# Adaptive Sampling for Generalized Sampling Based Motion Planners

Sandip Kumar and Suman Chakravorty

***Abstract*— In this paper, an *Adaptive Sampling* strategy is presented for the generalized sampling based motion planner, Generalized Probabilistic Roadmap (GPRM) [18, 19]. These planners are designed to account for stochastic map and model uncertainty and provide a feedback solution to the motion planning problem. Sampling intelligently, in this framework, can result in huge speedups when compared to naive uniform sampling. By using the information of transition probabilities, encoded in these generalized planners, the proposed strategy biases sampling to improve the efficiency of sampling, and increase the overall success probability of GPRM. The strategy was used to solve the motion planning problem of a fully actuated point robot on several maps of varying difficulty levels, and results show that the strategy helps solve the problem efficiently while simultaneously increasing the success probability of the solution. Results also show that these rewards increase with an increase in map complexity.**

## I. INTRODUCTION

The general motion planning problem in robotics is to find a collision free path for a robot from one configuration to another, in a given obstacle space.

Exact planners are intractable for most practical problems because the complexity grows exponentially with the dimensionality of the problem [1]. Randomized Sampling based methods were introduced to provide approximate solutions, while avoiding the prohibitive cost of computing the exact representation of the free space. Probabilistic Roadmaps (PRM) are one of the most successful sampling based methods, which sample the domain in a random fashion and build a roadmap over these samples to represent the free space [9]. To address highly constrained motions and domains, a key idea is to bias the sampling towards good regions of the configuration space, and various different sampling strategies to do the same have been proposed. These planners make local hypothesis that identify poor visibility regions [13] in the free space. Some use the information of workspace geometry, broadly categorized as *Workspace based sampling strategies*. Techniques in this category are watershed labeling algorithm [2], workspace importance sampling [3], and medial axis sampling [4, 5]. Some use geometric patterns and reject unpromising samples, categorized as *Filtering based sampling strategies*. Techniques under this category are Gaussian strategy [6], bridge test [7], and Vis-PRM [8]. Some use information gained during the roadmap construction, categorized as *Adaptive Sampling Strategies* and techniques include two-phase connectivity expansion strategies [9], and multiphase sampling [10]. There also exists a *Deformation Strategy for Sampling*, which tries to deform the domain into a more expansive domain [11]. These strategies spend more time generating a node when compared to a naive uniform sampling, with the expectation that a much smaller roadmap is required to answer queries, resulting in faster computation time. These strategies were studied and analyzed in refs. [12, 13, 14, 15] where various measures/ metrics such as *connectivity*, *coverage* and *completeness* were proposed to evaluate their effectiveness. In ref. [16], an attempt is made to provide metrics for sampling process during the roadmap construction. Thus, sampling intelligently can achieve huge speedup when compared to naive uniform sampling.

Unfortunately PRM and its variants work in the deterministic framework, and with the introduction of map and robot model uncertainty into the problem, the technique is no longer applicable. Furthermore, PRM does not account for the dynamics of the robot. Rapidly-exploring random trees (RRTs) incorporate randomized sampling of the domain, as in PRM, while also incorporating the dynamics of the robot while planning [17]. However, RRTs are open loop planners, as are PRMs, and thus cannot handle map and model uncertainty. The generalized sampling based motion planners, *Generalized-PRM* (GPRM) and *Generalized-RRT* (GRRT), were introduced to incorporate stochastic models of map and model uncertainty along with the dynamical constraints of the robot, and provide a feedback solution to the motion planning problem [18], [19]. We would like to mention other attempts to generalize PRMs and RRTs to handle map uncertainty [20, 21, 22, 23]. However, none of these techniques provide a feedback solution to the planning problem and are brittle under model uncertainty.

In this paper we introduce a novel strategy for adaptive sampling in GPRM. The strategy proposed here incorporates the information of the probabilities carried by the connections in GPRM. With this extra information, which is unique to planners dealing with uncertainty, the sampling strategy biases the samples such that the efficiency and the overall success probability for the planning increases in GPRM. We show that motion planning problem on complex maps can be efficiently solved using GPRM, in conjunction with the adaptive sampling strategy, while simultaneously increasing the success probability of the solution.

The rest of the paper is organized as follows. Section II discusses the Hierarchical methods and the methodology of the GPRM algorithm in brief, section III introduces conceptualization, development and the algorithm of the *Adaptive Sampling* Strategy for GPRM. Section IV discusses

Sandip Kumar is a Graduate Student in Department of Aerospace Engineering, Texas A&M University, College Station, TX - 77840, USA to.sandip@tamu.edu

Suman Chakravorty is a Faculty of Department of Aerospace Engineering, Texas A&M University, College Station, TX - 77840, USA schakrav@aeromail.tamu.edu

the application of GPRM along with Adaptive Sampling on a dynamical system along with results.

