Electronic Companion

EC.1. Remarks on The PIDS-PP Problem on Trees

The study of the PIDS-PP problem on trees is important for several reasons. First, the information propagation paths on social networks often resemble trees, as shown by Liben-Nowell and Kleinberg (2008) on Internet chain-letter data and by Kwak et al. (2010) on Twitter data. Second, from the perspective of the decomposition, Adcock et al. (2013) empirically demonstrate real social and information networks possess large-scale tree-like structure. Third, tree graphs are commonly used in stylized models on social networks to generate insights and obtain a better understanding of the underlying problem (see Jackson et al. 2020). Fourth, deriving tight and strong models on special cases can be considered as a starting point for obtaining stronger models on the general case for many combinatorial optimization problems (see Goemans 1994, Magnanti and Wolsey 1995). Our interest in trees is due to the fact that trees are the structurally simplest non-trivial graphs. Thus, a high-level idea is to derive useful results and insights from special cases which are polynomial solvable, and then apply them to the general case which is NP-hard. The model we derive for the PIDS-PP problem on trees plays a crucial role in developing our model on arbitrary graphs. The linear-time algorithm plays an important role in proving the tightness of MIP2 on trees. The rest of this electronic companion is organized as follows. Section EC.2 presents a linear-time dynamic algorithm for the PIDS-PP problem on tree. Section EC.3 proves that MIP2 is indeed the strongest possible formulation for the PIDS-PP problem on trees.

EC.2. Algorithm for The PIDS-PP Problem on Trees

In this section, we present a dynamic programming (DP) algorithm to solve the PIDS-PP problem on trees. The DP algorithm decomposes the problem into subproblems, starting from the leaves of the tree. A subproblem is defined on a star network, which has a single central node and (possibly) multiple child nodes. For each star subproblem, the DP algorithm solves the PIDS-PP problem for two cases. Consider the link that connects the star to the rest of the tree. We will refer to the node adjacent to the central node on this link as its parent. In the first case, the parent is not selected for direct targeting (for brevity, we will simply say that the node is "not selected"), whereas in the second case, the parent is selected for direct targeting (for brevity, we will simply say that the node is "selected"). Recall that when a node is selected for direct targeting it receives full payment. This process of solving star subproblems for two cases, followed by contraction of the star node, is repeated until we are left with a single star. The last star only requires the solution of one case, where the parent is not selected. After we exhaust all subproblems, a backtracking method is used

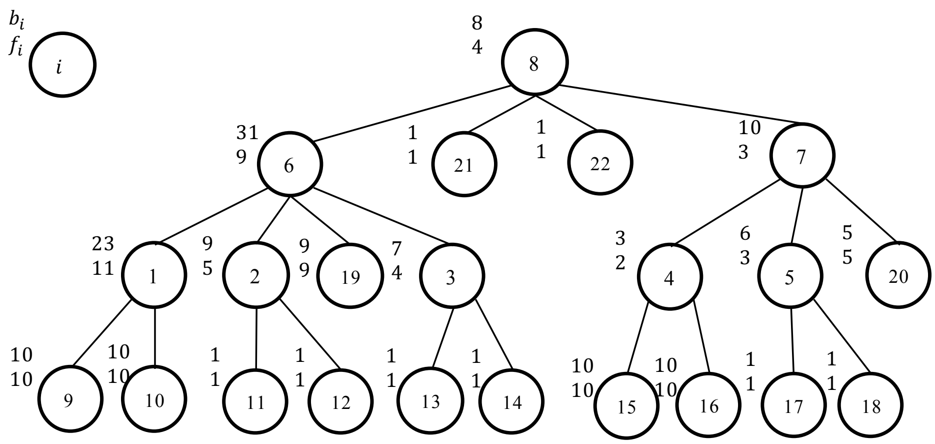


Figure EC.1 A PIDS-PP Problem Instance.

to combine the solution candidates from the star subproblems and identify a final solution (set of nodes selected for direct targeting, and the payment amounts to all nodes on the tree).

Algorithm 2 provides the pseudocode of the proposed algorithm. To create an ordering amongst the subproblems considered in the algorithm, it is convenient (but not necessary) to arbitrarily pick a root node (which we will denote by r). We will then prioritize the subproblems in order of how far their central nodes are from the root node of the tree (i.e., at every step among the remaining subproblems, we consider a subproblem whose central node is farthest from the root node). We call this a bottom-up traversal of the tree (this ordering can easily be determined a priori by conducting a breadth-first search (BFS) from the root node and by considering the non-leaf nodes of the tree in reverse BFS order). The global variable TC has the total cost of the optimal solution.

We now discuss how to solve the PIDS-PP problem on a star. To better illustrate the algorithm, we consider the instance shown in Figure EC.1. Let L denote the set of all leaf nodes in the original tree. Let c denote the central node of a star, all of the other nodes in the star are children of the central node and are denoted by $L(c)$ (notice that $L(c)$ need not be a subset of L), and refer to this star as star c. There are two cases to consider. First, we consider the case, where the parent of the central node c is not selected in the optimal solution. Let X_c^{NPS} represent the set of nodes selected in the solution to star c, let P_c^{NPS} be the payment made to the central node in the solution, and let C_c^{NPS} be the total cost of the solution for star c. Analogously, we consider the case, where the

Algorithm 2 Algorithm for the PIDS-PP problem on trees

```
Arbitrarily pick a node as the root node of the tree and let TC = 0.
Define the order of the star problems based on the bottom-up traversal of the tree
For all i ∈ L (the set of leaf nodes in G), set b_i^1 = b_i, and b_i^2 = 0.
for each star subproblem do
    StarHandling
end for
SolutionBacktrack
```

parent of the central node c is selected in the optimal solution with X_c^{PS} representing the set of nodes selected in the solution to star c, P_c^{PS} being the payment made to the central node in the solution, and C_c^{PS} denoting the total cost of the solution for star c.

In Figure EC.1, node 8 is selected as the root. Following the bottom-up ordering of the tree we would consider stars $1,2,3,4$ and 5 first. Notice that all of these stars have their children in L. We will refer to stars, where all children are members of L as "bottom stars". Notice that because influence can only propagate to a neighbor of a selected node (and no further), if the parent of any node $i \in L$ is not selected, then node i must be selected. This allows for a straightforward calculation to solve a bottom star. Either the central node of the bottom star is selected or all children in the bottom star are selected with an additional (potentially 0) partial payment to the central node. Specifically, for the case, where the parent of the central node c of a bottom star is not selected we compare the cost of b_c against the cost of $(\sum_{j\in L(c)} b_j + \max\{0, b_c - |L(c)|f_c\})$. If b_c is greater, then all of the children in the bottom star are selected and $X_c^{NPS} = L(c)$, $P_c^{NPS} = \max\{0, b_c - |L(c)|f_c\}$, and $C_c^{NPS} = (\sum_{j\in L(c)} b_j + \max\{0, b_c - |L(c)|f_c\})$. Otherwise, the central node c of the bottom star is selected and $X_c^{NPS} = \{c\}$, $P_c^{NPS} = b_c$, and $C_c^{NPS} = b_c$. At the same time, for the case where the parent of the central node c of the bottom star is selected, we compare the cost of b_c against the cost of $(\sum_{j\in L(c)} b_j + \max\{0, b_c - (|L(c)+1)|f_c\})$. If b_c is greater, then all of the children are selected and $X_c^{PS} = L(c)$, $P_c^{PS} = \max\{0, b_c - (|L(c)+1)|f_c\}$, and $C_c^{PS} = (\sum_{j\in L(c)} b_j + \max\{0, b_c - (|L(c)+1)|f_c\})$. Otherwise, the central node c is selected and $X_c^{PS} = \{c\}$, $P_c^{PS} = b_c$, and $C_c^{PS} = b_c$.

