In Algorithm 0.1, we present our implementation of the ECMP algorithm for link-path formulations. Procedure *ECMP_Allocation* is executed for every demand and is used to determine the values of $\boldsymbol{x}$ following the ECMP allocation. $\mathcal{P}^{nt}$ denotes a set of all shortest paths between node n and sink t with w_ℓ as link weights.

Algorithm 0.1 ECMP Allocation

```
procedure $ECMP\_Allocation(s, t, h_d, \mathcal{P}^{st})$
begin
  $\mathcal{S}^{st} = \{\ell \in \mathcal{L} : \ell$ is first link of path $\mathcal{P} \in \mathcal{P}^{st}\}$;
  $\delta^{st} := |S^{st}|$;
  $h' := \frac{h_d}{\delta^{st}}$;
  for $\ell \in \mathcal{S}^{st}$ do
    begin
      $n := otherend(\ell, s)$;
      $flow_\ell := flow_\ell + h'$;
      if $n \neq t$ then
        begin
          $\mathcal{P}^{nt} = \{\mathcal{P} \setminus \{\ell\} : \ell$ is first link of path $\mathcal{P} \in \mathcal{P}^{st}\}$;
          $ECMP\_Allocation(n, t, h', \mathcal{P}^{nt})$
        end
      return
    end
end {procedure}
```

Observe that the set of all shortest paths between node n and sink t, $\mathcal{P}^{nt}$, 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 $\mathcal{P}^{nt}$ is a subset of paths $\mathcal{P}^{st}$ which share the same links up till node n.