CP230 FINAL PROJECT REPORT
Sampling-based motion planning with cross-entropy based stochastic optimization

Saharsh
RBCCPS
IISc
Bengaluru, India

Sandeep
RBCCPS
IISc
Bengaluru, India

Sam Sachin
RBCCPS
IISc
Bengaluru, India

***Abstract*—Traditional sampling based motion planning algorithms like RRT and RRT* uses uniform distribution for sampling the new points. Though these methods proved efficiency in terms of time for high dimensional spaces, optimal path guarantee can not be stated for RRT. This paper tackles this problem by optimizing the sampling distribution itself. We used RRT* with adaptive sampling distribution using Cross-entropy (CE) based optimization. State cross entropy (SCE) RRT* and Trajectory cross entropy (TCE) RRT* algorithms are introduced for obtaining the optimum sampling distribution, capable of producing minimum cost trajectories. We used two-link manipulator model for demonstrating how this works and the simulations matched the expectations of knowing what is the best distribution to sample from for optimal sampling points.**

Index Terms*—Cross-entropy, optimal sampling distribution, RRT

I. Introduction

A. MOTIVATION

Motion planning is an integral part of designing autonomous systems capable of complex tasks in constrained environment. Optimized motion planning helps us in obtaining minimum cost paths for efficient use of our resources, especially in case of robotics.At the same time, computationally efficient algorithms are very much important because of time constraints for performing the tasks in practical view. For dealing with later problem, a well known motion planning algorithm is RRT[5] which is very time efficient in high dimensional spaces with the trade-off being cost optimality. To tackle this cost optimality, an improvised version of this algorithm is RRT*[5] but then comes the trade-off with time. This trade-off is inevitable in this setting because the main reason being same uniform sampling distribution for both.

This makes us think about the distribution that we use for sampling. If we can modify the distribution to always sample the points around the optimal path, then both the problems of time and cost optimality can be dealt at the same time. This motivates towards the idea that is discussed in the paper.

B. RELATED WORK

Algorithms like dijkstra[5] and A*[5] works in discretized space which are very inefficient in high dimensional spaces. So sampling based techniques are hired for dealing with the same. Sampling based technique, RRT*, proved to be efficient in terms of cost optimality but with increased time complexity than RRT. Though there are modified versions of RRT like [4],[6],[7] for providing optimality both in terms of time and cost, much time is wasted in sampling unused nodes because of the uniform sampling distribution.

For this purpose, biased sampling can be employed and is not new in motion planning. In addition, learning from previous motion planning runs is also studied[8].The idea that this paper discusses is optimizing the sampling distribution itself using stochastic optimization techniques in this sampling based planning context. CE method is employed for this purpose in the paper.

C. CONTRIBUTION

In cross-entropy stochastic optimization, we are estimating the optimal sampling distribution for minimum cost trajectories by considering low cost trajectories as a rare event among samples of trajectories obtained. We use Gaussian mixture model (GMM) as our approximate distribution to estimate the actual optimal distribution to sample nodes from, which we don't know. GMM 's are great choice because they are proved to be the universal density approximators[9]. We used goal-bias sampling method in RRT* for obtaining large no. of samples of trajectories for training our GMM model.

Rest of the paper is structure as follows,We stated our problem statement in section 2, In section 3, we introduced CE based stochastic optimization of sampling distribution for motion planning. In section 4,5, we introduced the GMM model structure and SCE RRT* and TCE RRT* algorithm. Convergence and termination conditions are given in section 6. The simulation results of two-link manipulator with SCE RRT* algorithm is given in section 7.

II. PROBLEM FORMULATION

Our goal is to find the best parameters to approximate the optimal density.
Thereby estimating optimal path cost γ^* where

$$\gamma^* = \min_{\forall z} J(z) \tag{1}$$

J(z) is the cost corresponding to trajectory z.
Structure of the distribution model and its parameters are discussed in subsequent sections.

III. CROSS-ENTROPY APPROACH FOR MOTION PLANNING

Cross-entropy method estimates the expectation of a rare event among the samples of a given probability distribution. In our problem, obtaining a minimum cost path is the rare event i.e., getting path z^* having cost γ^* is the rare event. We use multi-level approach to iteratively increase the expectation of a rare event with CE estimate updating our GMM model parameters.