## II. Generalized Sampling Based Methods

The basic motion planning problem is to find a collision free path for a robot in a given obstacle space. With the introduction of map and model uncertainty, one can no longer have the same formulation of the motion planning problem. In the presence of stochastic model uncertainty, there is a need for feedback control, which is then associated with a probability that the robot reaches the goal without hitting the obstacles. Generalized Sampling Based Algorithms [18, 19] were introduced to address the problem of feedback motion planning in such constrained work spaces. Before going into the details of the methodology of [18, 19], we note that the complexity of the motion planning problem has increased due to:

- Introduction of model uncertainty in the dynamics of the robot, which implies that we have to obtain satisfactory performance over an ensemble of paths instead of a single path.
- Introduction of map uncertainty, implies the planner has to succeed for an ensemble of maps.

The notion of collision avoidance and collision-free paths as the solution to the motion planning problem, can no longer be satisfied, and therefore the above criteria need to be replaced by a solution/ path with a high probability of success. The motion planning problem can be re-framed as : *To solve the motion planning problem in the presence of map uncertainty and model uncertainty, generate a feedback solution with a probability of success above an a-priori specified probability,* $p_{min}$.

### A. Hierarchical Methods and Generalized Probabilistic Roadmaps (GPRM)

If the uncertainties in the robot model and environment can be modeled probabilistically, the robot motion planning problem can be formulated as Markov Decision Problem (MDP) [24]. These MDPs are computationally intractable for anything but small state/ control spaces and especially hard to solve in continuous state and control spaces. Hierarchical Methods can be used to break down the complexity of the problem. The *Generalized Probabilistic Roadmaps*(GPRM) [18, 19], is a sampling based hierarchical method which extends the *Probabilistic Roadmaps* (PRM) [9] technique for deterministic path planning, to systems with stochastic model and map uncertainty.

In the following paragraph we briefly introduce GPRM, more details can be found in [18], [http://dnc.tamu.edu/wiki/images/4/44/Paperj_GPRM.pdf].

The state of the robot is given by $x = (q, \dot{q})$, where $q$ represents configuration of the robot and $\dot{q}$ the generalized velocities. The free region in the map corresponds to a free region in the configuration space, $C_{free}$, which induces a free region in the state-space of the robot, say $\chi_{free}$. GPRM samples equilibrium states (i.e. state wherein the velocities are zero) in $\chi_{free}$, which are called *landmarks*.

The planning problem of guiding the robot from the *start* landmark to the *goal* landmark is divided into two hierarchical levels. The lower level planner guides the robot from one landmark to another using feedback control and accounts for the model uncertainty in the robot dynamics, specified by the following equation:

$$\dot{x} = f(x) + g(x)u + h(x)w \tag{1}$$

where $x$ is the state of the robot, $w$ is a white noise perturbation, and $u$ is the control. However, the control does not account for constraints, i.e. obstacles in the map, which are specified by $p(O/y)$, the probability that a point $y$ in map is *occupied*. The interaction between feedback planner and the obstacles in the map result in a *transition probability* and *transition cost* for the robot from one landmark to next. Figure 1 depicts a sample path between the landmarks $s$ and $r$ given a feedback controller $u$ that guides it towards $r$. The control $u$ at state $s$, denotes the next landmark among the $k$-nearest neighbors of $s$ that the robot is guided to. The transition probability, of the path, $p_{s,r}$ is given by :

$$p_{s,r} = \prod_y (1 - p(O/y)) \tag{2}$$

where $y$ represents the grids along the path. A *failure state*, say $x_{fail}$, is introduced and $1 - p_{s,r}$ is the probability of landing in the failure state. The transition cost, $c_{s,r}$ is directly proportional to the probability of transitioning to the failure state, $x_{fail}$. Due to the presence of model uncertainty, the average cost $c(s, u)$ and transition probability $p(r/s, u)$ have to be formed by averaging over all such sample paths. This is achieved using Monte Carlo simulations.

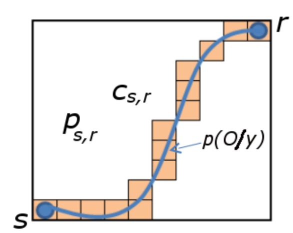



Fig. 1. *Transition cost* and *transition probability*

The top level planner, works on global map in the landmark space. It uses the information of the metrics of the lower level planner, minimizes the cost-to-go from each landmarks over all possible policies, and gives the optimal control policy over the landmark map.

The optimal control action $u^*(\cdot)$ for each state/ landmark of the map is the outcome of the top level planner. The optimal cost-to-go $J^*(\cdot)$, required in calculation of $u^*(\cdot)$, is found as the solution of the Bellman fixed point equation/ Dynamic Programming equation :

$$J^*(s) = \min_u \{c(s, u) + \sum_r (p(r/s, u)J^*(r)\} \tag{3}$$

$$u^*(s) = \operatorname*{argmin}_u \{c(s, u) + \sum_r p(r/s, u)J^*(r)\} \tag{4}$$

where $J^*(s)$ is the optimal cost-to-go from state $s$, $u^*(s)$ is the optimal control action to be taken at state $s$. Here, control $u$ at state $s$ is the next landmark among the $k$-nearest neighbors of $s$ that the robot is guided to, $p(r/s, u)$ is the probability of transition from $s \rightarrow r$ given the robot is guided towards the landmark specified by $u$, and $c(s, u)$ is the cost of transition. Note that $p(r/s, u)$ and $c(s, u)$ are got by evaluating the lower level feedback planner. The details of the algorithm and the calculation of metrics are in [18, 19].

