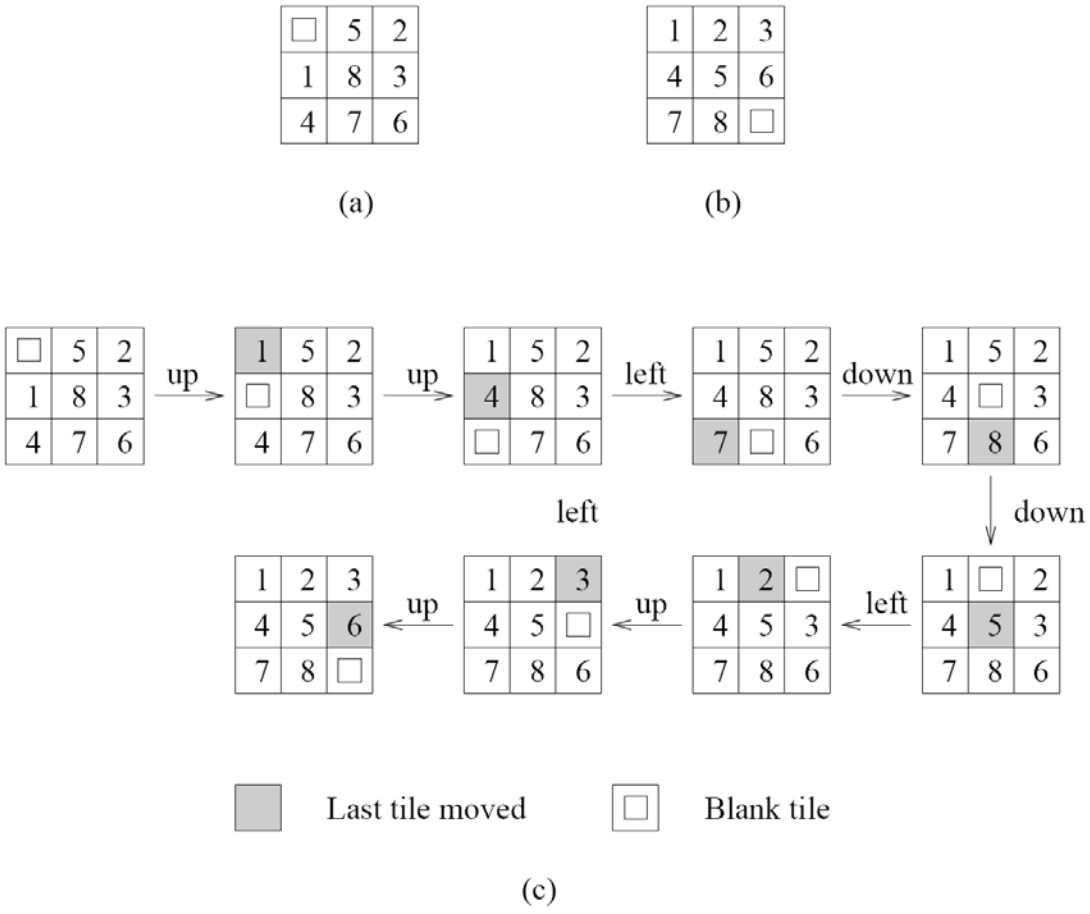

Figure 11.1 An 8-puzzle problem instance: (a) initial configuration; (b) final configuration; and (c) a sequence of moves leading from the initial to the final configuration.

