reduce to SD^i. Further, inequality (1h) implies that $p^i_j = 0$ as $x^i_j = 0$. As a result, both constraints (1f) and (4a) become redundant in the description of $\mathcal{E}$ and $\mathcal{G}$, since their LHS is a non-positive quantity $-p^i_{j-1} \leq 0$ and their RHS are non-negative values as $RU^i \geq 0$ and $RU^i - SU^i \geq 0$ by assumption. (iv) Assume that $x^i_{j-1} = x^i_j = 1$. It follows from constraints (1c) and (1e) that $y^i_j = \bar{y}^i_j = 0$. In this case, the RHS of both (1f) and (4a) reduce to RU^i, and the RHS of both (1g) and (4b) reduce to RD^i. □

Appendix B: Comparison of DD-BD Method with the Literature

In the literature, DDs have been used in conjunction with BD framework in different contexts. In this section, we give a detailed comparison between DD-BD approach of Section 3 and those used in the literature. In particular, we show that our framework generalizes other existing approaches, while addressing their shortcomings.

van der Linden (2017) proposes a BD framework for linear mixed integer programs that uses a similar approach to ours in defining the master problem over integer variables, and solving the subproblems for continuous variables. A DD is constructed to represent the solution set of the master problem, and is successively refined through cuts obtained from the subproblems. There is, however, a fundamental difference between our approach and that of van der Linden (2017) as they use separation for the feasibility cuts only since their proposed DD cannot contain any continuous variable. To incorporate the optimality cuts, the authors propose a so-called "cost-tuple" approach where, for each optimality cut, a cost parameter is calculated at each node of the DD to record the contribution of the variable assignments in relation to that cut. Such an approach requires the underlying DD to be *non-reduced*, limiting its practicality. In contrast, our proposed DD-BD approach provides a unified framework that treats both feasibility and optimality cuts similarly through separation as the continuous variable is directly incorporated in the DD layers. To elaborate, we first give a brief review of the cost-tuple approach used in van der Linden (2017) to handle optimality cuts.

The cost-tuple approach is given in Algorithm 3 where, without loss of generality, the master problem of a MIP is defined as $\max\{z \,|\, \boldsymbol{x} \in \mathcal{P}\}$ with a bounded $\mathcal{P} \subseteq \mathbb{Z}^n$. Consider $\mathcal{D} = (\mathcal{U}, \mathcal{A}, l(\cdot))$ to be the DD representing $\mathcal{P}$, with a property that each node (except the terminal) has a unique incoming arc, and the terminal receives a unique arc from each node of the previous layer. Let inequalities of the form $z \leq \boldsymbol{\alpha}^j \boldsymbol{x} + \alpha_0^j$, for $j \in J$, represent the set of added optimality cuts at the current iteration of the BD algorithm. For each node $u \in \mathcal{U}$, parameter $r^j(u)$ records the reward (cost, for a minimization variant) contribution of variable assignments at that node to the right-hand-side value of cut $z \leq \boldsymbol{\alpha}^j \boldsymbol{x} + \alpha_0^j$ for $j \in J$. In this algorithm, the accumulated reward $r(u)$ captures the contribution amount at the intersection of optimality cuts for each unique value assignment of variables, while the final reward r^* computes the optimal value of $\max\{z \,|\, \boldsymbol{x} \in \mathcal{P}, z \leq \boldsymbol{\alpha}^j \boldsymbol{x} + \alpha_0^j, \forall j \in J\}$.

The main shortcoming of Algorithm 3 is due to the assumption that the DD nodes have a unique incoming arcs. This requirement leads to an exponential growth of the DD size to distinguish between each r-t path. As a consequence, it is mentioned in van der Linden (2017) that the cost-tuple algorithm is not applicable to reduced DDs. Reduction in DDs is referred to the operation of merging nodes that share a similar subtree; hence it is regarded as a crucial component in building efficient DDs for practical applications and keeping them within a width limit. The DD-BD approach proposed in the present paper, on the other hand, can be applied to reduced DDs, and the separation of merged nodes, if needed, will be performed naturally through the separation operation. We illustrate this difference in the following example.