To illustrate, consider star 1. When node 1's parent (node 6) is not selected in the optimal solution, we compare the cost of selecting node 1 with a payment of 23 against the cost of selecting all of the leaf nodes of star 1 (i.e., nodes 9 and 10) at a cost of 20 units and a partial payment amounting to 1 unit to node 1 (for a total cost of 21). Thus, the solution is $X_1^{NPS} = \{9, 10\}$, $P_1^{NPS} = 1$, and $C_1^{NPS} = 21$. Now consider the case, where node 1's parent (node 6) is selected. In this case, we compare the cost of selecting the central node 1 (with a payment of 23) against selecting all of the leaf nodes of the star (i.e., nodes 9 and 10) with no additional partial payment to the central node (for a total cost of 20). Thus, the solution is $X_1^{PS} = \{9, 10\}$, $P_1^{PS} = 0$, and $C_1^{PS} = 20$. Similarly, for the bottom stars 2, 3, 4, and 5, we find: In star 2, $X_2^{NPS} = X_2^{PS} = \{11, 12\}$, $P_2^{NPS} = P_2^{PS} = 0$, and $C_2^{NPS} = C_2^{PS} = 2$. In star 3, $X_3^{NPS} = X_3^{PS} = \{13, 14\}$, $P_3^{NPS} = P_3^{PS} = 0$, and $C_3^{NPS} = C_3^{PS} = 2$. In star 4, $X_4^{NPS} = X_4^{PS} = \{4\}$, $P_4^{NPS} = P_4^{PS} = 3$, and $C_4^{NPS} = C_4^{PS} = 3$. In star 5, $X_5^{NPS} = X_5^{PS} = \{17, 18\}$, $P_5^{NPS} = P_5^{PS} = 0$, and $C_5^{NPS} = C_5^{PS} = 2$.

Next, once a star's solution candidates are determined, the star is contracted into a single child node for its parent's star subproblem. It may appear that we have considered all possible solution candidates X_c^{NPS} and X_c^{PS} for a given star c in the optimal solution. However, that is not necessarily the case. Consider star 1. Here $X_1^{PS} = X_1^{NPS} = \{9, 10\}$. In both cases, the leaf nodes 9 and 10 are

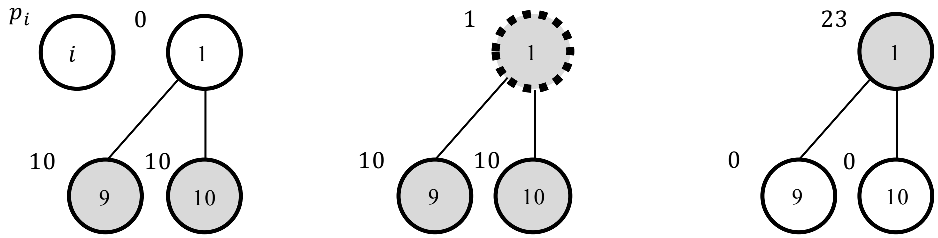


Figure EC.2 $(a) X_1^{PS} = \{9,10\}, P_1^{PS} = 0, C_1^{PS} = 20$, $(b) X_1^{NPS} = \{9,10\}, P_1^{NPS} = 1, C_1^{NPS} = 21$, (c) selected node 1 and $b_1 = 23$. **Possible Solutions for Star 1 in the Optimal Solution to the Problem.**

selected. Although star 1 does not need its central node 1 to be selected in either case, star 6 may need node 1 to be selected in order to activate its central node 6 because influence only propagates to the neighbors of the selected nodes. This may not be captured in the solutions X_c^{NPS} and X_c^{PS} computed so far for a given star. Hence, in addition to C_c^{NPS} and C_c^{PS}, we also use the cost b_c of the solution that selects the central node of star c. Figure EC.2 illustrates the situations for star 1. The nodes receiving payment are shaded. Among them, the ones with solid border are selected (i.e., directly targeted and receiving full payment) and the others with a dotted border receive partial payment. Figure EC.2(a) displays the solution X_1^{PS}, Figure EC.2(b) displays the solution X_1^{NPS}, and Figure EC.2(c) displays the solution, where the central node 1 is selected. Observe that the costs of the three solutions satisfy $C_c^{PS} \le C_c^{NPS} \le b_c$. Therefore, we must incur a cost of at least C_c^{PS} for star c in the optimal solution. This amount is added to the total cost TC. The remaining incremental amounts $b_c^1 = C_c^{NPS} - C_c^{PS}$ and $b_c^2 = b_c - C_c^{NPS}$ are computed and are used to solve the next star subproblem.

Unlike bottom stars, we have (e.g., star 6) stars containing both contracted stars as leaf nodes (nodes 1, 2, and 3), as well as the leaf nodes in L (node 19). Therefore, we compute $b_i^1 = b_i$, and $b_i^2 = 0$ for leaf nodes $i \in L$. Consider node 19 to explain this calculation. If its parent is not selected, node 19 must be selected with a cost of 9. If its parent is selected, node 19 gets influenced and its cost is 0. Thus, $b_{19}^1 = 9$, and $b_{19}^2 = 0$. Figure EC.3 displays stars 6 and 7 after contracting stars 1, 2, 3, 4 and 5, respectively. At this point $TC = 29$ and the calculated values of (b_i^1, b_i^2) are shown for all leaf nodes.

We are now ready to discuss how to solve the PIDS-PP problem on a star (earlier our discussion was limited to solving the problem on the bottom stars; the ensuing discussion applies to all stars). Consider a star c and the case, where the parent of node c is not selected in the optimal solution. We have two alternatives. Either we select the central node c with cost b_c to influence the entire star (if the central node c is selected then all children $i \in L(c)$ follow the solution X_i^{PS} whose cost is already included in TC) or we select a subset of nodes in $L(c)$ as cheaply as possible that influences the entire star (the nodes i in $L(c)$ that are not selected would then follow the solution X_i^{NPS}).

