In Algorithm 0.1, we present our implementation of the ECMP algorithm for link-path formulations. Procedure \textit{ECMP Allocation} is executed for every demand and is used to determine the values of \( \bar{x} \) following the ECMP allocation. \( \mathcal{P}^{nl} \) denotes a set of all shortest paths between node \( n \) and sink \( t \) with \( w_k \) as link weights.

\begin{algorithm}
\caption{ECMP Allocation}
\begin{algorithmic}
\Procedure{ECMP Allocation}{s, t, h_d, \mathcal{P}^{nl}}
\State \( S^{st} = \{ \ell \in \mathcal{L} : \ell \text{ is first link of path } \mathcal{P} \in \mathcal{P}^{st} \} \);
\State \( \delta^{st} := |S^{st}| \);
\State \( h' := \frac{b}{k} \);
\For {\( \ell \in S^{st} \)}
\State \( n := \text{otherend}(\ell, s) \);
\State \( \text{flow}_\ell := \text{flow}_\ell + h' \);
\If {\( n \neq t \)}
\State \( \mathcal{P}^{nl} = \{ \mathcal{P} \setminus \{ \ell \} : \ell \text{ is first link of path } \mathcal{P} \in \mathcal{P}^{st} \} \);
\State \( \text{ECMP Allocation}(n, t, h', \mathcal{P}^{nl}) \);
\EndIf
\EndFor
\State return
\EndProcedure
\end{algorithmic}
\end{algorithm}

Observe that the set of all shortest paths between node \( n \) and sink \( t \), \( \mathcal{P}^{nl} \), is derived from the set of all shortest paths used in the previous step of the recursion. As the recursion progresses in a depth-first-search fashion, the set of shortest paths from node \( n \) (under consideration) to sink \( t \) keeps on getting filtered. At the step of recursion at the node \( n \), the path set \( \mathcal{P}^{nl} \) is a subset of paths \( \mathcal{P}^{st} \) which share the same links up till node \( n \).