Algorithm 3: The cost-tuple approach of van der Linden (2017) to handle optimality cuts

Data: a DD $\mathcal{D} = (\mathcal{U}, \mathcal{A}, l(.))$ and a set of optimality cuts $z \leq \boldsymbol{\alpha}^j \boldsymbol{x} + \alpha_0^j$, for $j \in J$

initialize the root node reward as $r^j(s) = \alpha_0^j$ for $j \in J$

forall *cuts* $j \in J$, *node layers* $i \in N \setminus \{n\}$, *and node pairs* $(u, v) \in \mathcal{U}_i \times \mathcal{U}_{i+1}$ **do**

  compute $r^j(v) = r^j(u) + \alpha_i^j l_a$ where $a \in \mathcal{A}_i$ is the unique arc connecting u to v

forall *nodes* $u \in \mathcal{U}_n$ **do**

  compute accumulated reward $r(u) = \min_{j \in J}\{r^j(u) + \alpha_n^j l_a\}$ where $a \in \mathcal{A}_n$ is the unique arc connecting u to the terminal t

compute the final reward $r^* = \max_{u \in \mathcal{U}_n} r_u$

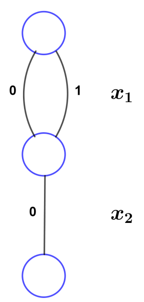
Example 5. Consider the final iteration of a BD algorithm with a master problem $\max\{z \mid \boldsymbol{x} \in \mathcal{P}\}$ where $\mathcal{P} \subseteq \mathbb{Z}^n$ is represented by DD $\mathcal{D}_1$ of Figure 8a. Assume that the following two optimality cuts are added

$$z \leq 3x_1 + 2x_2 \tag{9a}$$

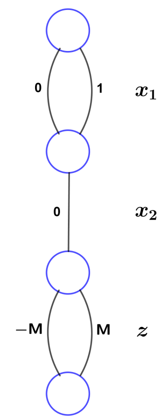
$$z \leq -3x_1 - 5x_2 + 3 \tag{9b}$$

It is clear that both points $(0,0)$ and $(1,0)$ give an optimal solution with optimal value $z^* = 0$. Since $\mathcal{D}_1$ is a reduced DD, the approach of van der Linden (2017) cannot be directly applied to obtain the optimal solution. It is suggested in that work that, when merging occurs, the component-wise maximum of the reward tuple for merging nodes must be selected as the reward tuple of the merged node. This operation guarantees that the final reward value is an upper-bound for the optimal reward value. Applying this operation here, however, creates an infinite loop without converging to a solution, as demonstrated next. We record the reward value for inequalities (9a) and (9b), respectively, in a tuple $(r^1(u), r^2(u))$ for each node u. For the root node, we have the reward tuple $(0,3)$ according to Algorithm 3. For the path including the 1-arc at the first layer, the reward of the end-node is $(3,0)$. Similarly, the path including the 0-arc at the first layer has the end-node reward $(0,3)$. The component-wise maximum yields the reward of $(3,3)$ for the merged node in the middle. The reward at the terminal node is $(3,3)$, giving the optimal value $\bar{z} = 3$. While $\bar{z}$ is an upper-bound for z^*, it can already be separated by the current optimality cuts (9a) and (9b) for both feasible solutions. As a result, the BD approach cannot refine it any further and is stuck at this iteration.

For the DD-BD approach of Algorithm 1, we represent the master problem as $\mathcal{M} = \max\{z \,|\, \boldsymbol{x} \in \mathcal{P}, z \in [-M, M]\}$ where valid bounds are imposed on z variable. A restricted DD $\mathcal{D}_2$ can be created for $\mathcal{M}$ using the equivalence class argument of Section 2.3 as given in Figure 8b. Next, we may simply refine the DD through a sequential separation with respect to optimality cuts (9a) and (9b), as shown in Figures 9a and 9b. It is easy to verify that a longest path in the resulting DD gives either point $(0,0)$ or $(1,0)$ with the optimal value $z^* = 0$. Since these solutions give the optimal value of the problem, Algorithm 1 is terminated due to bound pruning conditions in line 17.

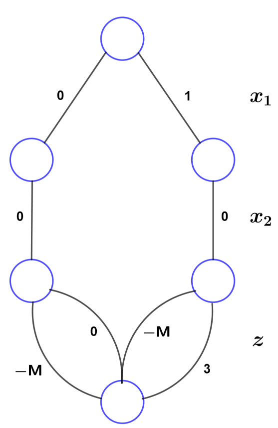


(a) $\mathcal{D}_1$ without continuous variable

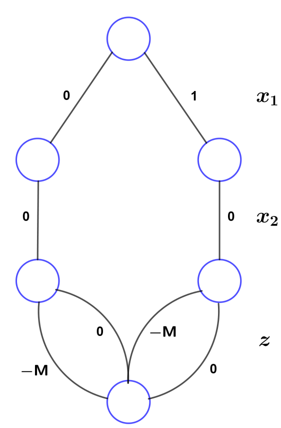


(b) $\mathcal{D}_2$ with continuous variable

Figure 8 DD representation of the master problem of Example 5



(a) Separation wrt (9a)



(b) Separation wrt (9b)

Figure 9 Separation of optimality cuts for the master problem of Example 5

We conclude this section by showing that the DD-BD approach of Algorithm 1 is a generalization of the cost-tuple approach of Algorithm 3, since the latter can be viewed as a special case of the former when the underlying DD is not reduced.