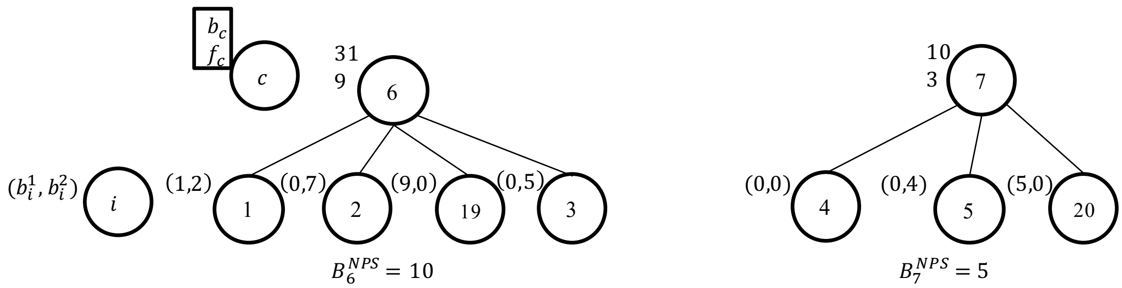


Figure EC.3 **(a) Star 6 after Contracting Stars 1, 2, and 3.** **(b) Star 7 after Contracting Stars 4 and 5.**

We need to compute the cost of the alternative, where a minimum cost subset of nodes in $L(c)$ are selected to influence the entire star. If the central node c of the star is not selected, we must at least incur the cost $B_c^{NPS} = \sum_{i \in L(c)} b_i^1$, given that all of the children $i \in L(c)$ must at least incur the cost of C_i^{NPS} when their parent is not selected. The influence factor f_c of the central node c is used as a filter. Any child $i \in L(c)$ with $b_i^2 \geq f_c$ will never be selected. When $b_i^2 \geq f_c$, paying b_i^2 units to a child node i sends f_c influence to node c, thus, the decrease in the threshold of the central node is less than what we spend. We could be better off by paying the central node directly and using the X_i^{NPS} solution for such children. Therefore, we collect the nodes $i \in L(c)$ with $b_i^2 < f_c$ in set S_c and sort them in ascending order of their b_i^2 values. The set $S_c^0 \subseteq S_c$ denotes the set of nodes $i \in S_c$ with $b_i^2 = 0$ (i.e., these nodes are essentially free to select). The cost of the solution depends on the size of the set S_c. Let $g_c = \lceil \frac{b_c}{f_c} \rceil$ and $l_c = b_c - (g_c - 1) f_c$.

Case 1 ($|S_c| < g_c$): All nodes in the set S_c are selected, and the payment to the central node c is determined as $l_c + f_c(g_c - 1 - |S_c|)$ for a total cost of $B_c^{NPS} + l_c + f_c(g_c - 1 - |S_c|) + \sum_{i \in S_c} b_i^2$.

Case 2 ($|S_c| \geq g_c$ and $|S_c^0| \geq g_c$): All nodes in the set S_c^0 are selected, and no additional payment to the central node is necessary, resulting in a total cost of B_c^{NPS}.

Case 3 ($|S_c| \geq g_c$ and $|S_c^0| < g_c$): We select the first g_c nodes in S_c in ascending order of their b_i^2 value (we denote this set as S_{g_c}) if the threshold for the g_c-th node is less than or equal to l_c for a total cost of $B_c^{NPS} + \sum_{i \in S_{g_c}} b_i^2$. Otherwise, we select the first $(g_c - 1)$ nodes in S_c in ascending order of their b_i^2 values (we denote this set as $S_{g_c - 1}$) and pay l_c to node c, resulting in a total cost of $B_c^{NPS} + l_c + \sum_{i \in S_{g_c - 1}} b_i^2$.

Comparing b_c, the cost of selecting the central node c against the cost of the solution just obtained from one of the three above cases (and selecting the one with a lower cost) provides us with the solution to the PIDS-PP problem on the given star.

Now, we consider star c and the case, where the parent of node c is selected in the optimal solution. Again, we have two alternatives. Either we select the central node c with cost b_c to influence the entire star, or we select a subset of nodes in $L(c)$ as cheaply as possible that influences the entire star. The cost of the alternative, where a minimum cost subset of the nodes in $L(c)$

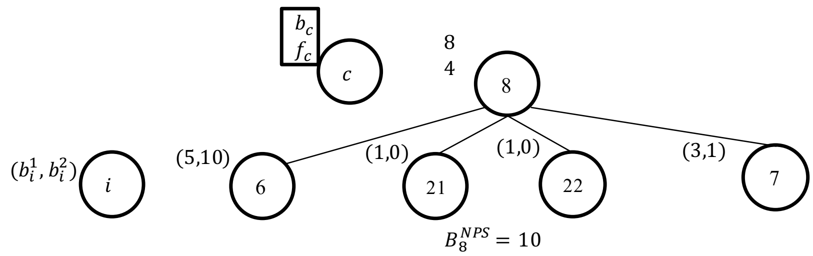


Figure EC.4 The Last Star after Contracting Stars 6 and 7.

are selected to influence the entire star, is calculated identically as the above three cases with the change that g_c is updated to $g_c - 1$ (to account for the fact that star c's parent has been selected), and is thus able to influence it.

Algorithm 3 provides the pseudocode associated with this calculation procedure. At its core is the function SOLVESTAR, which finds the optimal solution for a given star. When the procedure is applied to star 6, we start with $S_6 = \{19, 1, 3, 2\}$, $S_6^0 = \{19\}$ and $B_6^{NPS} = 10$. We get $X_6^{NPS} = \{19, 1, 3\}$, $P_6^{NPS} = 4$, and $C_6^{NPS} = B_6^{NPS} + P_6^{NPS} + \sum_{i \in X_6^{NPS}} b_i^2 = 21$; and $X_6^{PS} = \{19, 1\}$, $P_6^{PS} = 4$, and $C_6^{PS} = B_6^{NPS} + P_6^{PS} + \sum_{i \in X_6^{PS}} b_i^2 = 16$. Contracting star 6 gives $b_6^1 = 5$ and $b_6^2 = 10$. Similarly for star 7, we start with $S_7 = \{4, 20\}$, $S_7^0 = \{4, 20\}$ and $B_7^{NPS} = 5$ (notice that node 5 is not in S_7 because $b_5^2 = 4 > f_7 = 3$). We get $X_7^{NPS} = \{4, 20\}$, $P_7^{NPS} = 4$ and $C_7^{NPS} = 9$; and $X_7^{PS} = \{4, 20\}$, $P_7^{PS} = 1$, and $C_7^{PS} = 6$. Contracting star 7 gives $b_7^1 = 3$ and $b_7^2 = 1$. Now, TC is 51. After contracting stars 6 and 7, we only have one star left, as shown in Figure EC.4. Here, $S_8 = \{21, 22, 7\}$, $S_8^0 = \{21, 22\}$ and $B_8^{NFS} = 10$. We find that it is cheapest to select central node 8 (because its cost of 8 is less than $B_8^{NFS} = 10$) than any of its children. Hence, $X_8^{NPS} = \{8\}$, $P_8^{NPS} = 8$ and $C_8^{NPS} = 8$. Thus, TC is 59.

