Lecture Notes in Management Science (2013) Vol. **5**: 93–100
5th International Conference on Applied Operational Research, Proceedings

ISSN 2008-0050 (Print), ISSN 1927-0097 (Online)

A dynamic programming approach for Passive Optical Network design in tree graphs

Matthieu Chardy [1], Cédric Hervet [2] and Dedy Bossia [3]

[1] *Orange Labs, Issy-Les-Moulineaux, France*
matthieu.chardy@orange.com

[2] *ENSTA/CEDRIC – Orange Labs, Issy-Les-Moulineaux, France*
cedric.hervet@orange.com

[3] *Institut Galilée – Université Paris 13, Villetaneuse, France*

Abstract. The deployment of Fiber To The Home technologies is currently one of the most challenging issues for telecommunication operators. This paper focuses on the optimization of Passive Optical Network in tree graphs for which a dynamic programming solving approach is proposed. Tests performed on real-size instances prove the efficiency of this approach in comparison to integer linear programming based approaches.

Keywords: fiber to the home; passive optical network; dynamic programming

Introduction

During the last decade, the telecommunication ecosystem has drastically changed with the emergence of bandwidth-consuming service providers (let us name Google or YouTube), forcing network owners to adapt themselves to the subsequent increase in bandwidth needs from their subscribers. For long, the bottleneck of networks in terms of capacity has been located in the core of the network but it recently switched to the access network (the first miles from the customer premises). Concerning fixed access networks, legacy copper networks do no longer fit this need for bandwidth upgrade. A transformation of the physical transport layer has become ineluctable, with the introduction of the fiber technology as close as possible from the customers.

France-Telecom Orange favors the Fiber To The Home (FTTH) technology embedding the Passive Optical Network (PON) architecture. PON is a specific point to multipoint fiber architecture, where a level-1 fiber originating from the entry point of the core network (called the Optical Line Termination, OLT) is divided into several level-2 fibers when going through a passive network element called optical splitter (of level 1). This number of outgoing fibers is limited by the capacity of the splitter. Then, fibers of given level are successively divided into several ones of the upper level when going through passive optical splitters of this level. The number of splitters a fiber has to go through before reaching a demand node defines the number of levels of the PON architecture (see Figure 1). This paper focuses on the minimization of PON deployment costs in a tree structure, modeled as a joint location and routing optimization problem.

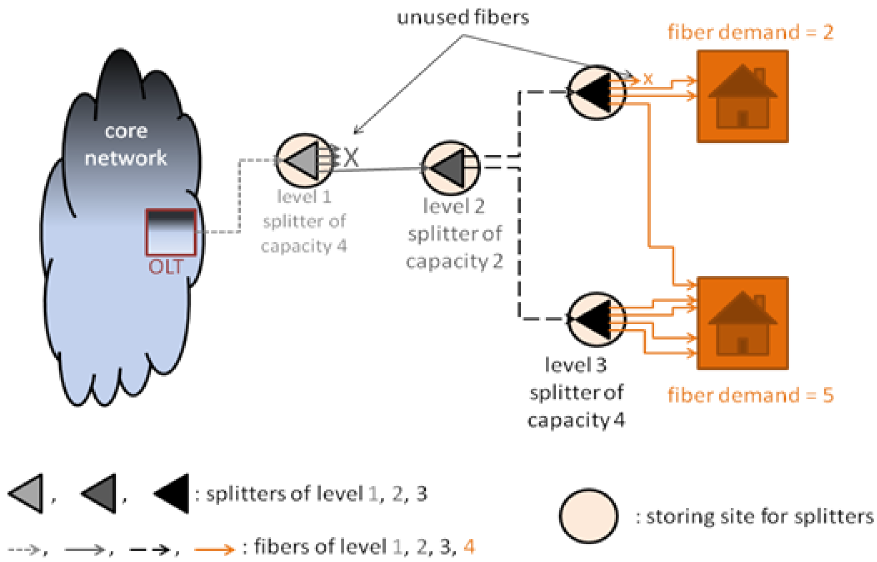

Fig. 1. An example of a PON architecture with 3 levels of optical splitters (and 4 levels of fibers).