### B. Algorithm GPRM

The pseudo-code for the generalized probabilistic roadmaps (GPRM) algorithm is shown below:

**Algorithm GPRM**

- Given : $x_0$, the starting point, $x_g$, the goal point of the robot, and $p_{min}$, the minimum probability of success
- Initialize GPRM with the nodes $x_0$ and $x_g$
  1) Sample the equilibrium states in $\chi_{free}$ probabilistically using a uniform distribution
  2) Build the connectivity graph, i.e. connect every sampled state in the domain with its $k$-nearest neighbors using suitable obstacle-free feedback controllers
  3) Evaluate the cost and transition probability associated with every connection in the resulting connectivity graph using Monte Carlo Simulations
  4) Plan on the resulting *SMDP* using evaluated edge cost and transition probability from step 3
  5) Evaluate the probability of success, $p_s$, of the resulting path from step 4. If $p_s > p_{min}$, end; else go to step 1
- End

A few points have to be made regarding the feedback controllers specified in step 2 above :

- Due to model uncertainty present in the dynamical system, it is impossible to control the robot exactly to the point $x_g$ even in the absence of obstacles
- In the case of stochastic systems, a feedback controller is necessary due to uncertainty. The feedback controller ensures that even in presence of uncertainty in the model, the robot reaches a neighborhood of the target equilibrium state with a high probability, in the absence of obstacles
- The feedback controller is designed for a workspace without any obstacles as otherwise the controller design is complicated

The feedback controller can be designed in many ways. For linear systems LQR based controllers can be used. For non-linear system, the system can be linearized about an equilibrium point and a feedback controller can be designed for the linearized system. Other non-linear feedback controllers may also be used, such as the dynamic feedback controller used for the non-holonomic system in [19]. The feedback controller operates between one landmark and another and in presence of model uncertainty ensures the robot reaches a neighborhood of the target equilibrium state in the absence of obstacles. In the presence of obstacles, Monte Carlo simulations are used to compute the *transition probability* and *transition cost* in using the feedback planner to guide the robot from one landmark to another. The feedback controller used in the work presented here is state-feedback based LQR controller.

The GPRM is capable of handling model and map uncertainty as discussed above, but as the complexity of the map, i.e the size of the map and the clutter of the obstacles increase, the number of landmarks required to find a solution becomes large, thereby greatly increasing the computational resources required. A logical extension for complicated maps is to sample in areas where samples are required, i.e. use an adaptive sampling strategy. The next section describes such an adaptive sampling algorithm.

## III. ADAPTIVE SAMPLING

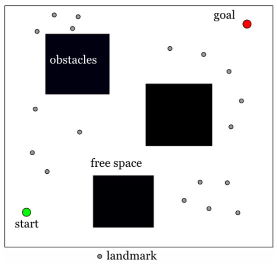
In sampling based motion planning algorithms, the number of samples determine the complexity of computation required to solve the problem. For a complex domain, a naive uniform sampling will require a large number of samples and hence, more computational resources. Introduction of adaptive sampling adds intelligence to the planning algorithm, by efficiently adding new samples.

### A. Adaptive Sampling Details

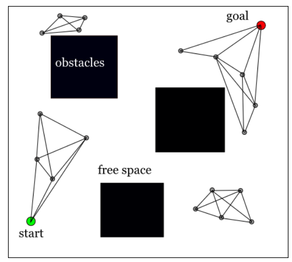
In a sampling based motion planners framework, coordinates of configuration space are sampled in random fashion which is mapped into the obstacle space as shown in Figure 2(a) (they are referred to as equilibrium states, $x_g$ or *landmarks* in GPRM framework). A connectivity graph is constructed over the landmarks as shown in Figure 2(b), it is based on the cost and transition probabilities computed from the lower level feedback planner used in GPRM.
Using the information encoded in the connectivity graph, we introduce the major ingredients of the adaptive sampling strategy in the following.

*1) Identification of Start and Goal Clouds:* A cloud of samples is referred to as a collection of landmarks which are inter-connected with transition probabilities higher than $p_{min}$, in the connectivity graph of the map/ obstacle space. The start and the goal clouds are the cloud of samples containing the start and the goal (or end) landmarks respectively. The motion planning problem, in the generalized sampling based motion planning framework, is to find a path[1] with success probability higher than $p_{min}$ between the start and the goal landmark. The idea is to identify the cloud of samples as shown in Figure 2(c) containing the *start landmark state* and the *goal landmark state* and try to connect them during the re-sampling phase and hence solve the motion planning problem.

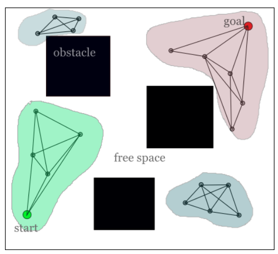
[1]A path here implies a local feedback controller guides the robot from one landmark to another, while the higher level planner guides the robot regarding the landmark, to go to next.