After we obtain the solution of the last star, which has the root node as its central node, we invoke a backtracking procedure to choose the solution from the candidates for each star subproblem and piece them together to obtain the final solution for this tree. Once the last star subproblem is solved, for each child node in this star, we know if it is selected or not and if its parent node is selected or not. For instance, if the central node is selected, all stars with the central node in $\{L(c) \setminus L\}$ will pick the Parent-Selected candidate. Otherwise, first, if a node i in $\{L(c) \setminus L\}$ is selected, we can proceed to the nodes in $L(i)$ and pick the Parent-Selected candidate. Second, if a node i in $\{L(c) \setminus L\}$ is not selected, star i will pick the NoParent-Selected candidate. With this information, we can now proceed down the tree, incorporating the solution candidate at a node based on the solution of its parent star. This backtracking procedure is described in Algorithm 4 SolutionBacktrack. Let r denote the root of the tree (as determined by Algorithm 2), $\mathbf{x}^*$ be a binary vector indicating the nodes selected with a value of 1s, and $\mathbf{p}^*$ be the vector of payment. Note that this $\mathbf{p}^*$ payment vector is slightly different from the p variables in the MIP formulations, as it

Algorithm 3 StarHandling

Input: star c.

$(X_c^{NPS}, P_c^{NPS}, C_c^{NPS}) \leftarrow$ SolveStar(star c, NoParent-Selected).
if star c is the last star **then**
  $TC = TC + C_c^{NPS}$
else
  $(X_c^{PS}, P_c^{PS}, C_c^{PS}) \leftarrow$ SolveStar(star c, Parent-Selected).
  The contracted node has $b_c^1 = C_c^{NPS} - C_c^{PS}$ and $b_c^2 = b_c - C_c^{NPS}$.
  $TC = TC + C_c^{PS}$
end if
function SolveStar(a star c, Flag)
  $B_c^{NPS} = \sum_{i \in L(c)} b_i^1$, $g_c = \left\lceil \frac{b_c}{f_c} \right\rceil$, $l_c = b_c - (g_c - 1) f_c$, $S_c = \{i \in L(c) : b_i^2 < f_c\}$ and $S_c^0 = \{i \in L(c) : b_i^2 = 0\}$.
  if Flag == Parent-Selected **then**
    $g_c = g_c - 1$.
  end if
  if $|S_c| < g_c$ **then**
    $C = \min\left\{b_c,\ B_c^{NPS} + \sum_{i \in S_c} b_i^2 + l_c + f_c(g_c - 1 - |S_c|)\right\}$
  else
    if $|S_c^0| \geq g_c$ **then**
      $C = \min\{b_c,\ B_c^{NPS}\}$
    else
      Let S_{g_c} and S_{g_c-1} be the sets of the cheapest g_c and $(g_c - 1)$ nodes in S_c by b^2, respectively.
      $C = \min\left\{b_c,\ B_c^{NPS} + \sum_{i \in S_{g_c}} b_i^2,\ B_c^{NPS} + \sum_{i \in S_{g_c-1}} b_i^2 + l_c\right\}$
    end if
  end if
  $X \leftarrow c$ and $P = b_c$ **if** C is b_c.
  $X \leftarrow S_c$ and $P = l_c + f_c(g_c - 1 - |S_c|)$ **if** C is $B_c^{NPS} + \sum_{i \in S_c} b_i^2 + l_c + f_c(g_c - 1 - |S_c|)$.
  $X \leftarrow S_c^0$ and $P = 0$ **if** C is B_c^{NPS}.
  $X \leftarrow S_{g_c}$ and $P = 0$ **if** C is $B_c^{NPS} + \sum_{i \in S_{g_c}} b_i^2$.
  $X \leftarrow S_{g_c-1}$ and $P = l_c$ **if** C is $B_c^{NPS} + \sum_{i \in S_{g_c-1}} b_i^2 + l_c$.
  return X, P, C.
end function

includes both full and partial payments. Figure EC.5 shows the final solution. The nodes receiving payment are shaded. Among them, the ones with a solid border are selected (i.e., directly targeted and receiving full payment), and the others with a dotted border receive partial payment.

Proof of Proposition 2: The correctness of the algorithm can be established via induction, using identical arguments to the preceding discussion. We now discuss the running time. There are at most $|V|$ stars. For each star c, we need to find g_c cheapest children that takes $O(deg(c))$ time. Finding the g_cth order statistics can be done in $O(deg(c))$ time by the Quickselect method in Chapter 9 of Cormen et al. (2009); thus, it takes $O(deg(c))$ time to go through the list to collect the g_c cheapest normal children. For the whole tree, this is bounded by $O(|V|)$ time. In the backtracking procedure, we pick the final solution for each node, which takes $O(|V|)$ time over the tree. Therefore, the running time for the dynamic algorithm is linear on the number of nodes. □

Algorithm 4 SolutionBacktrack

Let $\mathbf{x}^* = \mathbf{0}$ and $\mathbf{p}^* = \mathbf{0}$. Then, call PIECING($r$, $\mathbf{x}^*$, $\mathbf{p}^*$, NoParent-Selected) for the root node r.
function PIECING(c, $\mathbf{x}^*$, $\mathbf{p}^*$, $Flag$)
  if $Flag =$ Parent-Selected **then**
    $X' \leftarrow X_c^{PS}$, $x_i = 1\, \forall i \in X'$, and $P' = P_c^{PS}$.
  else
    $X' \leftarrow X_c^{NPS}$, $x_i = 1\, \forall i \in X'$, and $P' = P_c^{NPS}$.
  end if
  if $c \in X'$ **then**
    $\forall i \in \{L(c) \setminus L\}$ call PIECING(i, $\mathbf{x}^*$, $\mathbf{p}^*$, Parent-Selected).
  else
    $p_c^* = P'$ and $\forall i \in \{L(c) \setminus (X' \cup L)\}$ call PIECING(i, $\mathbf{x}^*$, $\mathbf{p}^*$, NoParent-Selected).
    for $i \in X'$ **do**
      $\forall j \in \{L(i) \setminus L\}$ call PIECING(j, $\mathbf{x}^*$, $\mathbf{p}^*$, Parent-Selected).
    end for
  end if
  return $\mathbf{x}^*$ and $\mathbf{p}^*$.
end function

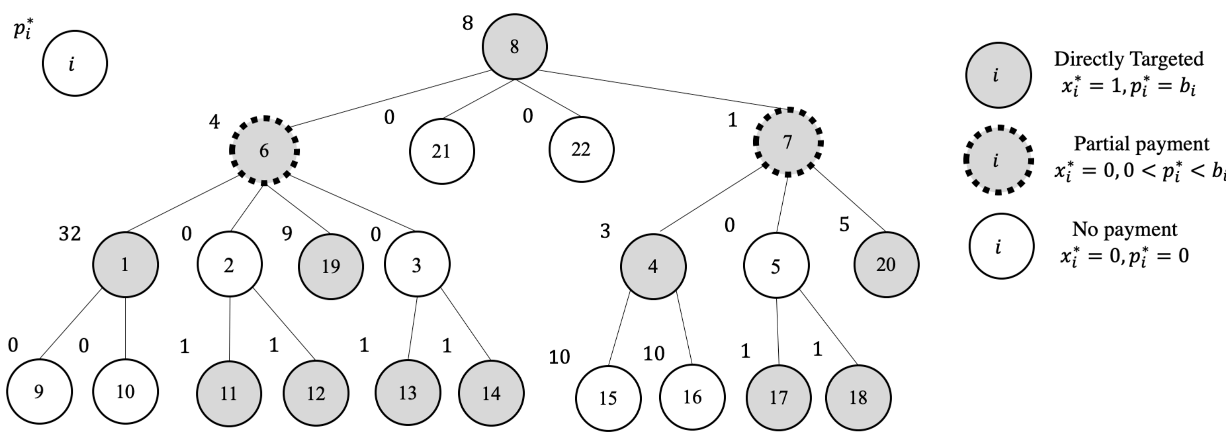


Figure EC.5 The Solution Obtained by Our DP Algorithm. Nodes 1, 4, 7, 8 11, 12, 13, 14 17, 18, 19, and 20 are Selected (i.e., Directly Targeted and Receive Full Payment). Nodes 6 and 7 Receive Partial Payment.