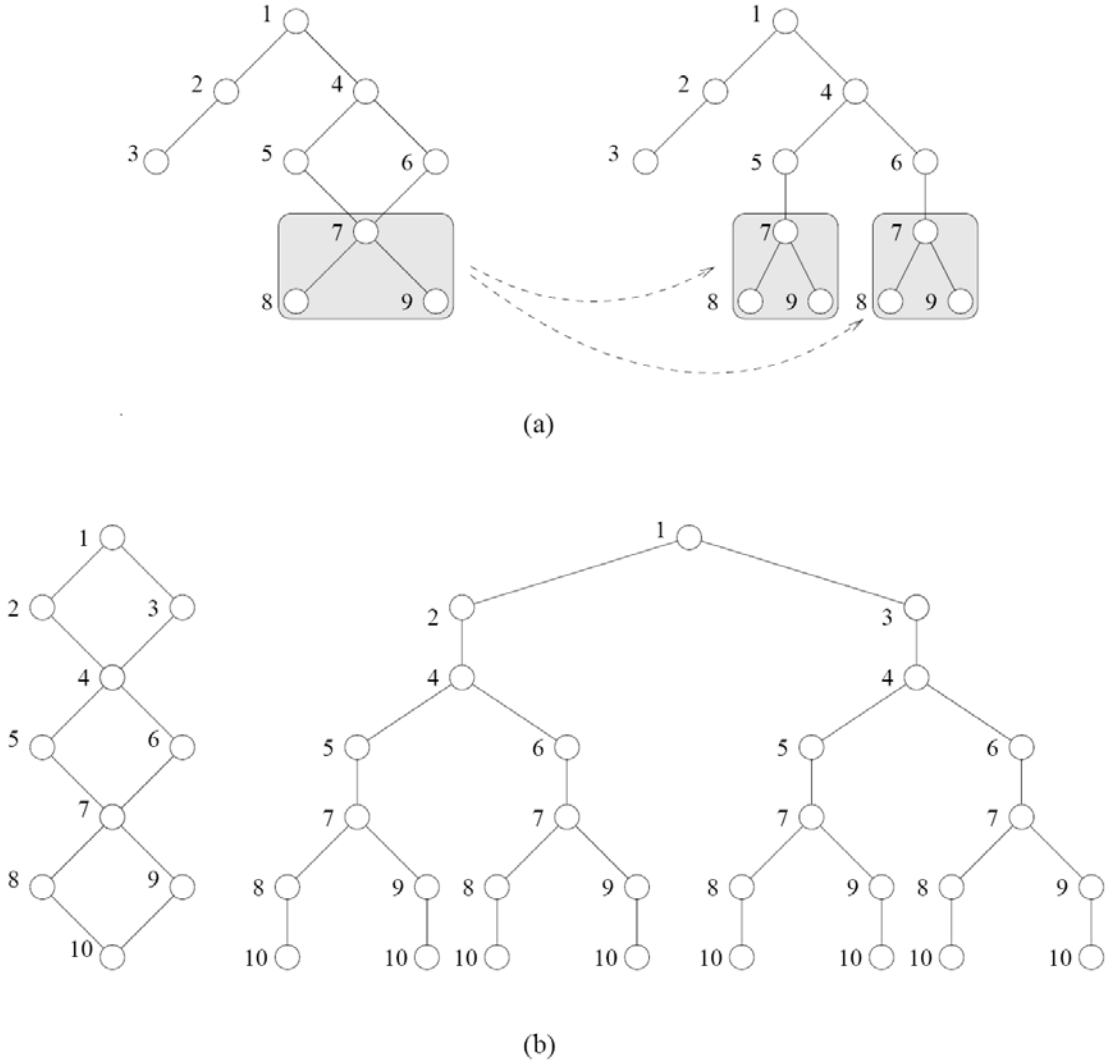

Figure 11.3 Two examples of unfolding a graph into a tree.

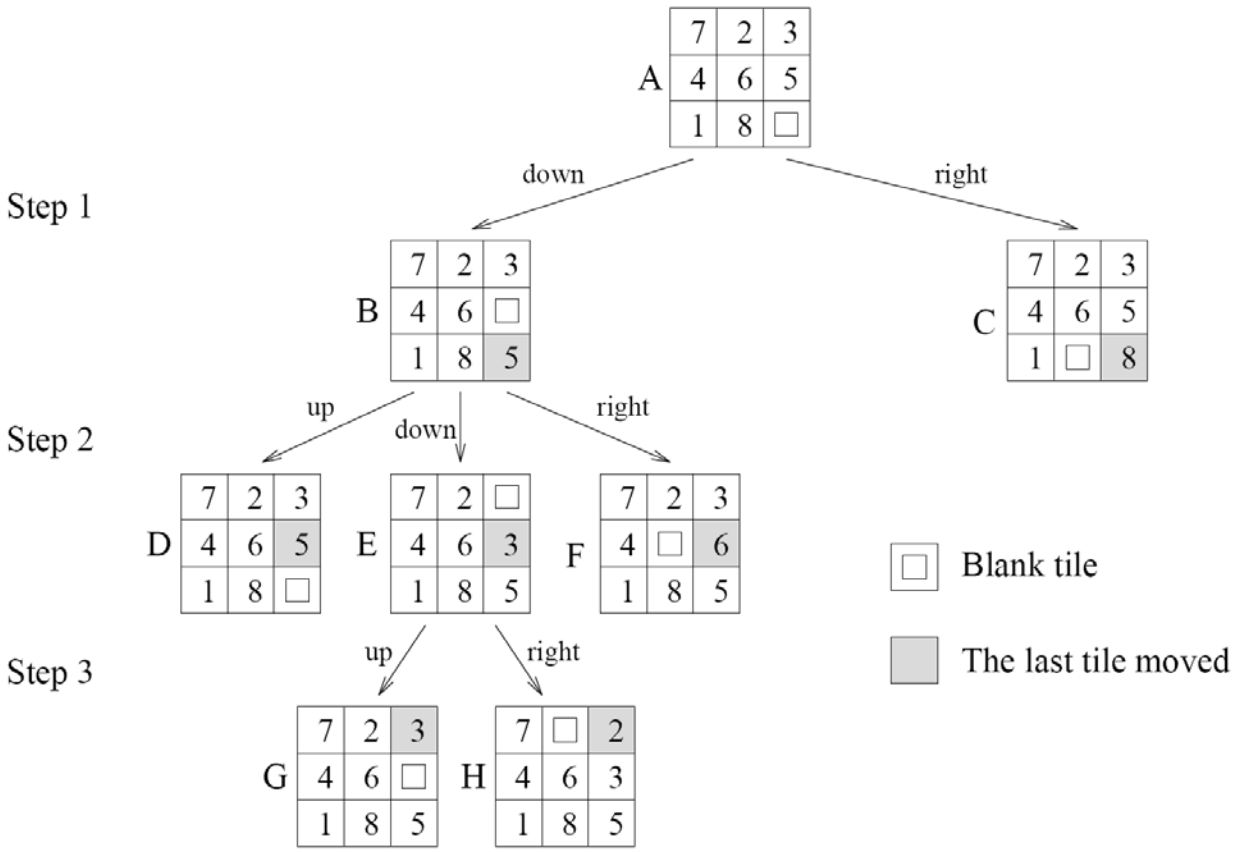

Figure 11.4 States resulting from the first three steps of depth-first search applied to an instance of the 8-puzzle.

