We developed a sampling-based motion planning algo-
rithm that combines long-term temporal logic goals with short-term reactive requirements. The mission specifi-
cation has two parts: (1) a global specification given as a
Linear Temporal Logic (LTL) formula over a set of static
service requests that occur at the regions of a known en-
vironment, and (2) a local specification that requires ser-
vicing a set of dynamic requests that can be sensed lo-
dally during the execution. The proposed computational
framework consists of two main ingredients: (a) an off-line
sampling-based algorithm for the construction of a global
transition system that contains a path satisfying the LTL
formula, and (b) an on-line sampling-based algorithm to
generate paths that service the local requests, while mak-
ing sure that the satisfaction of the global specification is
not affected. The off-line algorithm has three main fea-
tures. First, it is incremental, in the sense that the pro-
cedure for finding a satisfying path at each iteration scales
only with the number of new samples generated at that iter-
ation. Second, the underlying graph is sparse, which guar-
antees the low complexity of the overall method. Third,
it is probabilistically complete, i.e. the probability of not
finding a satisfiable path approaches 1 as the number of
samples increases.

On-line Algorithm

The on-line planning algorithm is based on RRT, which we
do in order to find local paths which preserve the sat-
isfaction of the global specification \( \Phi_{\text{G}} \), while servicing
on-line requests and avoiding locally sensed obstacles.

To keep track of validity of samples (random configura-
tions) with respect to the global specification \( \Phi_{\text{G}} \), we pro-
pose a method that combines the ideas presented in [5]
on monitors for LTL formulae and [3] on potential functions.
Monitoring functions are used to determine whether an infin-
itely satisfiable or violates an LTL formula based on a finite prefix
of the state sequence. We also show that this is the best possible complexity for sparse transition systems among incremental algorithms, which have a "locally" property [4].

Planning Algorithm

Given the global LTL- \( \Phi_{\text{G}} \) specification \( \Phi_{\text{G}} \), the priority func-
tion for on-line requests \( \text{prior} \) and the initial configur-
ation \( s_0 \), the steps of the path planning procedure are:

1. Convert \( \Phi_{\text{G}} \) to Büchi automaton \( \mathcal{B} \).
2. Compute \( T_0 \) and \( \delta_0 \) at \( s_0 \) using the RGG.
3. Compute potential function \( \Phi_{\text{G}}^*(s) \).
4. path ← emptyList(); \( x \leftarrow s_0 \), \( x \leftarrow x/\Phi_{\text{G}}^*(x) \).
5. Repeat indefinitely:
   - \( 5.1: \text{getLocalRequests}() \)
   - \( 5.2: \text{if} \) (checkPath( path ) \( \land \) (path hasNext() ))
     - \( \text{path} ← \text{planLocally}(s, \text{path}) \).
   - \( 5.3: x \leftarrow \text{path.nextState}() ; \text{enforce} x/\Phi_{\text{G}}^*(x) \).

In the above procedure, local paths are generated by the
RRT-based procedure \( \text{planLocally} (s) \) such that they satisfy
\( \Phi_{\text{G}} \). This requirement is achieved by tracking of Büchi
states for local samples and connecting the leafs of local
tree to states in \( T_0 \) which have (finite) minimum po-
tential after traversing the corresponding branch of the lo-
tal tree. Also, the line segment between the leaf state from
the tree and the state in \( T_0 \) must be collision free w.r.t. local
obstacles.

Theorem: Using the above planning procedure, the re-
turned infinite path \( x = x_1 \ldots \) satisfies the global miss-
ion specification \( \Phi_{\text{G}} \) if every call of the local planner
\( \text{planLocally} (s) \) terminates in finite time.

References


Acknowledgment

This work was partially supported by the ONR under grants MURI N00014-09-1051 and MURI N00014-10-10952 and by the NSF under grant NSF CNS-1305388.