(a) Landmarks

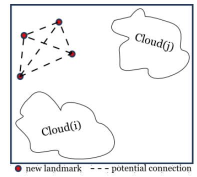



(b) Connections

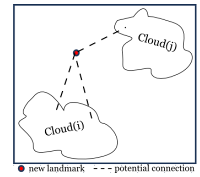



(c) Cloud of samples

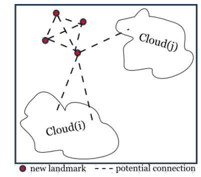
Fig. 2. Problem Domain with free space, obstacles, start, goal positions



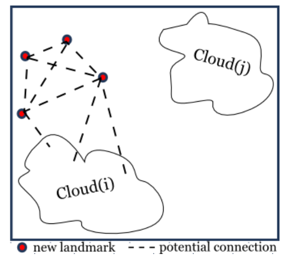

(a) $C_1$



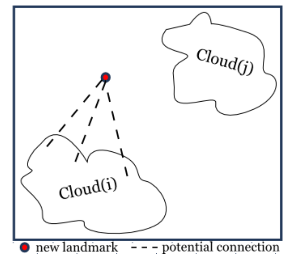

(b) $C_2$




(c) $C_3$




(d) $C_4$




(e) $C_5$

Fig. 3. Categories of New landmarks sampled

To identify these clouds the information carried by the connectivity graph is used. The connectivity graphs in the generalized motion planner framework encode both the *transition cost* and the *transition probability* information.

We assign *goal proximity probability*, $\bar{p}_g(x)$, and *start proximity probability*, $\bar{p}_s(x)$, to each of the landmarks, $x$. Proximity probability is a metric defined between two landmark states (say $x_a$ and $x_b$), and it carries the information of the probability of transition from state $x_a$ to $x_b$, given by $\bar{p}_b(x_a)$, and vice-versa, $\bar{p}_a(x_b)$. The goal proximity probability, $\bar{p}_g(x)$ is defined as the proximity probability between a landmark, $x$, and the goal landmark state, $x_g$, along with a constraint that $\bar{p}_g(x) > p_{min}$, where $p_{min}$ is given. It is calculated by traversing from the goal landmark, $x_g$ to the concerned landmark, $x$, keeping track of all the transition probability in the path. Similarly, the start proximity probability $\bar{p}_s(x)$ is calculated by traversing from the start landmark, $x_s$, to the concerned landmark, $x$, and keeping track of the transition probabilities along the way. These metrics, once calculated, will suggest landmarks which are connected to the goal and the start landmarks, with a overall transition probability greater than the threshold probability ($p_{min}$) of the domain. In this way the cloud of samples connected to the start and the goal with a high transition probability can be computed.

*2) Identification of other clouds:* We also want to identify clouds other than the start and the goal clouds that are present in the workspace. We compute this information to identify and differentiate between the *good* and *bad* samples. These *good* and *bad* samples will be discussed in item 3 below.

The process of computing the information about clouds can be stated as:

- pick a landmark, $x$, and assign a group identification, $g_{id}(x)$, representing cloud information
- all the directly and indirectly connected landmarks are assigned the same group identification, $g_{id}(\cdot)$
- restart the process with a new landmark which has no assigned group yet
- continue till all the landmarks are covered, i.e each landmark has a assigned group identification, $g_{id}(\cdot)$

*3) Sampling* ***good*** *landmarks:* Sampling of landmarks in the configuration space is usually done using uniform sampling over the configuration space, when no knowledge is available to bias the sampling. Sampling ***good*** landmarks $x_{good}$ involves sampling landmarks which have the potential to solve the motion planning problem, or about rejecting the bad landmarks $x_{bad}$ from a set of sampled landmarks. We re-sample the space, i.e. generate a set of new landmarks, $X_{new}$, and find the *k*-nearest neighbors of each landmark in $X_{new}$. Based on the potential connections[2], every new landmark can be categorized completely, refer Figure 3, as a landmark whose neighbors:

i) $\in X_{new}$ only, the set of new landmarks generated, (refer Figure 3(a))
ii) $\in$ different clouds, (refer Figure 3(b))
iii) $\in$ different clouds and $X_{new}$, (refer Figure 3(c))
iv) $\in X_{new}$ and a specific cloud, (refer Figure 3(d))
v) $\in$ a specific cloud only, (refer Figure 3(e))

Samples in $Category\ (iv)\ and\ (v)$ are categorized as "bad" since obviously they have minimal potential to solve the problem. Hence, using the cloud information we reject the identified bad samples.

[2]The connections with *k*-nearest neighbors, prior to computing the transition probabilities, which either establishes a connection or discards it.