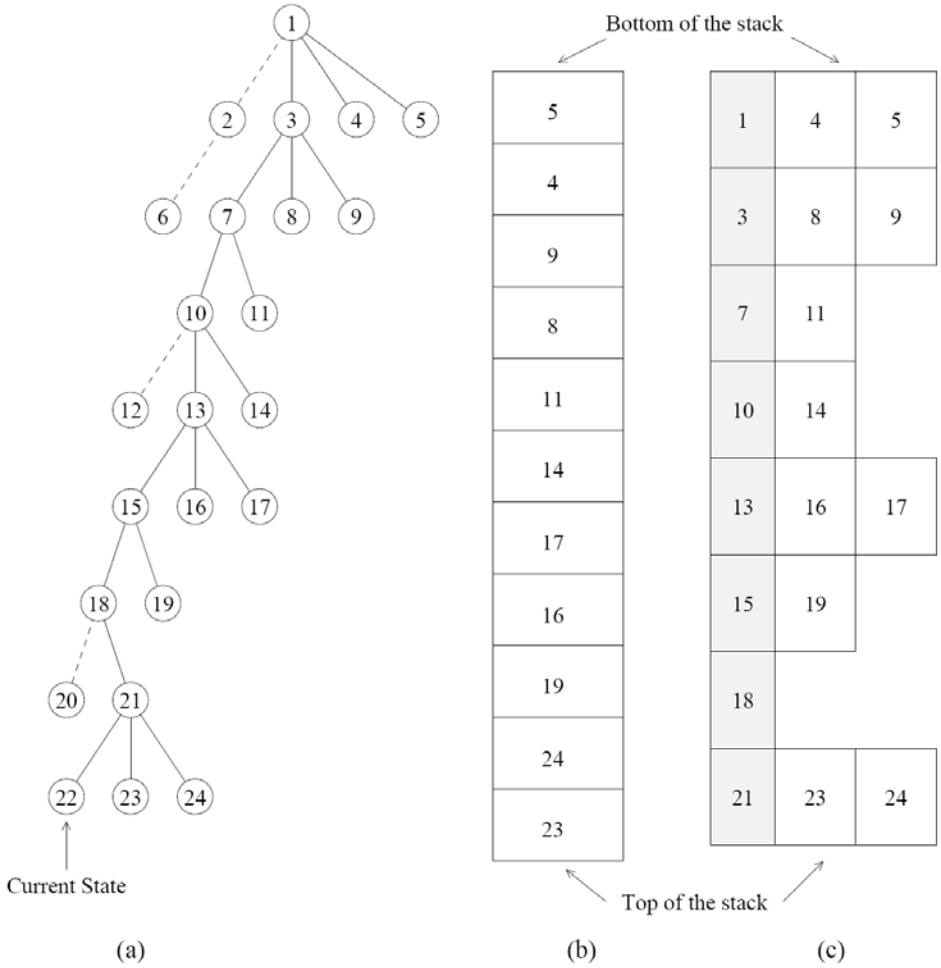

Figure 11.5 Representing a DFS tree: (a) the DFS tree; successor nodes shown with dashed lines have already been explored; (b) the stack storing untried alternatives only; and (c) the stack storing untried alternatives along with their parent. The shaded blocks represent the parent state and the block to the right represents successor states that have not been explored.