Optimal PON design is for sure a key problem for telecommunication operators, considering both the equipments and labor costs involved. A key feature is undeniably the re-use of the existing civil engineering infrastructure. The motivation for tackling this specific version of the problem is both practical and theoretical. Previous related work by Chardy et al. (2012) show that the real-life civil engineering infrastructures are close to tree graphs. Moreover Chardy et al. (2012), Hervet et al. (2012), Gollowitzer and Ljubic (2011) and Gollowitzer et al. (2011) suggest that PON design problems are hard to solve to optimality in a general framework by the mean of integer linear programming approaches: the exact solving of the PON design problem in a tree structure should therefore be of interest for decomposition methods.

The aim of the paper is to present dynamic programming approaches (introduced by Bellman (1954)) dedicated to the Passive Optical Network design problem in a tree structure as defined in Kim et al. (2011), for which authors propose both mixed integer formulations and branch and bound based exact and heuristic solving approaches. The remainder of the paper is organized as follows. The next section is dedicated to the definition and modeling of the problem and the formulation of several optimal solutions' properties, from which we derive a dedicated labeling algorithm presented in the following section. Numerical results are reported in the next section to assess the efficiency of the approach, before concluding.

Mathematical modeling

This section provides a reformulation of the integer linear program proposed by Kim et al. (2011): this formulation is a straightforward adaptation of models proposed by Chardy et al. (2012) to tree-shaped civil engineering structures.

Let us introduce the model for the PON design problem for a *K*-level of splitters architecture (i.e. *K+1* levels of fibers). Let *T(V,A)* be an arborescence with a set of vertices *V* (the root is denoted by *olt*) and a set of arcs *A*. A fiber demand $a_i \in \mathbb{N}$ is associated to each leaf node $i \in V$. Let us respectively denote by $m^k$and $C^k, \forall k = 1 \dots K$ the capacity and cost of a level-*k* splitter. Fibers are routed in cables contained in a set *L*. Each type of cable $l \in L$ has a capacity $b^l \in \mathbb{N}$ and a routing cost d_{ij}^l through each arc $(i,j) \in A$.

With respect to decision variables, the number of level-*k* splitters installed at a node *i* is denoted by $\mathbf{z}_i^k \in \mathbb{N}$. Likewise, $\mathbf{f}_{ij}^k \in \mathbb{N}$ denotes the number of level-*k* fibers routed along the arc *(i,j)*. In our context, the cabling policy specifies that at most one cable can be chosen per arc: the cable selected for the arc $(i,j) \in A$ is thus described though the set of binary variables $\mathbf{c}_{ij}^l \in \mathbb{N}, l \in L$.

The PON_Kdesign problem is to find a deployment of fibers from the OLT to the demand nodes (going through the required number of levels of splitters i.e. *K*) of minimal cost, defined as a combination of splitters and cables installation costs. The set of feasible solutions, denoted by $PON_K(T)$, can be formulated as follows:

$$(\mathbf{z},\mathbf{f},\mathbf{c}) \in PON_K(T) \Leftrightarrow$$

$$\begin{cases}
\mathbf{f}^1_{\Gamma^{-1}(i)i} = \mathbf{z}^1_i + \sum\limits_{j\in\Gamma^+(i)} \mathbf{f}^1_{ij} & \forall i \in V\backslash\{olt\}, & (1)\\
\mathbf{f}^k_{\Gamma^{-1}(i)i} + m^{k-1}\mathbf{z}^{k-1}_i \geq \mathbf{z}^k_i + \sum\limits_{j\in\Gamma^+(i)} \mathbf{f}^k_{ij} & \forall i \in V\backslash\{olt\}, \forall k = 2...K, & (2)\\
\mathbf{f}^{K+1}_{\Gamma^{-}(i)i} + m^K\mathbf{z}^K_i \geq a_i + \sum\limits_{j\in\Gamma^+(i)} \mathbf{f}^{K+1}_{ij} & \forall i \in V\backslash\{olt\}, & (3)\\
\sum\limits_{k=1}^{K+1} \mathbf{f}^k_{ij} \leq \sum\limits_{l\in L} b^l\mathbf{c}^l_{ij} & \forall (i,j) \in A, & (4)\\
\sum\limits_{l\in L} \mathbf{c}^l_{ij} \leq 1 & \forall (i,j) \in A, & (5)\\
\mathbf{f}^k_{ij} \in \mathbb{N},\ \forall (i,j) \in A,\ k \in 1..K+1, & \mathbf{z}^k_i \in \mathbb{N}, \forall i \in V\backslash\{olt\},\ k \in 1..K & \\
\mathbf{c}^l_{ij} \in \{0,1\},\ \forall (i,j) \in A,\ \forall l \in L, & &
\end{cases}$$

where $\Gamma^-(i)$ and $\Gamma^+(i)$ respectively denote the father-node and the set of son-nodes of node *i* in the arborescence.