*4) Identifying Weak Link / Links in a Low Probability Connected Path:* The connectivity graph of a map has the *transition cost* and *transition probability* information. In contrast, in the deterministic framework of sampling based motion planners, such as PRM, these graphs only carry the *transition cost* information. In GPRM the top level planner searches for a high probability path over the domain, and returns a path connecting the start landmark $x_s$ and the goal landmark $x_g$, and a success probability associated with it, say $p_{path}$. There could be cases where in spite of the connectivity, the $p_{path}$ is less than the desired threshold success probability $p_{min}$. Such an outcome can be used as a starting point for finding a neighboring path, $path'$ with a path probability $p_{path'}$, which has a success probability $p_{path'} > p_{min}$. We identify the *weak link / links*[3] of the low probability path and then sample around these in search of $path'$. Finding a neighboring path with a higher probability of success in the vicinity of a low-probability path may not always be feasible, as has been experienced during numerical simulations, but results show that the technique works fine most of the time.

Based on the ingredients of the Adaptive Sampling Strategy as described above, the algorithm can be summarized as follows :

**Algorithm Adaptive Sampling GPRM**

Step 1 Invoke GPRM over the given map initially with a small number of randomly selected landmarks and $p_{min}$

Step 2 If a path with high success probability found STOP, else start Adaptive Sampling

Step 3 Assign start proximity $\bar{p}_s(\cdot)$ and goal proximity probability $\bar{p}_g(\cdot)$ to all the landmarks

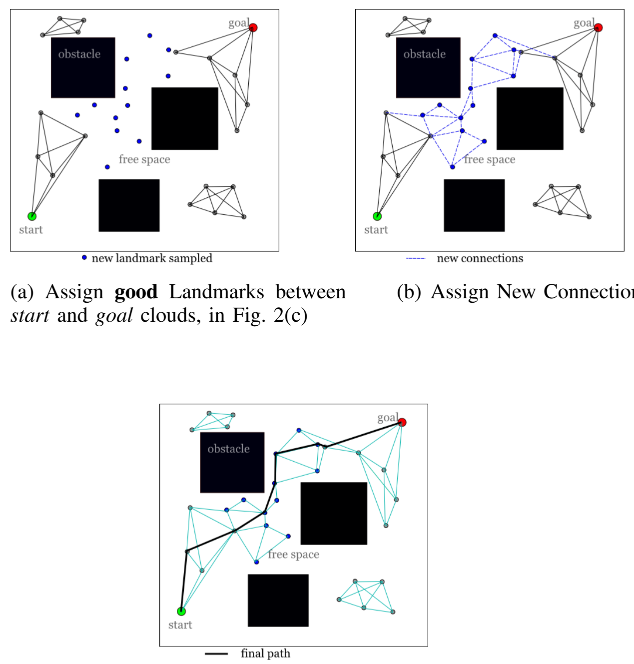
Step 4 Identify the landmarks connected with high probability to the start landmark and to the goal landmark, also identify the different clouds in the connectivity graph, using the group identification $g_{id}(\cdot)$

Step 5 Pick a pair of landmarks, one having high $\bar{p}_g(\cdot)$ and another having high $\bar{p}_s(\cdot)$

1) Sample[4] between these landmarks
2) Identify bad samples $x_{bad}$ and reject them
3) With the remaining good samples, perform GPRM
4) IF a path with high success probability found STOP, ELSE IF no solution, discard the samples added, go to next pair of landmarks in Step 5
5) ELSE IF a low probability path found
   a) Find weak link / links in the low probability path
   b) Sample[5] between the pair of landmark associated with the weak link
   c) Remove bad samples, and perform GPRM
   d) If a path with $p_{path} > p_{min}$ found STOP, else GOTO Step 3

Step 6 End

The Figure 4 depicts in brief the stages in the adaptive sampling methodology.




(a) Assign **good** Landmarks between *start* and *goal* clouds, in Fig. 2(c)

(b) Assign New Connections

(c) The final path

Fig. 4. Adaptive Sampling in steps, (build-up on Fig. 2)

### *B. Resolution Complete*

Sampling based motion planners such as PRM and GPRM are focused on solving multiple queries[6] on the domain. A major criteria, as stated in [12], for the success of randomized sampling based methods is that the connectivity graph for the map has to be *resolution complete*, i.e. any valid query can always be solved via the roadmap. In order to achieve a resolution complete roadmap, multiple randomly generated queries[7] are solved over the map using GPRM with *Adaptive Sampling*. The resolution completeness of the roadmap has been depicted in Figure 5, and further results are given in Section IV.

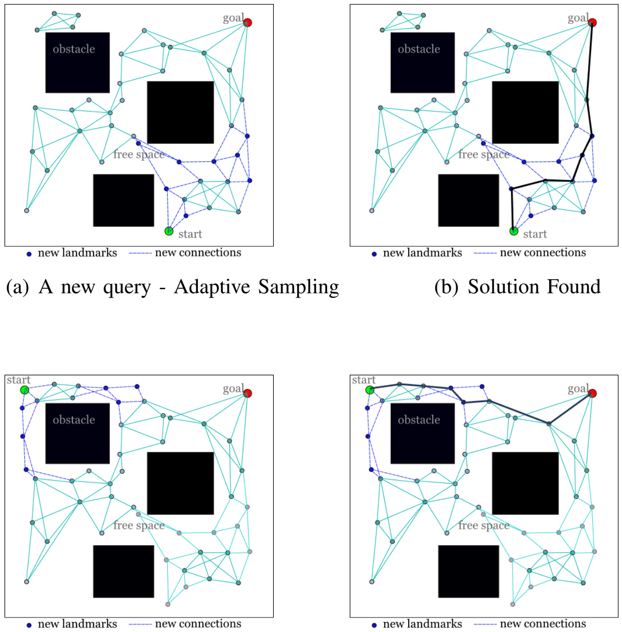
[3]Connections in the connectivity graph, which are responsible for low success probability of the path.