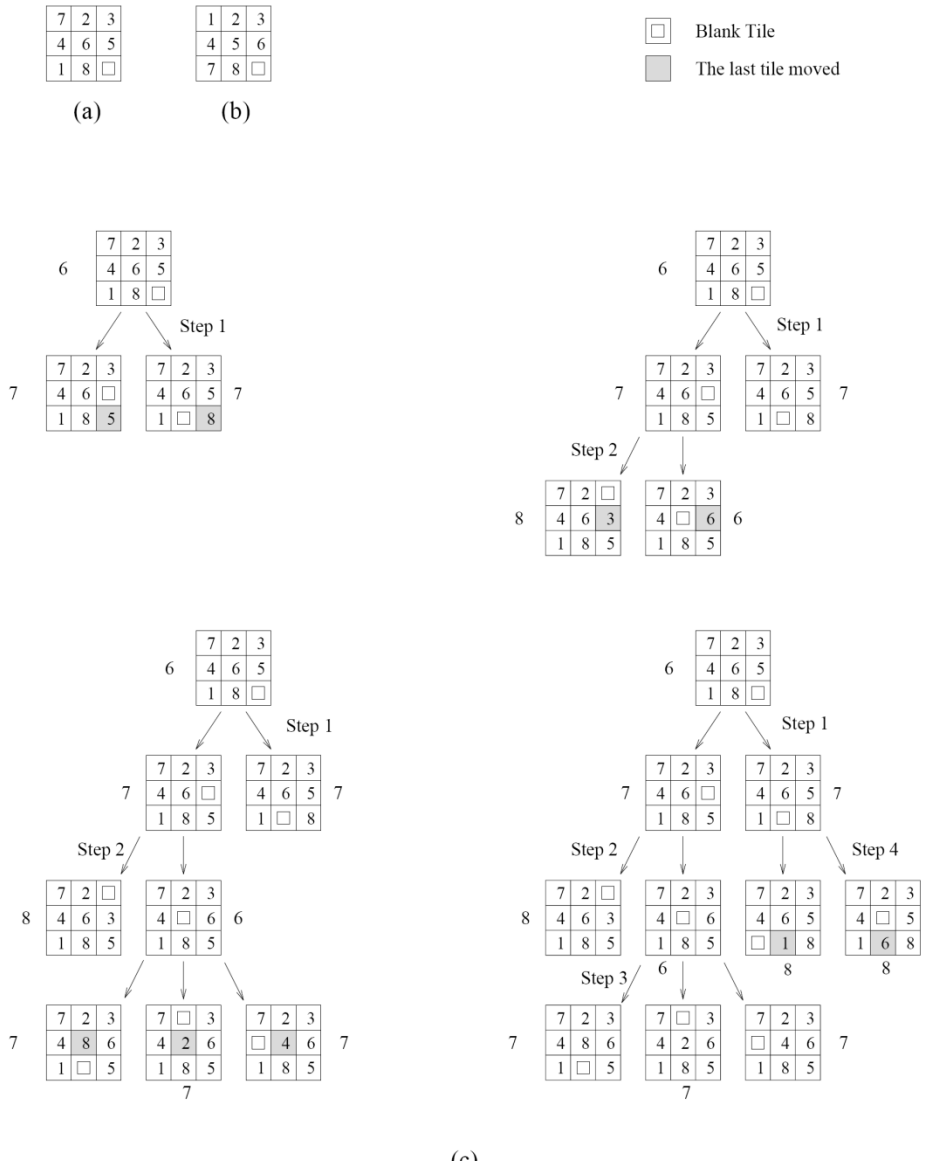

Figure 11.6 Applying best-first search to the 8-puzzle: (a) initial configuration; (b) final configuration; and (c) states resulting from the first four steps of best-first search. Each state is labeled with its h-value (that is, the Manhattan distance from the state to the final state).

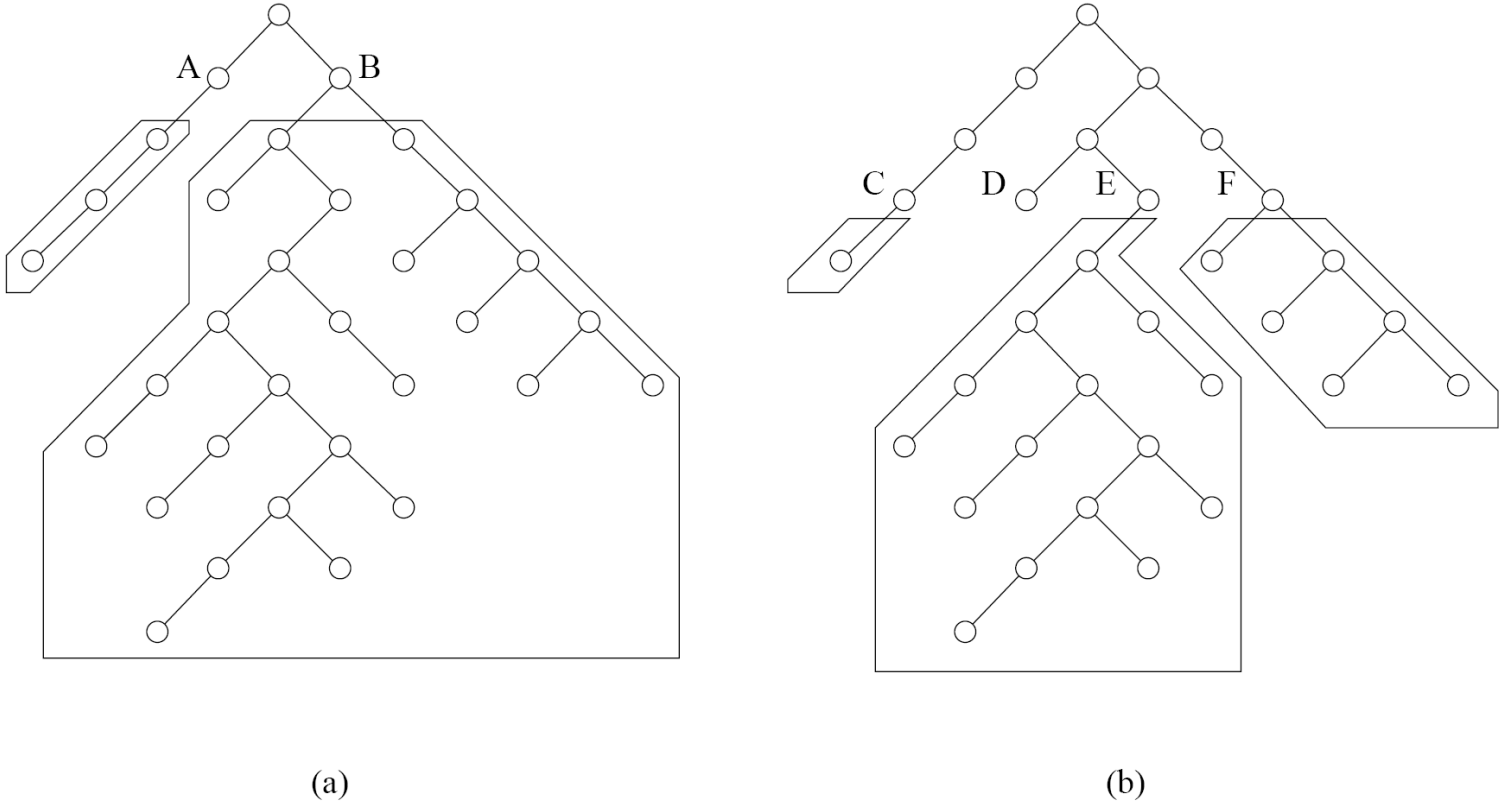

Figure 11.7 The unstructured nature of tree search and the imbalance resulting from static partitioning.