EC.3. Proof of Theorem 1: MIP2 is Tight for The PIDS-PP Problem on Trees

The linear relaxation of MIP2 is the following LP problem:

$$\text{(LP2)} \quad \text{Min} \quad \sum_{i \in V} p_i + \sum_{i \in V} b_i x_i \tag{EC.1}$$

$$\text{Subject to } (s_{id}) \quad -y_{id} - y_{di} = -1, \quad \forall \{i,d\} \in E_t, \tag{EC.2}$$

$$(t_{ij}) \quad x_i - y_{dj} \geq 0, \quad \forall i \in V, j \in n(i), \tag{EC.3}$$

$$(u_{id}) \quad y_{id} - x_i \geq 0, \quad \forall i \in V, d \in a(i), \tag{EC.4}$$

$$(v_i) \quad p_i + b_i x_i + \sum_{d \in a(i)} f_i y_{di} \geq b_i, \quad \forall i \in V, \tag{EC.5}$$

$$(w_i) \quad p_i + l_i g_i x_i + \sum_{d \in a(i)} l_i y_{di} \geq l_i g_i, \quad \forall i \in V, \tag{EC.6}$$

$$p_i,\, x_i \geq 0, \qquad \forall i \in V, \tag{EC.7}$$

$$y_{id},\, y_{di} \geq 0, \qquad \forall \{i,d\} \in E_t. \tag{EC.8}$$

We refer to this LP problem as LP2. We have s_{id}, t_{ij}, u_{id}, v_i, and w_i as dual variables for the constraints (EC.2), (EC.3), (EC.4), (EC.5), and (EC.6) respectively. The dual to LP2, which is referred to as DLP2 is as follows:

$$\text{(DLP2) Max} \quad \sum_{i\in V} b_i v_i + \sum_{i\in V} l_i g_i w_i - \sum_{\{i,d\}\in E_t} s_{id} \tag{EC.9}$$

$$\text{Subject to } (y_{id}) \qquad -s_{id} + u_{id} \leq 0, \qquad \forall i \in V, d \in a(i), \tag{EC.10}$$

$$(y_{di}) \qquad -s_{id} - t_{ji} + f_i v_i + l_i w_i \leq 0, \qquad \forall d \in D, i \in a(d), \tag{EC.11}$$

$$(x_i) \quad \sum_{j\in n(i)} t_{ij} - \sum_{d\in a(i)} u_{id} + b_i v_i + l_i g_i w_i \leq b_i, \quad \forall i \in V, \tag{EC.12}$$

$$(p_i) \qquad v_i + w_i \leq 1, \qquad \forall i \in V, \tag{EC.13}$$

$$s_{id} \text{ free}, \qquad \forall \{i,d\} \in E_t, \tag{EC.14}$$

$$t_{ij} \geq 0, \qquad \forall i \in V, j \in n(i), \tag{EC.15}$$

$$u_{id} \geq 0, \qquad \forall i \in V, d \in a(i), \tag{EC.16}$$

$$v_i,\, w_i \geq 0, \qquad \forall i \in V. \tag{EC.17}$$

First, based on this pair of primal and dual problems, we have the complementary slackness (CS) conditions:

$$(-s_{id} + u_{id}) y_{id} = 0, \qquad \forall i \in V, d \in a(i), \tag{EC.18}$$

$$(x_i - y_{dj}) t_{ij} = 0, \qquad \forall i \in V, j \in n(i), \tag{EC.19}$$

$$(y_{id} - x_i) u_{id} = 0, \qquad \forall i \in V, d \in a(i), \tag{EC.20}$$

$$(p_i + b_i x_i + \sum_{d\in a(i)} f_i y_{di} - b_i) v_i = 0, \qquad \forall i \in V, \tag{EC.21}$$

$$(p_i + l_i g_i x_i + \sum_{d\in a(i)} l_i y_{di} - l_i g_i) w_i = 0, \qquad \forall i \in V, \tag{EC.22}$$

$$(-s_{id} - t_{ji} + f_i v_i + l_i w_i) y_{di} = 0, \qquad \forall d \in D, i \in a(d), \tag{EC.23}$$

$$(\sum_{j\in n(i)} t_{ij} - \sum_{d\in a(i)} u_{id} + b_i v_i + l_i g_i w_i - b_i) x_i = 0, \quad \forall i \in V, \tag{EC.24}$$

$$(v_i + w_i - 1) p_i = 0, \qquad \forall i \in V. \tag{EC.25}$$

The solution vector $\mathbf{x}^*$, $\mathbf{p}^*$ obtained in the dynamic programming algorithm (Algorithm 2 in Appendix EC.2) can be transferred into a feasible solution for LP2. For any node i that is selected node set $x_i = 1$ and $p_i = 0$. Then, set $y_{di} = 0$ and $y_{id} = 1$ for all d in $a(i)$. For all other nodes i that are not selected, set $x_i = 0$ and $p_i = p_i^*$. Next, let $S_i^d = \{d \in a(i) \cap a(j) : x_j = 1\ \forall j \in n(i)\}$. For all $d \in S_i^d$, set $y_{di} = 1$ and $y_{id} = 0$. For all $d \in a(i) \setminus S_i^d$, set $y_{di} = 0$ and $y_{id} = 1$.

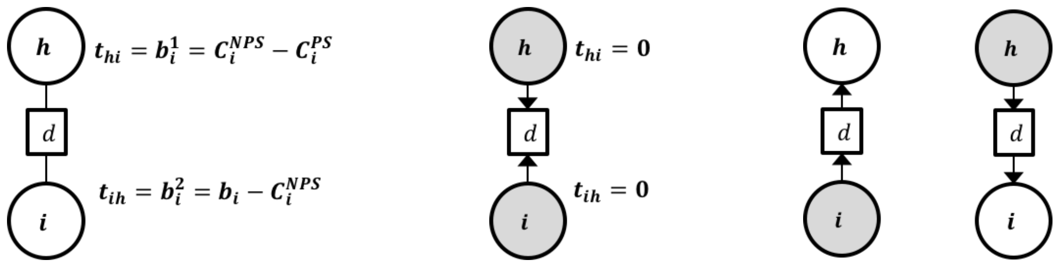


Figure EC.6 (a) **t Variables.** (b) **Condition (EC.19) when** $x_i \neq y_{dh}$**.** (c) **Case 1: A Leaf Node** i $p_i = 0$ **and** $x_i = y_{id}$**.**

With this primal feasible solution in hand, a dual feasible solution will be constructed in the following proof, and we show that the CS conditions are satisfied by this pair of primal and dual solutions. Throughout the proof, we always have $u_{id} = s_{id}$. Thus, the CS condition (EC.18) is satisfied. We can focus on the remaining CS conditions.