[4]In simulations the samples were drawn from a Gaussian distribution, with mean placed at the arithmetic mean of generalized positions of start and goal landmark and the standard deviation $\sigma$ being 2-norm of the distance between start and mean.

[5]In simulations, a biased distribution was assumed, i.e. an elliptical distribution with major axis aligned along start and goal configurations.

[6]A query is to find a path between initial and goal configuration. Multiple queries is to find paths between multiple initial and goal configurations.

[7]Within the context of the GPRM aided by Adaptive Sampling, a new query is a new start configuration.




(a) A new query - Adaptive Sampling (b) Solution Found

(c) Another query - Adaptive Sampling (d) Solution Found

Fig. 5. Multiple Query - Resolution Complete

## IV. Results and Discussion

The Adaptive Sampling methodology developed is applied to a fully actuated point robot:

$$\ddot{q} = u + w, \tag{5}$$

where $q$ is the generalized position vector of the robot, $u$ are the input forces and $w$ is a white noise term that quantifies the uncertainty in the motion model of the robot. This case was solved using basic GPRM, refer [18]. Numerical simulation results are presented for a set of maps with varying degrees of complexity. In general, the results indicate that:

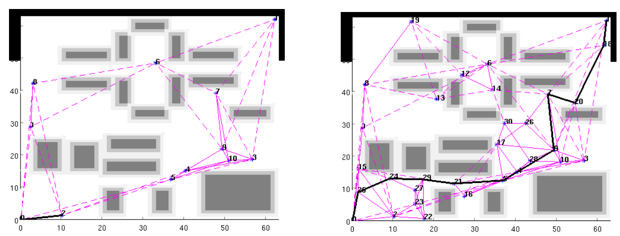
- The quality of sampling improved, i.e the landmarks were generated in required regions.
- The number of landmarks required to solve any complex map is approximately reduced to half the number required for solving the same map with basic GPRM, with progressively higher rewards in larger/ complex maps (refer Table I).

Each of the maps (Figure 6 - 9) discussed in the results section has two sub-figures : sub-figure (a) represents the initial landmarks the adaptive sampling starts with, and sub-figure (b) represents the final solution for the map with the additional landmarks sampled, and a path shown between the start and goal query. In the results shown in Figure 6, the final connectivity graph on the map shows the adaptive nature of the sampling done to solve the map. There are areas in the map where more sampling was done and areas where no sampling has been done. This is something to be expected from an adaptive sampling algorithm. Maps with more complexity were also solved and Figures 7 - 9 represent the solutions. Some maps have always challenged the sampling and motion planning algorithms, one of them is the single passage map, the solution to which is given in Figure 10. The algorithm was able to solve the map with minimal increase in landmarks for the map.

TABLE I

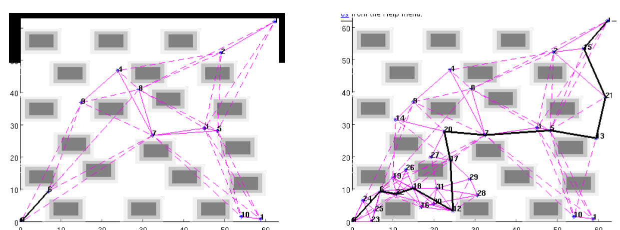
Result Comparison : **GPRM** with and without **Adaptive Sampling**

| Number of Samples Required ($p_{min} = 0.8$) | | | |
|---|---|---|---|
| Map # | basic GPRM | GPRM + Adaptive Sampling | Efficiency ($\eta$) |
| 1 | 62 ($p_s$ = 0.896) | 30 ($p_s$ = 1.0) | 2.07 |
| 3 | 72 ($p_s$ = 0.889) | 32 ($p_s$ = 0.889) | 2.25 |
| 5 | 72 ($p_s$ = 1.0) | 61 ($p_s$ = 1.0) | 1.18 |
| 6 | 162 ($p_s$ = 0.889) | 64 ($p_s$ = 0.889) | 2.53 |
| 10 | 182 ($p_s$ = 1.0) | 52 ($p_s$ = 1.0) | 3.50 |