Consider z_1,z_2,z_3,z_4,..,z_n are the trajectories from start to goal sampled from initial guessed sampling distribution using goal-biased RRT* algorithm. Let J(z) be the cost function for a single trajectory. γ_i is used in place of γ^* for i^{th} iteration and J(z) $\leq \gamma_i$ is the rare event for the i^{th} iteration.

We arrange the J(z_1),..,J(z_n) costs in ascending order and then select the first $\delta * n$ high quality samples (these are the best paths to train GMM to approach optimal distribution as the no.of iterations increase) where δ is a user-defined parameter (we use 20^{-2}). Those samples are called elite samples

γ_i for ith iteration is then choosen as follows

$$\gamma_i = J(z_{\delta * n}) \tag{2}$$

The sampling distribution for trajectories is represented as p(.,v) where v are the parameters for the GMM model. The parameters are updated by the elite samples in the ith iteration using:

$$v = \underset{\forall v}{\operatorname{argmax}} \frac{1}{\mid Z_k^i \mid} \sum_{z \in Z_k^i} \ln p(z, v) \tag{3}$$

where Z_k are the elite samples.

IV. SAMPLING DISTRIBUTION MODEL

We are using Gaussian mixture model (GMM) as our sampling distribution model. Mixture of gaussian models enable us to estimate normally distributed sub-spaces in the state region. They do not require subspace label with training data points and can learn in an unsupervised manner.

Now, for a Guassian Mixture Model, given the parameters:

- K, the number of Guassian components
- $\alpha_1 ... \alpha_k$, the mixture weights of the components
- $\mu_1 ... \mu_k$, the mean of each component
- $\Sigma_1 ... \Sigma_k$, the variance of each component

We can generate samples $z_1, z_2 ... z_n$ from the distribution.

$$p(z, v) = \sum_{k \in K} \frac{\alpha_k}{\sqrt{2\pi \mid \Sigma_k \mid}} e^{-\frac{1}{2}(z-\mu_k)^T \Sigma_k^{-1} (z-\mu_k)} \tag{4}$$

V. TCE-RRT* AND SCE-RRT* ALGORITHM

The primitives of a trajectory is a set of parameters which represent a trajectory. The cost function J(z) is designed on the basis of the primtives. The sampling distribution samples trajectory in the finite parametric space spanned by the primitives. We always sample trajectories from a given sampling distribution using goal-biased RRT* algorithm.

When the primitives of the trajectory is composed of parameters, element of a finite parametric space, the cost depends on the parameters which encode the states of the trajectory as well as the relationship between the states especially suited for narrow space constrained environment like bug-traps. The sampling distribution samples from the parametric space. The trajectory is computed from the sampled primitive. We sample group of states from this trajectory. This version of RRT* algorithm is known as Trajectory cross entropy (TCE) RRT* algorithm.

A. ALGORITHM OVERVIEW OF TCE RRT*

The algorithm steps as follows:
STEP 1. Expand the RRT* tree and sample nodes with goal bias.
STEP 2. Obtain $z_1, .., z_n$ trajectories from start to goal.
STEP 3. Extract primitives from each trajectory.
STEP 4. Select elite subset of set of primitives.
STEP 5. Update sampling distribution with elite primitives.
STEP 6. Sample one or more states.
STEP 7. Go to STEP 1 until termination.

When the primitives are the sequence of states of the trajectory and the sampling distribution on state space, samples only next state from the state space, this version of RRT* algorithm is known as State cross entropy (SCE) RRT* algorithm. This is suited for less constrained obstacle environment as compared with TCE RRT* algorithm.

B. ALGORITHM OVERVIEW OF SCE RRT*

The algorithm steps as follows:
STEP 1. Expand the RRT* tree and sample nodes with goal bias.
STEP 2. Obtain $z_1, .., z_n$ trajectories from start to goal.
STEP 3. Discretize each trajectory into sequence of states.

STEP 4. Select elite subset of set of discretized trajectories.
STEP 5. Update sampling distribution with elite set of states.
STEP 6. Sample only one (next) state.
STEP 7. Go to STEP 1 until termination.