In DLP2, only t variables interact between two nodes in V. If their values are fixed, we can isolate each node in V and assign values to s, u, v and w variables. Following the bottom-up order as defined in Algorithm 2, we start with the bottom level of the original tree. Let node i be the current node and node h be its parent node in the original tree G. Recall that b_i^1, b_i^2, S_i, X_i^{NPS}, P_i^{NPS}, X_i^{PS}, and P_i^{PS} are obtained in Algorithm 2. We set $t_{ih} = b_i^2$ and $t_{hi} = b_i^1$, as shown in Figure EC.6(a). For condition (EC.19), it requires $t_{ih} = 0$ when $x_i = 1$ and $y_{dh} = 0$. This means that the corresponding $x_h = 1$. Given that node h and node i are both selected, it implies that $C_i^{PS} = b_i$. Thus, $b_i^1 = b_i^2 = 0$ because $C_i^{PS} \leq C_i^{NPS} \leq b_i$. Thus, $t_{ih} = 0$ and $t_{hi} = 0$, as shown in Figure EC.6(b). For other situations, we have $x_i = y_{dh}$. Thus, condition (EC.19) is satisfied.

Now, three cases are considered to assign associated dual variables for a node i in V. All s, u, v and w variables are initialized as zeros. Then, in the following proof, we only change those variables that need to be non-zeros.

Case 1: Suppose that node i is a leaf node and node h is its parent node in G. When node i is selected, $p_i = 0$, $x_i = 1$, and $y_{id} = 1$. Otherwise, $p_i = 0$, $x_i = 0$ and $y_{id} = 0$. This means that $p_i = 0$, and $x_i = y_{id}$, as shown in Figure EC.6(c). Also, $t_{ih} = 0$ and $t_{hi} = b_i$ because $b_i^1 = b_i$ and $b_i^2 = 0$. Set $v_i = 1$. All primal and dual constraints are binding for conditions (EC.20), (EC.21), (EC.22) (EC.23), (EC.24), and (EC.25) given that $b_i = f_i$. Thus, they are satisfied.

Next, we consider the non-leaf nodes in G. There are two cases for them.

Case 2: Suppose that node i is not a leaf node in G and $x_i = 0$. Let $S_i^j = \{j \in n(i) : x_j = 1\}$, a subset of node i's neighbors in the original tree G receiving full payment. We have two situations. First, when $p_i = 0$, it means that node i has g_i or more incoming arcs, as shown in Figure EC.7(a). Let $\delta_i = \max\{t_{ji} : j \in S_i^j\}$. It is the biggest t_{ji} value among the nodes in S_i^j. Now, if S_i^j only contains the child nodes of node i, then, each of them must have $b_j^2 = t_{ji} \leq l_i$, given that $p_i = 0$. If S_i^j contains node i's parent (node h), it implies that $|X_i^{PS}| \geq |g_i| - 1$, and X_i^{PS} only contains the

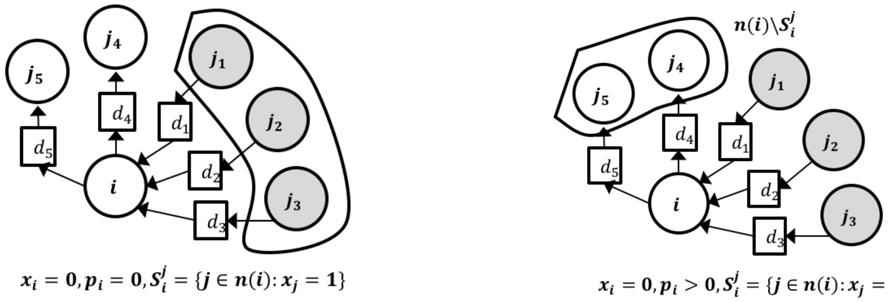


Figure EC.7 **Case 2:** (a) $|S_i^j| \geq g_i$ **and** $\delta_i = \max\{t_{ji} : j \in S_i^j\}$. (b) $|S_i^j| < g_i$ **and** $\gamma_i = \min\{f_i, \min\{t_{ji} : j \in n(i) \setminus S_i^j\}\}$.

child nodes of node i. We can infer the solution for node i when its parent is not selected. When $|X_i^{PS}| > |g_i| - 1$, no extra payment is needed ($P_i^{NPS} = 0$). When $|X_i^{PS}| = |g_i| - 1$, either node i is paid l_i ($P_i^{NPS} = l_i$), or a child node j with $b_j^2 \leq l_i$ is selected in X_i^{NPS}, compared to X_i^{PS}. Thus, $b_i^1 = C_i^{NPS} - C_i^{PS} \leq l_i$. As a result, we have $\delta_i \leq l_i$. Set $w_i = \frac{\delta_i}{l_i}$. Then, set $s_{id} = u_{id} = \delta_i - t_{ji}$ for all j in S_i^j and d in S_i^d. Condition (EC.20) is satisfied because $y_{id} = x_i = 0$ for all d in S_i^d, and $u_{id} = 0$ for all d in $a(i) \setminus S_i^d$. Condition (EC.21) is satisfied because $v_i = 0$. Condition (EC.22) is satisfied because constraint (EC.6) is binding when there are exactly g_i incoming arcs, and $w_i = \delta_i = 0$ when there are more than g_i incoming arcs. Condition (EC.23) is satisfied because constraint (EC.11) is binding for all d in S_i^d, and $y_{di} = 0$ for all d in $a(i) \setminus S_i^d$. Constraint (EC.12) is satisfied because its left-hand side is

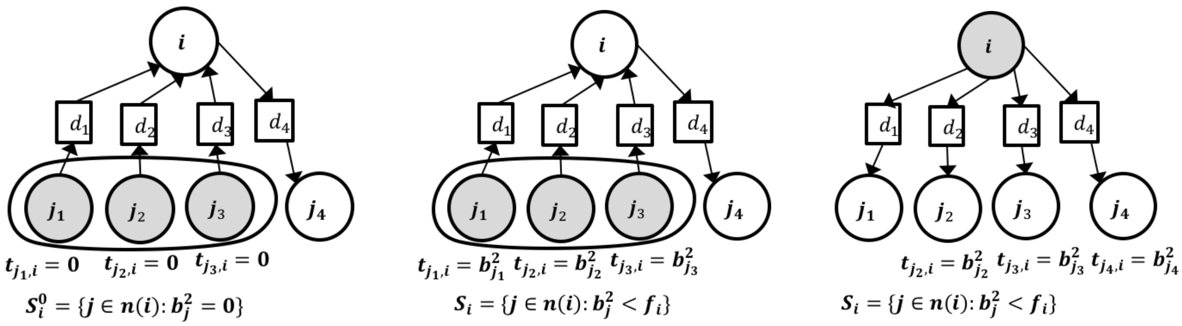
$$
\begin{aligned}
& B_i^{NPS} + b_i^2 - \sum_{j \in S_i^j} (\delta_i - t_{ji}) + g_i \delta_i \qquad \text{(Note: } \sum_{j \in n(i)} t_{ij} = B_i^{NPS} + b_i^2, u_{id} = \delta_i - t_{ji}, w_i = \frac{\delta_i}{l_i}) \\
= & \begin{cases} B_i^{NPS} + \sum_{j \in X_i^{NPS}} b_j^2 + b_i^2 - (|S_i^j| - g_i)\delta_i & \text{if } h \notin S_i^j \\ B_i^{NPS} + \sum_{j \in X_i^{PS}} b_j^2 + b_i^1 + b_i^2 - (|S_i^j| - g_i)\delta_i & \text{if } h \in S_i^j \end{cases} \\
= & \; C_i^{NPS} + b_i^2 - (|S_i^j| - g_i)\delta_i \leq b_i. \qquad \text{(Note: } |S_i^j| \geq g_i, \delta_i \geq 0)
\end{aligned}
$$