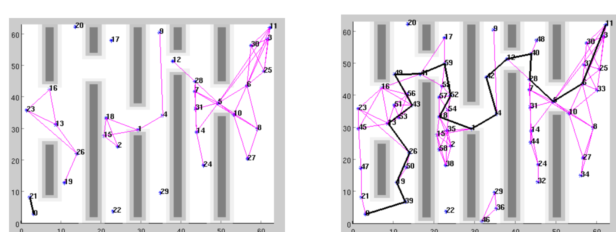
(a) Initial Landmarks (b) Query Solved

Fig. 6. Adaptive Sampling with GPRM : Map 1



(a) Initial Landmarks (b) Query Solved

Fig. 7. Adaptive Sampling with GPRM : Map 3

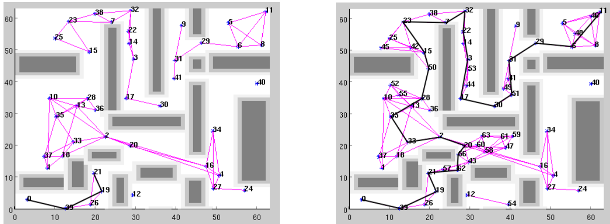


(a) Initial Landmarks (b) Query Solved

Fig. 8. Adaptive Sampling with GPRM : Map 5

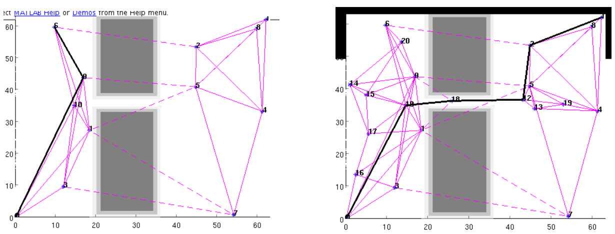
The maps discussed till now have dimensions 60 x 60 units. Figure 11 represents one of the largest map the algorithm was tried on, it is 150 x 150 units in area.

The maps discussed here were also solved using the basic GPRM algorithm and the results when compared with the



(a) Initial Landmarks (b) Query Solved

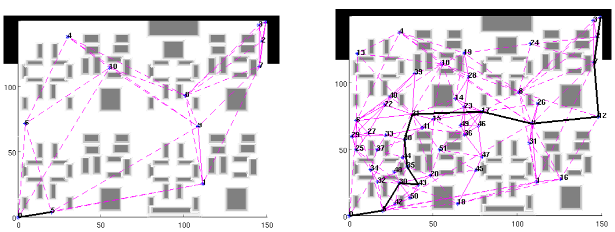
Fig. 9. Adaptive Sampling with GPRM : Map 6



(a) Initial Landmarks (b) Query Solved

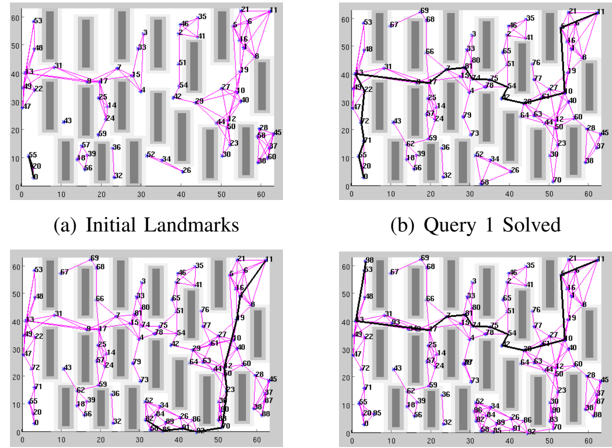
Fig. 10. Adaptive Sampling with GPRM : Map 9

adaptive sampling case, suggest that the number of samples required to solve the maps have approximately been reduced by half or more (refer Table I).



(a) Initial Landmarks (b) Query Solved

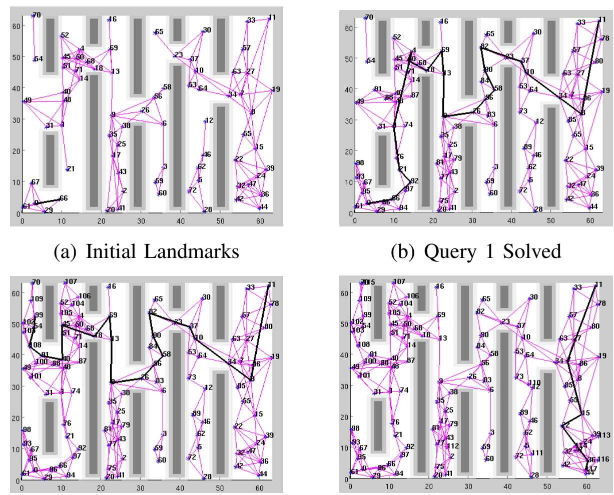
Fig. 11. Adaptive Sampling with GPRM : Map 10



(a) Initial Landmarks (b) Query 1 Solved

(c) Query 2 Solved (d) Query 3 Solved

Fig. 12. Multiple Query, Resolution Complete : Map 1



(a) Initial Landmarks (b) Query 1 Solved

(c) Query 2 Solved (d) Query 3 Solved

Fig. 13. Multiple Query, Resolution Complete : Map 2

The maps shown in the Figures 12 and 13 represent the final solution for the map after multiple queries have randomly been solved to achieve *Resolution Completeness* (Refer III-B). As discussed , one of the major requirement for algorithm trying to solve the multiple query case is that the final connectivity graph for the map should be resolution complete, i.e. be able to solve any query. These results show that GPRM aided with Adaptive Sampling, achieves *Resolution Complete roadmaps* for these maps.