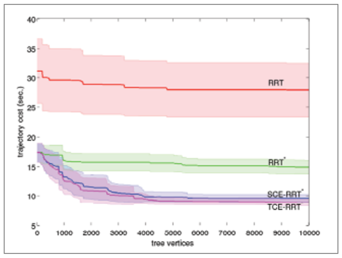

Fig. 1. Evolution of trajectory cost with increase in number of tree vertices was compared for RRT, RRT*, TCE RRT*, SCE RRT*

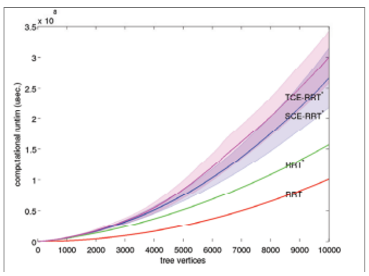

Fig. 2. Evolution of computational time with increase in number of tree vertices was compared for RRT, RRT*, TCE RRT*, SCE RRT*

In Fig 3 and Fig 4, we can observe that with increase in number of tree vertices sampled the trajectory cost for TCE RRT* and SCE RRT* is significantly getting reduced but as a trade-off the computational time is also increasing.

VI. Termination and Convergence

The termination of TCE RRT* and SCE RRT* algorithm is done when the change in the sampling distribution is less than a small no. ϵ or the GMM covariances have nearly shrunk to zero.

The formal convergence to the optimal solution is not guaranteed as the samples can be concentrated in a local sub-space and we can not reach the global best solution. We need to add noise factor in our sampling distribution for global exploration.

VII. Simulation Results

We performed simulations on a two-link manipulator model with base link pivoted at ground and free link with a joint at the other end of the base link. Objective is to find the optimal path (minimum euclidean length) from the initial point to the goal point avoiding the obstacle.

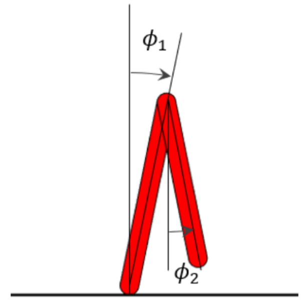

Fig. 3. Two link manipulator

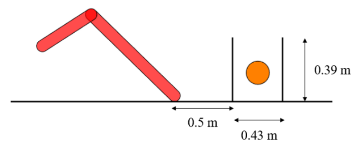

Fig. 4. bin obstacle with the ball

we will use goal-biased RRT* algorithm with CE based sampling in two-dimensional state space with ϕ_1 , ϕ_2 as the states.

TABLE I
Problem Specifications

manipulator base-link length	1 m
manipulator free-link length	0.5 m
manipulator base-link angle range ϕ_1	-0.8 - 0.8 rad
manipulator free-link angle range ϕ_2	-1 - 1 rad
manipulator initial node in state space	[-0.8,1] rad
manipulator goal node in state space	[0.8,-1] rad

We implemented SCE RRT* algorithm on the given two-link manipulator model in Matlab 2021a and got the following results:

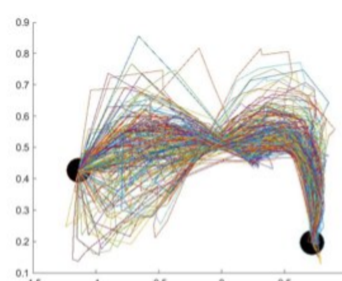

Fig. 5. Here the trajectories are shown in euclidian space with two black circles as the initial point and the goal point with respect to the end-effector of the manipulator. We ran RRT* for 500 times in each iteration (to get train data for GMM) and plotted the trajectories whenever RRT* returned a trajectory. Trajectories in this figure are sampled from GMM with the initial parameters that we gave