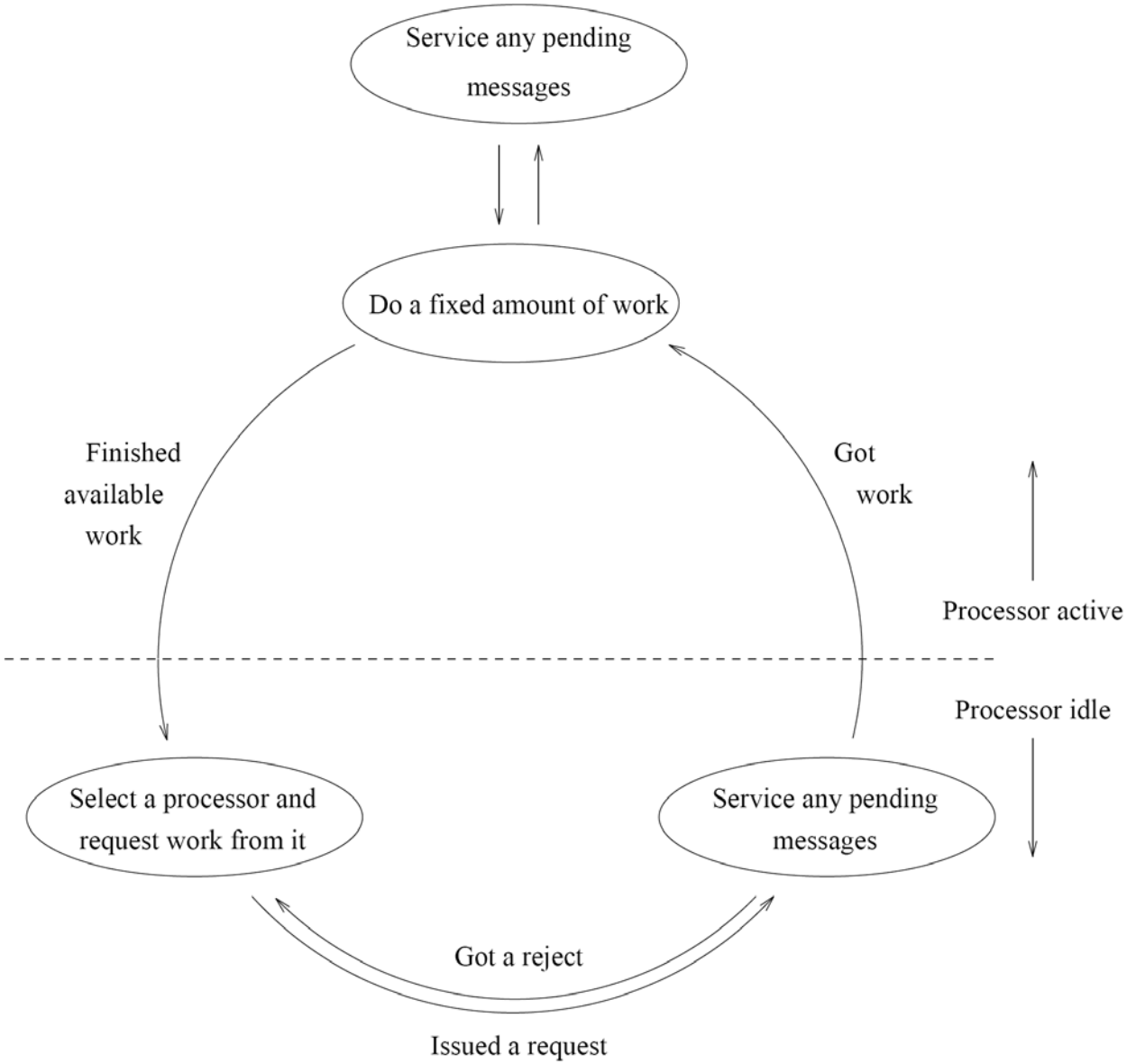

Figure 11.8 A generic scheme for dynamic load balancing.