PROPOSITION 5. *Consider the master problem* $\max\{z \,|\, \boldsymbol{x} \in \mathcal{P}\}$ *of a MIP with a bounded* $\mathcal{P} \subseteq \mathbb{Z}^n$*, and let* $\mathcal{D} = (\mathcal{U}, \mathcal{A}, l(\cdot))$ *be the DD representing* $\mathcal{P}$*, with a property that each node (except the terminal) has a unique incoming arc, and the terminal receives a unique arc from each node of the previous layer. Let inequalities of the form* $z \le \boldsymbol{\alpha}^j \boldsymbol{x} + \alpha_0^j$*, for* $j \in J$*, represent the set of optimality cuts to be added at the current iteration of the BD algorithm. Define* $\bar{\mathcal{D}} = (\bar{\mathcal{U}}, \bar{\mathcal{A}}, \bar{l}(\cdot))$ *to be the DD constructed from* $\mathcal{D}$ *by adding a layer representing* z *variable as the last arc layer. Then, the reward value obtained from Algorithm 3 applied to* $\mathcal{D}$ *is equal to the optimal value obtained from Algorithm 1 when applied to* $\bar{\mathcal{D}}$*.*

Proof. We first describe the relation between $\mathcal{D}$ and $\bar{\mathcal{D}}$. All nodes and arcs from layer one to layer $n-1$ are similar in both DDs, i.e., $\mathcal{A}_i = \bar{\mathcal{A}}_i$ for $i \in N \setminus \{n\}$ and $\mathcal{U}_i = \bar{\mathcal{U}}_i$ for $i \in N$. The difference is that $\bar{\mathcal{D}}$ has an extra layer that represents z variable. This layer is created as follows. For each node $u \in \bar{\mathcal{U}}_n$, we create a node $v \in \bar{\mathcal{U}}_{n+1}$, and connect them with an arc $a \in \bar{\mathcal{A}}_n$ with the same label as the unique arc connecting the replica of node u in $\mathcal{U}_n$ to the terminal node of $\mathcal{D}$. Then, for each node $v \in \bar{\mathcal{U}}_{n+1}$, we create two arcs with labels $-M$ and M in layer $\bar{\mathcal{A}}_{n+1}$ that connect v to the terminal node in $\bar{\mathcal{D}}$. These label values represent some valid lower and upper bounds for variable z. According to Algorithm 1, $\bar{\mathcal{D}}$ is refined with respect to the optimality cuts, and then the longest path is found over the refined DD weighted by the objective function rates. To perform the refinement, the solution set satisfying each optimality cut is modeled by a DD called *threshold diagram*; see Bergman et al. (2016) for an exposure to threshold diagrams. The last layer of such DDs corresponds to variable z and contains two arcs with labels representing a lower and upper bound for z. Since for each node $v \in \bar{\mathcal{U}}_{n+1}$, there exists a unique arc-specified path $\mathrm{P} = (a_1, \dots, a_n) \in \bar{\mathcal{A}}_1 \times \ldots \times \bar{\mathcal{A}}_n$ from the root node to v, it follows from the assumption that the threshold diagrams have the same structure as $\bar{\mathcal{D}}$, with the difference that the upper bound label on arcs connecting v to the terminal is computed as $M^j(v) = \sum_{i=1}^n \alpha_i^j l_{a_i} + \alpha_0^j$ for each optimality cut $j \in J$. As a result, refining $\bar{\mathcal{D}}$ with respect to the optimality cuts reduces to intersecting label value intervals associated with arcs at the last layer, which are of the form $[-M, M^j(v)]$ for each node $v \in \bar{\mathcal{U}}_{n+1}$. This intersection yields the lower bound label $-M$, and the upper bound label $M(v) = \min_{j \in J} M^j(v)$ for pair of arcs connected to v in the last layer of the refined DD. Note that $M(v)$ is equal to $r(v)$ as computed in Algorithm 3. Finally, to find the longest path on the refined $\bar{\mathcal{D}}$, all arcs in layers 1–n are assigned zero weight and arcs in the last layer are assigned weight 1 deduced from the objective function of the master problem. Therefore, the length of the longest r-t path in $\bar{\mathcal{D}}$ is computed as $w^* = \max_{v \in \bar{\mathcal{U}}_{n+1}} M(v)$, which is equal to r^* obtained from Algorithm 3. Since $\bar{\mathcal{D}}$ represents an exact DD for the master problem and has been refined with respect to optimality cuts, w^* is returned as the output of Algorithm 1. $\square$