## V. CONCLUSIONS

This paper presents an adaptive sampling strategy for the generalized sampling based motion planners framework. The strategy was tested on an idealized point robot with fully actuated dynamics with stochastic map and model uncertainty. The numerical simulations were done on several complicated maps. The results are promising when compared to basic GPRM, and suggests that a solution to complicated maps, where a basic GPRM might fail or would require a high number of landmarks, is possible with significantly less number of landmarks, using the adaptive sampling strategy. The next step is to test this adaptive sampling strategy along with GPRM on a high dimension system, such as an $n$-link manipulator or biological systems such as a robotic arm, in complex workspaces. The results here indicate that we have increased the efficiency of sampling, and the probability of success associated with the solution of GPRM algorithms.

## REFERENCES


[1] J. Reif, "Complexity of the mover's problem and generalizations extended abstract," in *Proceedings of the 20th Annual IEEE Conference on Foundations of Computer Science*, 1979, pp. 421–427.

[2] J. van den Berg and M. Overmars, "Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners," *The International Journal of Robotics Research*, vol. 24, no. 12, p. 1055, 2005.

[3] H. Kurniawati and D. Hsu, “Workspace importance sampling for probabilistic roadmap planning,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings*, vol. 2.

[4] L. Guibas, C. Holleman, and L. Kavraki, “A probabilistic roadmap planner for flexible objects with aworkspace medial-axis-based sampling approach,” in *1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999. IROS’99. Proceedings*, vol. 1, 1999.

[5] J. Lien, S. Thomas, and N. Amato, “A general framework for sampling on the medial axis of the free space,” in *IEEE International Conference on Robotics and Automation*, vol. 3. Citeseer, 2003, pp. 4439–4444.

[6] V. Boor, M. Overmars, and A. Van Der Stappen, “The gaussian sampling strategy for probabilistic roadmap planners,” in *1999 IEEE International Conference on Robotics and Automation, 1999. Proceedings*, vol. 2, 1999.

[7] D. Hsu, T. Jiang, J. Reif, and Z. Sun, “The bridge test for sampling narrow passages with probabilistic roadmap planners,” in *IEEE International Conference on Robotics and Automation*, vol. 3. Citeseer, 2003, pp. 4420–4426.

[8] T. Simeon, J. Laumond, and C. Nissoux, “Visibility-based probabilistic roadmaps for motion planning,” *Advanced Robotics*, vol. 14, no. 6, pp. 477–493, 2000.

[9] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, “Probabilistic roadmaps for path planning in high-dimensionalconfiguration spaces,” *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[10] M. Morales, L. Tapia, R. Pearce, S. Rodriguez, and N. Amato, “A machine learning approach for feature-sensitive motion planning,” *Algorithmic Foundations of Robotics VI*, pp. 361–376, 2004.

[11] D. Hsu, L. Kavraki, J. Latombe, R. Motwani, and S. Sorkin, “On finding narrow passages with probabilistic roadmap planners,” in *Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics*, 1998, pp. 141–154.

[12] R. Geraerts *et al.*, “Sampling-based motion planning: Analysis and path quality,” 2006.

[13] D. Hsu, J. Latombe, and H. Kurniawati, “On the probabilistic foundations of probabilistic roadmap planning,” *Robotics Research*, pp. 83–97.

[14] L. Kavraki, M. Kolountzakis, and J. Latombe, “Analysis of probabilistic roadmaps for path planning,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 166–171, 1998.

[15] S. Lindemann and S. LaValle, “Current issues in sampling-based motion planning,” *Robotics Research*, pp. 36–54.

[16] R. Pearce, M. Morales, and N. Amato, “Structural Improvement Filtering Strategy for PRM,” *Robotics: Science and Systems IV*, p. 167, 2009.

[17] S. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” *In*, 1998.

[18] S. Chakravorty, S. Kumar, *Generalized Sampling based Motion Planners*, http://dnc.tamu.edu/wiki/images/4/44/Paperj_GPRM.pdf.

[19] S. Chakravorty and S. Kumar, “Generalized sampling based motion planners with application to nonholonomic systems,” in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2009*, 2009, pp. 4077–4082.

[20] P. Missiuro and N. Roy, “Adapting probabilistic roadmaps to handle uncertain maps,” in *Proceedings of the International Conference on Robotics and Automation*, 2006.

[21] B. Burns and O. Brock, “Sampling-based motion planning with sensing uncertainty,” in *2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3313–3318.

[22] N. Melchior and R. Simmons, “Particle RRT for path planning with uncertainty,” in *2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 1617–1624.

[23] R. Alterovitz, T. Siméon, and K. Goldberg, “The stochastic motion roadmap: A sampling framework for planning with Markov motion uncertainty,” in *Robotics: Science and Systems*. Citeseer, 2007.

[24] S. LaValle, *Planning algorithms*. Cambridge Univ Pr, 2006.