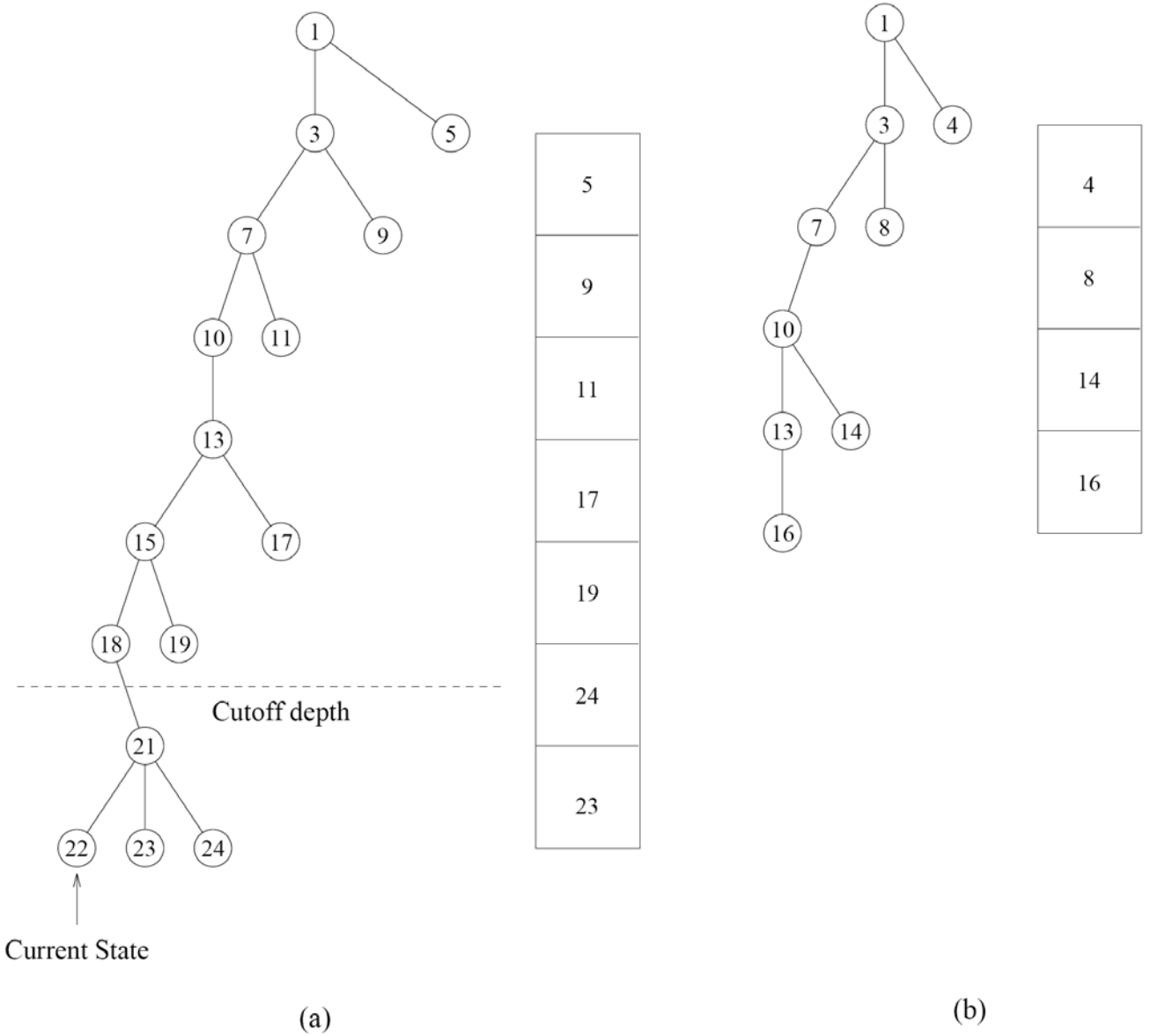

Figure 11.9 Splitting the DFS tree in Figure 11.5. The two subtrees along with their stack representations are shown in (a) and (b).

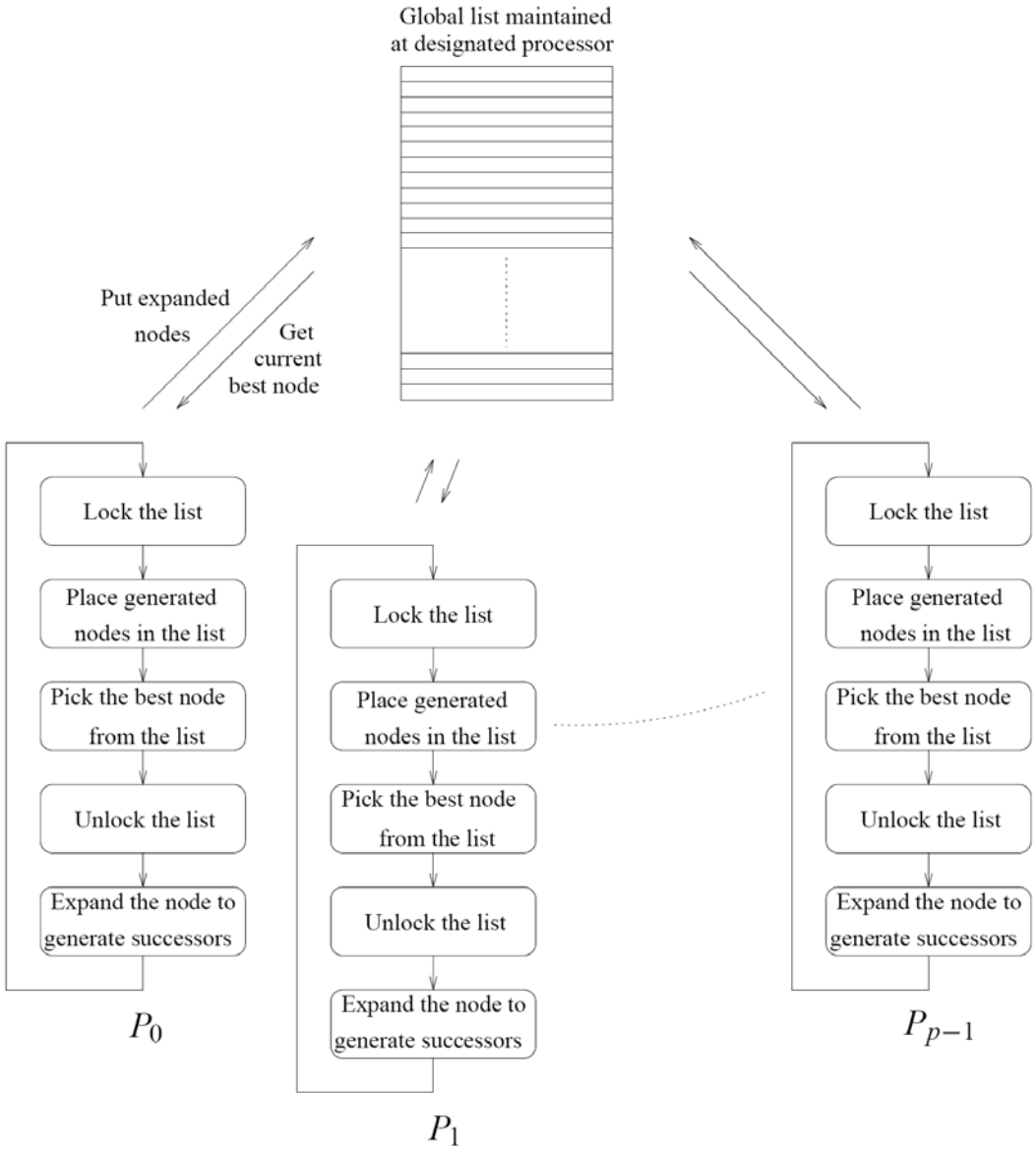

Figure 11.14 A general schematic for parallel best-first search using a centralized strategy. The locking operation is used here to serialize queue access by various processors.