Constraint (EC.13) is respected because $w_i = \frac{\delta_i}{l_i} \leq 1$. Thus, conditions (EC.24) and (EC.25) are satisfied because $x_i = p_i = 0$.

Second, when $p_i > 0$, it means that node i has less than g_i incoming arcs, as shown in Figure EC.7(b). Let $\gamma_i = \min\{f_i, \min\{t_{ji} : j \in n(i) \setminus S_i^j\}\}$. We set v_i and w_i by solving the following equations:

$$
f_i v_i + l_i w_i = \gamma_i,
$$
$$
v_i + w_i = 1.
$$

Then, set $s_{id} = u_{id} = \gamma_i - t_{ji}$ for all j in S_i^j and all d in S_i^d. Condition (EC.20) is satisfied because $y_{id} = x_i = 0$ for all d in S_i^d, and $u_{id} = 0$ for all d in $a(i) \setminus S_i^d$. When $p_i \geq l_i + f_i$, $|S_i^j| < g_i - 1$, constraint (EC.5) is binding and constraint (EC.6) is satisfied. It has $\gamma_i = f_i$, $v_i = 1$ and $w_i = 0$. When



(a) $X_i^{PS} \neq \{i\}$ and $|X_i^{PS}| \geq g_i$ (b) $X_i^{PS} \neq \{i\}$ and $|X_i^{PS}| \leq g_i - 1$ (c) $X_i^{PS} = \{i\}$

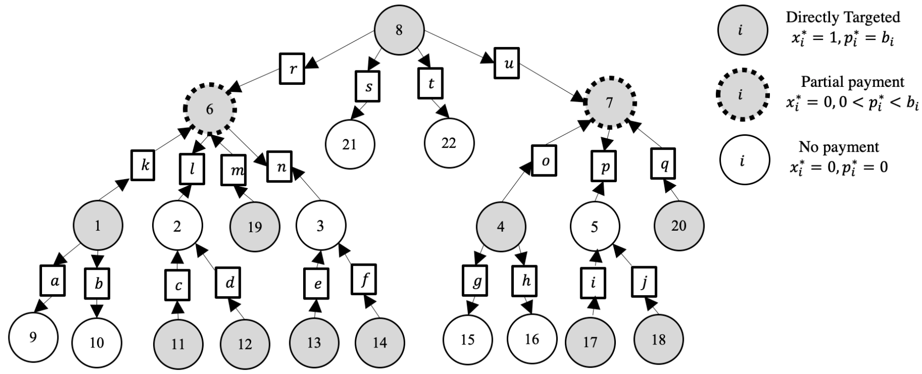
Figure EC.8 Case 3.

$p_i = l_i$, $|S_i^j| = g_i - 1$ and both constraints (EC.5) and (EC.6) are binding. Thus, conditions (EC.21) and (EC.22) are satisfied. Condition (EC.23) is satisfied because constraint (EC.11) is binding for all d in S_i^d and $y_{di} = 0$ for all d in $a(i) \setminus S_i^d$. Analogous to the previous situation, constraint (EC.12) is binding. The left-hand side of constraint (EC.12) is:

$$
\begin{aligned}
& B_i^{NPS} + b_i^2 - \sum_{j \in S_i^j} (\gamma_i - t_{ji}) + b_i v_i + l_i g_i w_i \\
= & \begin{cases}
B_i^{NPS} + \sum_{j \in X_i^{NPS}} b_j^2 + b_i - |X_i^{NPS}| f_i + b_i^2 & \text{if } h \notin S_i^j \text{ and } p_i \geq l_i + f_i \\
B_i^{NPS} + \sum_{j \in X_i^{PS}} b_j^2 + b_i - (|X_i^{PS}| + 1) f_i + b_i^1 + b_i^2 & \text{if } h \in S_i^j \text{ and } p_i \geq l_i + f_i \\
B_i^{NPS} + \sum_{j \in X_i^{NPS}} b_j^2 + b_i v_i + l_i g_i w_i - (g_i - 1)\gamma_i + b_i^2 & \text{if } h \notin S_i^j \text{ and } p_i = l_i \\
B_i^{NPS} + \sum_{j \in X_i^{PS}} b_j^2 + b_i v_i + l_i g_i w_i - (g_i - 1)\gamma_i + b_i^1 + b_i^2 & \text{if } h \in S_i^j \text{ and } p_i = l_i
\end{cases} \\
= & C_i^{NPS} + b_i^2 = b_i.
\end{aligned}
$$

Thus, Condition (EC.24) is satisfied because $x_i = 0$. Condition (EC.25) is satisfied because constraint (EC.13) is binding.

Case 3: Suppose that node i is not a leaf node in G and $x_i = 1$. This means that $p_i = 0$, $y_{id} = 1$ and $y_{di} = 0$ for all d in $a(i)$. Then, CS conditions (EC.20), (EC.21), and (EC.22) are satisfied because those corresponding primal constraints are satisfied and binding. Constraints (EC.11) and (EC.13) are satisfied. Thus, conditionss (EC.23) and (EC.25) are satisfied because $y_{di} = 0$ and $p_i = 0$. Because $x_i = 1$, constraint (EC.12) must be binding. Let LHS denote the value of the left-hand side of constraint (EC.12). So far, $LHS = \sum_{j \in n(i)} t_{ij} = B_i^{NPS} + b_i^2$. If $LHS = b_i$, we are done. Otherwise, the idea of the following proof is to show that a dual solution can first be constructed to ensure that $LHS \geq b_i$ and then it can be adjusted to have $LHS = b_i$. Recall that X_i^{PS} is a set of nodes selected in the Parent-Selected case for node i in the DP. Based on X_i^{PS}, we consider three situations. First, suppose that $X_i^{PS} \neq \{i\}$ and $|X_i^{PS}| \geq g_i$, as shown in Figure EC.8(a). This means that the size of node i's child nodes with a zero b^2 value is at least g_i ($|S_i^0| \geq g_i$) and $X_i^{PS} = X_i^{NPS}$. Thus, $LHS = \sum_{j \in n(i)} t_{ij} = B_i^{NPS} + b_i^2 = C_i^{PS} + b_i^1 + b_i^2 = b_i$ because $b_i^1 = 0$ and $B_i^{NPS} = C_i^{PS}$.