The optimal value for the K-levels PON design problem in an arborescence, denoted by $FTTH_K$, can be formulated as follows:

$$FTTH_K = \min_{(\mathbf{z},\mathbf{f},\mathbf{c}) \in PON_K(T)} \sum_{i \in V \setminus \{olt\}} \sum_{k=1}^{K} C^k \mathbf{z}_i^k + \sum_{(i,j) \in A} \sum_{l \in L} d_{ij}^l \mathbf{c}_{ij}^l$$

Constraints (1)-(3) are fiber flow constraints, taking into account the fact that not all fibers going out of a splitter are necessarily used. The set of constraints (4) ensures that fibers going through an arc must be contained in a subset of cables while constraints (5) induce that at most one cable is used per arc. Let us note that the assumption is made that the cable of greatest capacity is large enough and that the arc capacities are neglected (supposed large enough too).

The rest of this section is dedicated to the presentation of some optimal solution properties that are of interest for the design of the labeling algorithm presented in the next section.

Proposition 1. *Let us denote by T(j) the sub-tree of T(V,A) rooted at node $j \neq olt$. A lower bound of level-k splitters installed in any sub-tree is given by the following recursion:*

$$\sum_{i \in T(j)} \mathbf{z}_i^K \geq \left\lfloor \frac{\sum_{i \in T(j)} a_i}{m^K} \right\rfloor \quad \forall j \in V$$

$$\sum_{i \in T(j)} \mathbf{z}_i^{k-1} \geq \left\lfloor \frac{\sum_{i \in T(j)} \mathbf{z}_i^k}{m^{k-1}} \right\rfloor \quad \forall j \in V,\ \forall k \in 2..K$$

Proof. Due to the size of the article, the proof is not given. Nevertheless, this property is a straight-forward generalization of results from Kim et al. (2011) and the reader can refer to it for the proof dedicated to cases $k = 1{,}2$.

Corollary 1. *In any optimal solution, a lower bound for the number of level-k splitters installed at any node $j \neq olt$ is given by:*

$$\mathbf{z}_j^K \geq \left\lfloor \frac{\sum_{i \in T(j)} a_i}{m^K} \right\rfloor - \sum_{i \in T(j') | j' \in \Gamma^+(j)} \mathbf{z}_i^K$$

$$\mathbf{z}_j^k \geq \left\lfloor \frac{\sum_{i \in T(j)} \mathbf{z}_i^k}{m^{k-1}} \right\rfloor - \sum_{i \in T(j') | j' \in \Gamma^+(j)} \mathbf{z}_i^{k-1} \quad \forall k \in 2..K$$

Proposition 2. *An upper bound for the number level-k splitters installed in any sub-tree is given by the following recursion:*

$$\sum_{i\in T(j)} \mathbf{z}_i^K \le \sum_{i\in T(j)} \left\lceil \frac{a_i}{m^K} \right\rceil \quad \forall j \in V$$

$$\sum_{i\in T(j)} \mathbf{z}_i^{k-1} \le \sum_{i\in T(j)} \left\lceil \frac{\mathbf{z}_i^k}{m^{k-1}} \right\rceil \quad \forall j \in V,\ \forall k \in 2..K$$

Proof. Due to the size of the article, we only present a sketch of the proof. The key idea is that the strategy consisting in installing $\left\lceil \frac{a_i}{m^K} \right\rceil$ level-K splitters at each demand node *i* defines a feasible solution to serve the clients' fiber demand if they are properly served by level-*(K-1)* splitters. Then we notice that this strategy is the most consuming in terms of level-*K* splitters in any sub-tree (no sharing of splitter capacity between demand nodes). The same reasoning applies to any level of splitter $k = 1 \ldots K - 1$ considering the installed level-(*k+1)* splitters as the demand.