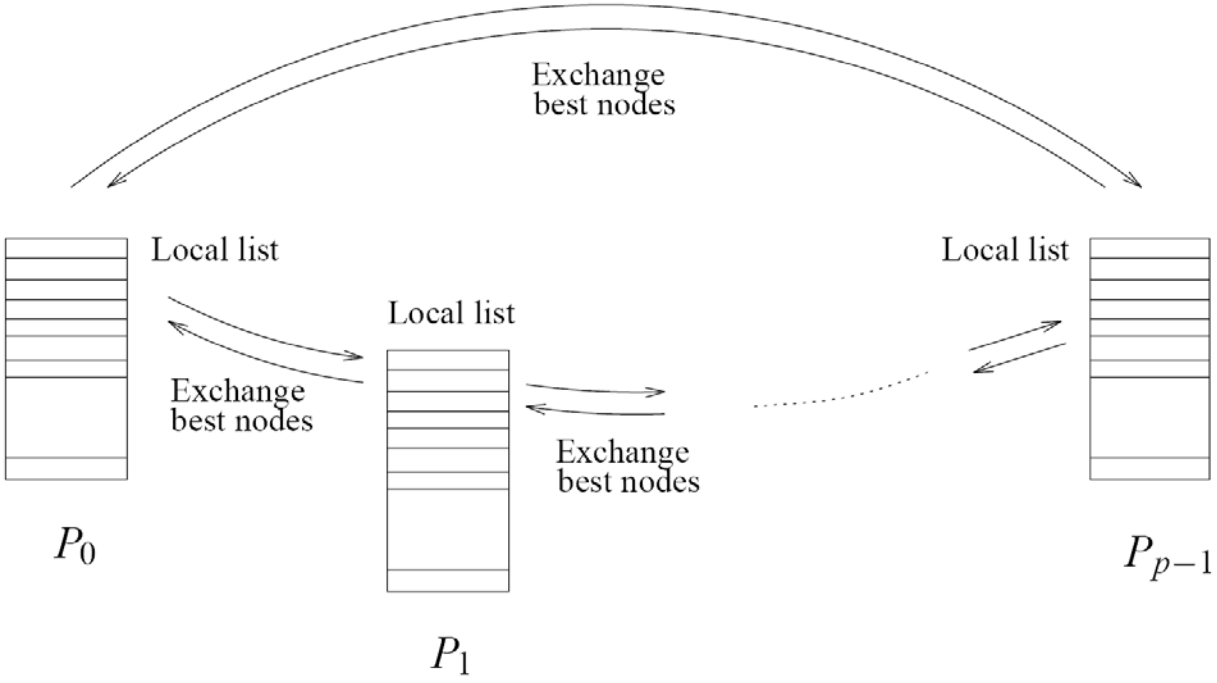

Figure 11.15 A message-passing implementation of parallel best-first search using the ring communication strategy.