$t_{1,9}=10$	$t_{7,20}=5$	$v_{18}=1$
$t_{1,10}=10$	$t_{6,8}=10$	$v_{19}=1$
$t_{2,11}=1$	$t_{8,6}=5$	$v_{20}=1$
$t_{2,12}=1$	$t_{8,21}=1$	$v_{21}=1$
$t_{3,13}=1$	$t_{8,22}=1$	$v_{22}=1$
$t_{3,14}=1$	$t_{7,8}=1$	$w_1=1$
$t_{4,15}=10$	$t_{8,7}=3$	$s_{1,a}=1$
$t_{4,16}=10$	$v_9=1$	$s_{1,b}=1$
$t_{5,17}=1$	$v_{10}=1$	$s_{4,g}=17$
$t_{5,18}=1$	$v_{11}=1$	$s_{8,r}=2$
$t_{1,6}=2$	$v_{12}=1$	$v_6=0.2$
$t_{6,1}=1$	$v_{13}=1$	$w_6=0.8$
$t_{2,6}=7$	$v_{14}=1$	$s_{6,k}=3$
$t_{6,19}=9$	$v_{15}=1$	$s_{6,m}=5$
$t_{3,6}=5$	$v_{16}=1$	$v_7=1$
$t_{5,7}=4$	$v_{17}=1$	$s_{7,o}=3$
		$s_{7,q}=3$

Figure EC.9 **(a) Primal Solution for the PIDS-PP Problem Instance from Figure EC.1. (b) Non-Zero Dual Variable Values Except for u.**

Second, suppose that $X_i^{PS} \neq \{i\}$ and $|X_i^{PS}| \le g_i - 1$, as shown in Figure EC.8(b). This implies that $X_i^{PS} \neq X_i^{NPS}$. Set $w_i = 1$ if $|X_i^{PS}| = g_i - 1$. Set $v_i = 1$ if $|X_i^{PS}| < g_i - 1$. Then, set $u_{id} = s_{id} = f_i v_i + l_i w_i - t_{ji} = f_i v_i + l_i w_i - b_j^2$ for all j in X_i^{PS} and d in $a(i) \cap a(j)$. Then,

$$
\begin{aligned}
LHS &= \sum_{j \in n(i)} t_{ij} - \sum_{d \in a(i)} u_{id} + b_i v_i + l_i g_i w_i \\
&= \sum_{j \in n(i)} t_{ij} + \sum_{j \in X_i^{PS}} b_j^2 - |X_i^{PS}|(f_i v_i + l_i w_i) + b_i v_i + l_i g_i w_i \\
&= \begin{cases} B_i^{NPS} + b_i^2 + \sum_{j \in X_i^{PS}} b_j^2 + l_i & \text{if } w_i = 1 \text{ and } |X_i^{PS}| = g_i - 1 \\ B_i^{NPS} + b_i^2 + \sum_{j \in X_i^{PS}} b_j^2 + (g_i - 2 - |X_i^{PS}|) f_i + l_i + f_i & \text{if } v_i = 1 \text{ and } |X_i^{PS}| < g_i - 1 \end{cases} \\
&\ge C_i^{PS} + b_i^1 + b_i^2 = b_i
\end{aligned}
$$

The last inequality holds because $C_i^{PS} = B_i^{NPS} + \sum_{j \in X_i^{PS}} b_j^2$ and $b_i^1 \le l_i$ when $|X_i^{PS}| = g_i - 1$, and $b_i = (g_i - 1) f_i + l_i$, $C_i^{PS} = B_i^{NPS} + \sum_{j \in X_i^{PS}} b_j^2 + (g_i - 2 - |X_i^{PS}|) f_i + l_i$ and $b_i^1 \le f_i$ when $|X_i^{PS}| < g_i - 1$.

Lastly, suppose that node i is selected in the Parent-Selected solution ($X_i^{PS} = \{i\}$), as shown in Figure EC.8(c). This implies that $X_i^{NPS} = X_i^{PS}$. Let $DS_i = S_i$ initially. If $|S_i| \ge g_i$, let $DS_i = S_{g_i - 1}$. Set $v_i = 1$ and $u_{id} = s_{id} = f_i - t_{ji} = f_i - b_j^2$ for all j in DS_i and d in $a(i) \cap a(j)$. Thus, $LHS = B_i^{NPS} + \sum_{j \in DS_i} b_j^2 - |DS_i| f_i + b_i = B_i^{NPS} + \sum_{j \in DS_i} b_j^2 + (g_i - 1 - |DS_i|) f_i + l_i > b_i$ as $b_i = (g_i - 1) f_i + l_i$, $b_i^2 = 0$ and node i is selected by the DP algorithm. At this point, we ensure that $LHS \ge b_i$. Then, if $LHS > b_i$, set $u_{i\tilde{d}} = s_{i\tilde{d}} = u_{i\tilde{d}} + LHS - b_i$ for a $\tilde{d}$ in $a(i)$. Thus, condition (EC.24) is satisfied because constraint (EC.12) is binding by construction. $\square$

Figure EC.9(a) is the transformed graph, and its solution is based on the instance in Figure EC.1. We use it to illustrate the procedure for setting the dual variables. The non-zeros dual variables are displayed in Figure EC.9(b), except for u variables, which have the same value as s variables.

For clarity, we use a comma to separate the two subscripts in u_{ij}, t_{ij}, and s_{ij} (i.e., we write them as $u_{i,j}$, $t_{i,j}$, and $s_{i,j}$).

First, we assign values for t variables. Their values are in the first two columns of the table in Figure EC.9(b) and are displayed in Figure EC.9(a) as well. For Case 1, we have nodes 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, and 22. Set their associated v as 1. For Case 2, we have nodes 2, 3, 5, 6, and 7. For nodes 2, 3, and 5, $\delta_2 = \delta_3 = \delta_5 = 0$. Thus, no variables need to be changed. For node 6, it has $S_6^j = \{1, 8, 19\}$, $p_6 = 4$, and $\gamma_6 = \min\{9, \min\{5, 7\}\} = 5$. Thus, $v_6 = 0.2$, $w_6 = 0.8$, $s_{6,k} = u_{6,k} = 5 - 2 = 3$, and $s_{6,m} = u_{6,m} = 5$. For node 7, it has $S_7^j = \{4, 8, 20\}$, $p_7 = 1$, and $\gamma_7 = \min\{3, \min\{4\}\} = 3$. Thus, $v_7 = 1$, $s_{7,o} = u_{7,o} = 3$, and $s_{7,q} = u_{7,q} = 3$. For Case 3, we have nodes 1, 4, and 8. For node 1, set $w_1 = 1$, $s_{1,a} = u_{1,a} = 1$, and $s_{1,b} = u_{1,b} = 1$ because $X_1^{PS} = \{9, 10\}$. For nodes 4 and 8, set $s_{4,g} = u_{4,g} = 20 - 3 = 17$ and $s_{8,r} = u_{8,r} = 10 - 8 = 2$. The objective value is 59, which is exactly the same as that of the solution obtained by Algorithm 2 in Appendix EC.2.