Corollary 2. *In any optimal solution, an upper bound for the number of splitters installed at any node $j \neq olt$ is given by:*

$$\mathbf{z}_j^K \le \sum_{i\in T(j)} \left\lceil \frac{a_i}{m^K} \right\rceil - \sum_{i\in T(j')|j'\in\Gamma^+(j)} \mathbf{z}_i^K$$

$$\mathbf{z}_j^{k-1} \le \sum_{i\in T(j)} \left\lceil \frac{\mathbf{z}_i^k}{m^{k-1}} \right\rceil - \sum_{i\in T(j')|j'\in\Gamma^+(j)} \mathbf{z}_i^{k-1} \quad \forall k \in 2..K$$

A dedicated labeling algorithm

For the dynamic programming framework, we define a recursive function in order to model the problem, close to those of location-allocation problems (the reader can refer to Bauguion et al. (2011) for the design of such algorithm in a cache location context).

Definition 1. For a given sub-tree *T(i)* rooted at node *i* and a feasible configuration *s*=(**z**, **f**, **c**) in this sub-tree, which total number is limited thanks to corollaries 1 and 2, we define the remaining demand of level-*k* fibers (respectively the subscriber and splitter demand for *k=K+1* and *k=1...K*) associated to the configuration *s* as $\hat{a}_i^k(s) \in \mathbb{N}$ by the recursive function:

$$\hat{a}_i^k(s) = \max\{\sum_{j\in\Gamma^+(i)} \hat{a}_j^k(s) - m^{k-1}\mathbf{z}_i^{k-1}(s); 0\},\ \ \forall k = 2..K+1$$

$$\hat{a}_i^1(s) = \sum_{j\in\Gamma^+(i)} \hat{a}_j^1(s) + \mathbf{z}_i^1(s)$$

with, for any leaf node *i'* of the sub-tree :

$$\hat{a}_{i'}^{K+1}(s) = \max\left\{a_{i'}(s) - m^K \mathbf{z}_{i'}^K(s); 0\right\},$$
$$\hat{a}_{i'}^{k}(s) = \max\left\{\mathbf{z}_{i'}^k(s) - m^{k-1}\mathbf{z}_{i'}^{k-1}(s); 0\right\}, \ \forall k = 2..K,$$
$$\hat{a}_{i'}^{1}(s) = \mathbf{z}_{i'}^1$$

Definition 2. For a given sub-tree *T(i)* rooted at $i \neq olt$ and a feasible configuration *s*=(**z**, **f**, **c**) in this sub-tree, we define the cost of that configuration with $cost_i(s)$ through the recursive function :

$$cost_i(s) = \sum_{j \in \Gamma^+(i)} cost_j(s) + \sum_{k=1}^{K} C^k \mathbf{z}_i^k(s) + \sum_{l \in L} d_{\Gamma^-(i)i}^l \mathbf{c}_{\Gamma^-(i)i}^l(s)$$

with, for any leaf node *i'* of the sub-tree:

$$cost_{i'}(s) = \sum_{k=1}^{K} C^k \mathbf{z}_{i'}^k(s) + \sum_{l \in L} d_{\Gamma^-(i')i'}^l \mathbf{c}_{\Gamma^-(i')i'}^l(s)$$

Definition 3. For a given sub-tree graph *T(i)*, let *s*=(**z**, **f**, **c**) and *s'*=(**z'**, **f'**, **c'**) be two feasible configurations for the problem, i.e. they both belong to $PON_K(T(i))$. We define the domination (denoted by $\succcurlyeq$) between those two configurations as follows :

$$s \succeq s' \Leftrightarrow \begin{cases} \hat{a}_i^k(s) \leq \hat{a}_i^k(s'), \ \forall k = 1..K+1 \\ cost_i(s) \leq cost_i(s') \end{cases}$$

Definition 4. A solution *s*=(**z**, **f**, **c**), for a given sub-tree *T(i)*, is defined as relevant for this sub-tree if it is not dominated by any other solution of $PON_K(T(i))$. The set of relevant solutions for the sub-tree *T(i)* is denoted by $S_{\succcurlyeq}(i)$.

