Algorithm 1 Algorithm to determine optimal parallelization strategy for an operator DAG

procedure $\textsc{FindOptStrategy}(G = (V, E),\ nprocs)$
  for all $v \in V$ **do**
    $v.neighbors \leftarrow \{u : (u, v) \in E \vee (v, u) \in E\}$
    $v.processed \leftarrow \mathsf{False}$
  end for

  $tbl \leftarrow \emptyset$
  while $\exists x \in V : x.processed = \mathsf{False}$ **do**
    $S \leftarrow \{v : v \in V \wedge v.processed = \mathsf{False}\}$ ▷ Set of unprocessed nodes

    ▷ Choose a vertex with least no. of unprocessed neighbors from S
    for all $v \in S$ **do**
      $v.unprocessed\_neighbors \leftarrow \{u : u, v \in S \wedge u \in v.neighbors\}$
      $v.processed\_neighbors \leftarrow v.neighbors \setminus v.unprocessed\_neighbors$
    end for
    $v_{min} \leftarrow \arg\min_{v \in S} |v.unprocessed\_neighbors|$

    ▷ Each element in C is a set of pairs $\{(u_1, cfg_1), \ldots, (u_n, cfg_n)\}$, where u_i is an unprocessed neighbor of v_{min} and cfg_i is a valid configuration of u_i
    $C \leftarrow \prod_{u \in v_{min}.unprocessed\_neighbors} \{(u, cfg) : cfg \in u.configs\}$

    ▷ For each combination of configuration of unprocessed neighbors of v_{min}, find a config of v_{min} that minimizes the cost
    for all $c \in C$ **do**
      $c_{min} \leftarrow \arg\min_{c_v \in v_{min}.configs}(\textsc{Cost}(v_{min}, c_v, c, v_{min}.processed\_neighbors))$
      $tbl \leftarrow tbl \cup \{(v_{min}, c, c_{min})\}$
    end for

    $v.processed \leftarrow \mathsf{True}$
  end while

return $\textsc{OptConfig}(\mathrm{tbl})$
end procedure