

Algorithm 1 Node Mapping Algorithm

- 1: For each possible security demand k , sort all substrate nodes in a candidate node queue $queue(k)$ in descending order of heuristic value H .
 - 2: For all nodes $m \in G_i^S$, initialize their state by setting $Occupied(m) = \text{FALSE}$.
 - 3: **repeat**
 - 4: Get an unmapped node n randomly from G_i^V .
 - 5: $k = dem^V(n)$.
 - 6: **if** \exists node $m \in queue(k)$ s.t. $Occupied(m) = \text{FALSE}$ **and** $dem^S(m) \leq lev^V(n)$ **and** $cpu^S(m) \geq cpu^V(n)$ **then**
 - 7: $Occupied(m) = \text{TRUE}$.
 - 8: Map the virtual node n onto the substrate node m .
 - 9: **else**
 - 10: Release all resources occupied by G_i^V .
 - 11: **return** MAP_FAILED.
 - 12: **until** all nodes in G_i^V are mapped successfully.
 - 13: **return** NODE_MAP_SUCCESS.
-