Finally we introduce the recursive relation between sets of relevant solutions:

$$S_{\succeq}(i) = \{ \prod_{j \in \Gamma^+(i)} S_{\succeq}(j) \cap PON_K(T(i)) \mid \forall (s, s') \in S_{\succeq}(i)^2, \ s \succeq s' \Rightarrow s = s' \}$$

Numerical results

The objective of this section is to assess the efficiency of the method to solve real-life instances. As mentioned in the introduction, civil engineering infrastructures for deploying FTTH are close to tree graphs of size up to several thousands of nodes and links. Therefore, preliminary tests have been performed on random instances of tree graphs of 500 to 4000 nodes, with a mean number of sons per node equal to 3. Only 1 and 2 levels of splitters architectures (referred as "single'" and "double") are considered with splitters of capacity equal to 8 (i.e. $m^1 = m^2 = 8$). 3 solution

methods are compared: first the labeling algorithm described in the previous section (column "Dyn Prog"), then the CPLEX 12.2 default branch and bound algorithm on Kim et al. and PON_K respective mixed integer formulations (respective columns "Kim et al. formulation" and "PON formulation"). Note that a 1000 seconds time limit is set for any branch and bound solving.
For each "Instance" (characterized by the number of levels of the architecture and the size of the tree graph in terms of number of nodes), we report for the value of the optimal solution ("FTTH*" column) and the computation time ("CPU time" column) for each method. In addition, we provide the CPLEX gap when the branch and bound does not converge within the time limit (the absence of gap means that no feasible integer solution is found).

Table 1. Tests results.

Instance	Dyn Prog		PON formulation		Kim et al. formulation	
	FTTH*	CPU time (sec)	CPU time (sec)	gap (%)	CPU time (sec)	gap (%)
single_50	29034	0.0	5.1	0.0	27.9	0.0
single_100	60934	0.0	987.3	0.0	1000	-
single_500	308052	0.5	1000	65.3	1000	-
single_1000	601888	2.1	1000	-	1000	-
single_1500	900058	18.5	1000	-	1000	-
single_3000	1794255	148.9	1000	-	1000	-
single_4000	2455523	1678.1	1000	-	1000	-
double_50	30082	0.1	237.2	0	1000	-
double_100	63334	0.2	1000	3.5	1000	-
double_500	315411	1.8	1000	-	1000	-
double_1000	617771	20.1	1000	-	1000	-
double_1500	927072	248.2	1000	-	1000	-
double_3000	1908644	1904.3	1000	-	1000	-

The results clearly indicate the tractability of the dynamic programming approach, which solves real-size instances. Note that only the largest instance of each category of architecture exceeds the time limit. In comparison, branch and bound-based approaches seem to be limited to tree graphs of few dozen nodes. Note that the comparison of the mixed integer formulations suggest that the reformulation of Kim at al. (2011) model that we propose is efficient, notably as it seemingly breaks some symmetry: nevertheless this has to be confirmed by further testing on smaller instances.

Conclusion

This paper deals with a Passive Optical Network design problem in tree infrastructures. A new integer linear formulation is proposed for this problem, embedding a general K-level PON architecture. An exact labeling algorithm is proposed taking benefit from the structure of some optimal solutions.

The tests results clearly assess the efficiency of a dynamic programming approach. Nevertheless these are preliminary results that need to be enriched: first, a sensibility analysis with respect to splitter capacity and mean number of son-nodes would be interesting to conduct; second the comparison of the mixed integer formulation is to be deeper investigated both theoretically and with tests performed on small instances. Another research avenue for the future should be the refinement of the cabling policy and the integration of specific rules for field deployments.

Acknowledgments—This research work has been supported by Orange Labs.

References

Bauguion P, Ben-Ameur W and Gourdin E (2011). Cache Location in tree networks; preliminary results. Proceedings of INOC 2011: 517-522.

Bellman R (1954). The theory of Dynamic Programming. Bulleting of the American Mathematical Society 60: 503-515.

Chardy M, Costa MC, Faye A and Trampont M (2012). Optimizing splitter and fiber location in a multilevel optical FTTH network. European Journal of Operational Research 222: 430-440.

Hervet C and Chardy M (2012). Passive optical network design under operations administration and maintenance considerations. Journal of Applied Operational Research 4(3): 152-172.

Gollowitzer S, Gouveia L and Ljubic I (2011). The Two Level Network Design Problem with Secondary Hop Constraints. Proceedings of INOC 2011: 71-76.

Gollowitzer S and Ljubic I (2011). MIP models for connected facility location: a theoretical and computational study. Computers & Operations Research 38(2): 435-449.

Kim Y, Lee Y and Han J (2011). A splitter location-allocation problem in designing fiber optic access networks. European Journal of Operational Research 210: 425-435.