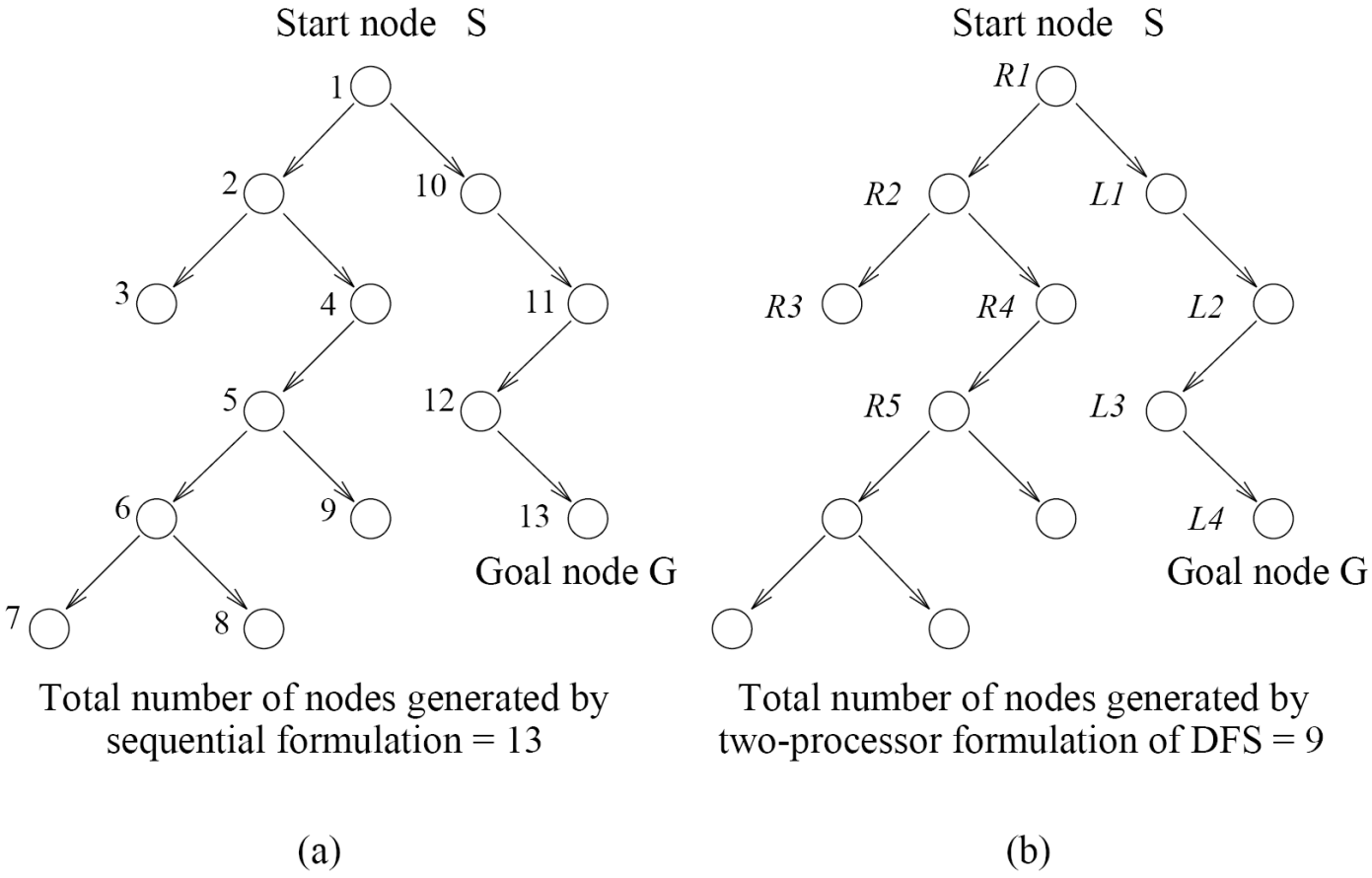

Figure 11.17 The difference in number of nodes searched by sequential and parallel formulations of DFS. For this example, parallel DFS reaches a goal node after searching fewer nodes than sequential DFS.

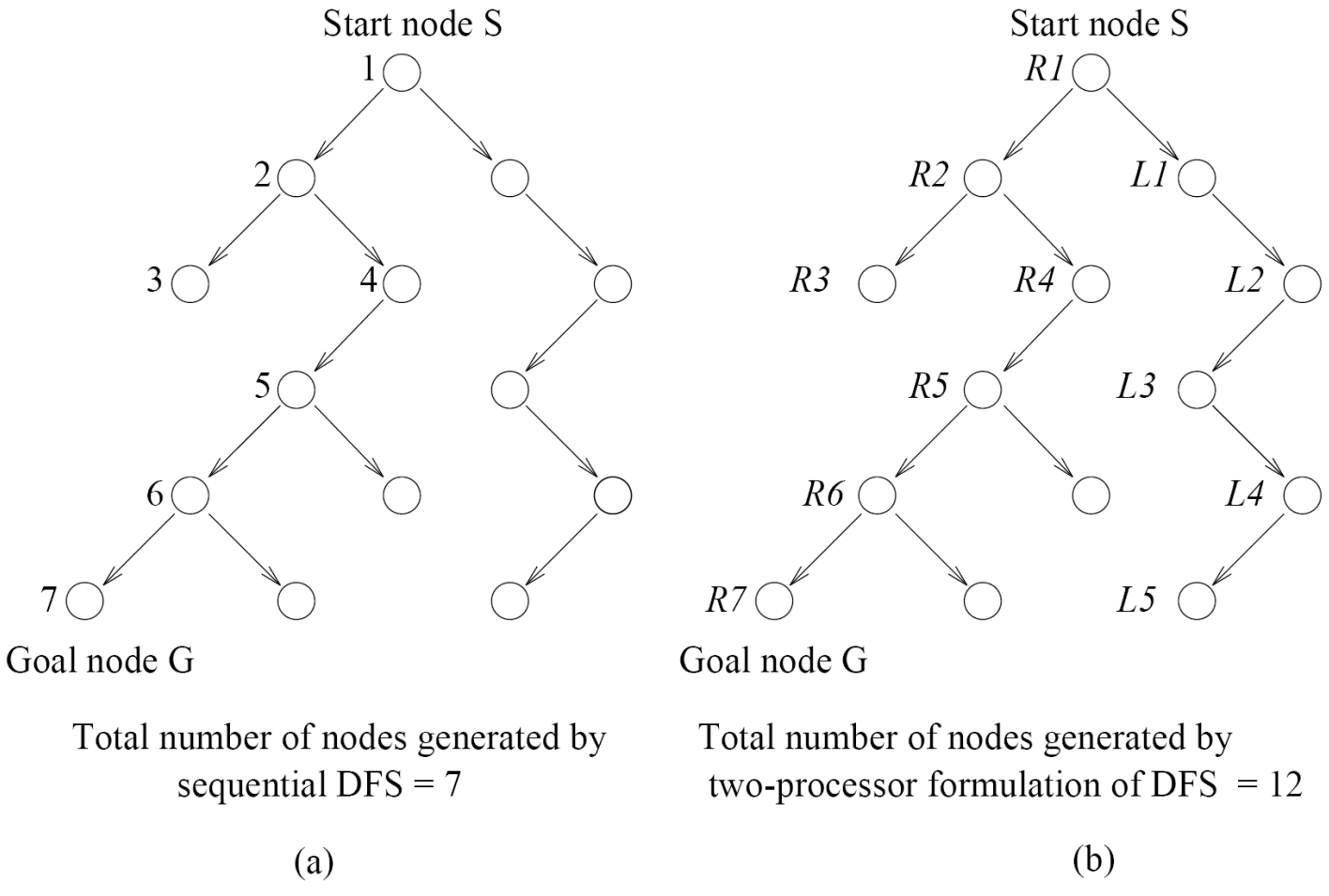

Figure 11.18 A parallel DFS formulation that searches more nodes than its sequential counterpart.