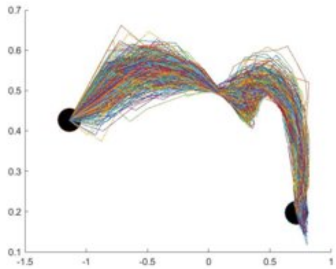

Fig. 6. Elite trajectories in the second iteration

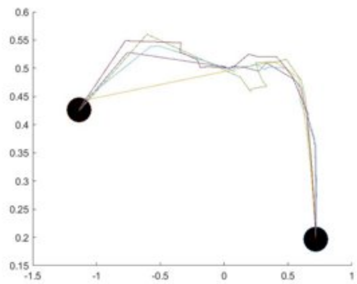

Fig. 7. Elite trajectories in the third iteration

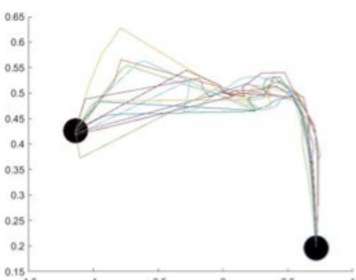

Fig. 8. Elite trajectories in the fourth iteration

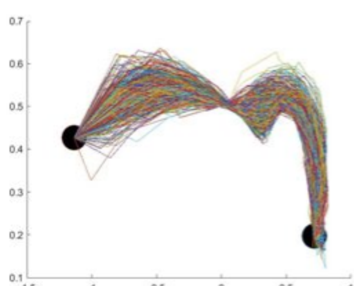

Fig. 9. Elite trajectories in the fifth iteration

TABLE II
SIMULATION SPECIFICATIONS

number of iterations for optimization	6
maximum tree nodes for RRT* tree	1000
number of trajectories sampled	500
δ **for elite samples**	0.2
initial GMM weights	[0.3,0.3,0.4]
initial GMM covariance matrices	$[I_2, I_2, I_2]$
initial GMM means	[0.3,0.1;0.4,0.3;-0.2,0]
goal tolerance	0.01
obstacle tolerance	0.1
RRT* neighbourhood radius	0.1

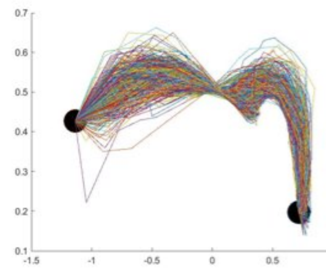

Fig. 10. Elite trajectories in the sixth iteration

OBSERVATION

In the first iteration, the elite trajectories are not optimized and covers wider region around the optimal path but with second iteration the elite region is getting concentrated along the optimal path and by the sixth iteration it is narrower than the second iteration. In the third and fourth iteration we are getting less elite samples so GMM is not able to train well and we have higher cost but by the sixth iteration, cost is decreasing as in Fig 11.
We can also observe that there are 3 regions where maximum points seems to be sampled from. One is start, other is goal and there is a region in between. This is because we used 3 component GMM model. This may vary with the no.of components of GMM. Optimal no.of components depends on problem setup and there is no single number which suits for all problems.

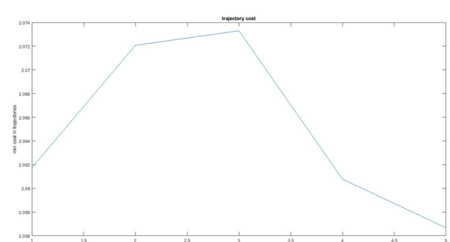

Fig. 11. Minimum cost of elite trajectories in each iteration

CONCLUSION

We obtained a sampling distribution close to the optimum sampling distribution as in the sixth iteration, the minimum cost is 2.05 which is close to the optimal cost of 1.928 calculated using euclidean norm. We can converge to a better sampling distribution with more number of iterations.

We are highly obliged that we got this opportunity to study the referred paper and implement it's results on a two-link manipulator model.

REFERENCES

[1] Kobilarov, Marin. "Cross-entropy motion planning." The International Journal of Robotics Research 31, no. 7 (2012): 855-871.

[2] Medeiros, Felipe Leonardo Lôbo, and José Demisio Simões da Silva. "A Dijkstra algorithm for fixed-wing UAV motion planning based on terrain elevation." In Brazilian Symposium on Artificial Intelligence, pp. 213-222. Springer, Berlin, Heidelberg, 2010.

[3] Geraerts, Roland. "Sampling-based motion planning: Analysis and path quality." In Utrecht University. 2006.

[4] Hussein Mohammed,Mohammad Jaradat, Lotfi Romdhane. "RRT*N: An improved rapidly-exploring random tree approach for reduced processing times".

[5] Steven M.Lavalle. "Planning Algorithms".

[6] Jiankun Wang,Charles X.-t.Li,Wenzhend Chi,Max Q.-H.Meng. "Tropistic RRT : An Efficient Planning Algorithm Via Adaptive Restricted Sampling space".

[7] Ahmed Hussain Qureshi,Yasar Ayaz. "Intelligent Bidirectional RRT for optimal motion planning".

[8] Hsu D, Sanchex-Ante G and Sun Z. " Hybrid PRM sampling with a cost-sensitive adaptive strategy.

[9] Bishop. "pattern recognition and machine learning".