The time-scaling algorithm finds the time-optimal trajectory along a given path. What if our real goal is to find the time-optimal trajectory between two states when we are free to choose any collision-free path? Can we use the time-scaling algorithm in conjunction with a collision-free path planner? Conceptually, imagine running the time-scaling algorithm on all possible paths between the start and goal states. Then the fastest of these is the global time-optimal trajectory. Naturally, the problem is how to efficiently test a large number of possible paths. One approach to this problem for robot manipulators was proposed by Shiller and Dubowsky [383]. The approach is quite involved, and we only sketch it here. The first step is to define a grid on the workspace and construct all collision-free paths (without sharp turns) between the start and goal states moving along edges or diagonals of the grid. The next step is to quickly compute rough lower-bound estimates of the traveling times of these paths using a maximum velocity limit during the motion. The fastest paths are selected and smoothed by using the grid points as control points for cubic splines. The best of these paths is then submitted to the full time-scaling algorithm, generating an upper bound on the optimal travel time. All paths with lower bounds above this upper bound can be pruned. Of the remaining paths, the lower bounds are more carefully calculated, and the process continues, using increasingly accurate estimates of the lower bounds as the number of candidate paths is reduced. When the pruning process has ended, only the best path in each path-neighborhood is considered further. These best paths are submitted to a local optimization that may locally alter the paths to allow them to be executed more quickly. This process uses the travel times returned by the time-scaling algorithm as the objective function. This approach combines collision-free path planning and time scaling in an iterative fashion to arrive at a global near-time-optimal trajectory. In the next section,we discuss methods that do not separate the path-planning and time-scaling problems, but solve directly in the state space..