A Probabilistic Decision Engine for Navigation of Autonomous Vehicles under Uncertainty

Zhenhua Jiang, *Senior Member, IEEE*
Power and Energy Division
University of Dayton Research Institute
Dayton, OH, USA
Email: zjiang4@udayton.edu

Seyed Ata Raziei, *Student Member, IEEE*
Electrical and Computer Engineering
University of Dayton
Dayton OH, USA
Email: razieis1@udayton.edu

***Abstract*- Navigation or route planning under uncertainty is a very challenging but important task for autonomous vehicles such as self-driving cars, drones, unmanned aerial systems (UAS), etc. Probabilistic methods to the modeling and optimization of these systems are attractive in quantitatively capturing the uncertainty present in their dynamic environment. This paper will present a novel probabilistic decision engine that can serve as the core for the navigation of autonomous vehicles under uncertain conditions. This probabilistic decision engine takes in a network connection matrix (based on maps and graph theory) and a cost matrix (with entries of the cost's mean values and probability distributions) as its input and generates the probability distributions of the optimal routes as its output. The proposed probabilistic decision engine consists primarily of a stochastic network standardization module, a stochastic network decomposition module and a probabilistic solver (i.e., decision kernel). A deterministic network reduction method based upon Dijkstra's algorithm is first used to derive a standard, reduced network, augmented by a stochastic network reduction process. The standard network is then decomposed into a series of stochastic subnetworks by using sequential convolution, PDF (probability distribution function) shifting and reshaping techniques. A purely-analytical probabilistic solver is finally used to solve the stochastic decision-making problem. In this paper, the principle of operation and implementation methods of the entire probabilistic decision engine will be discussed in detail. Some representative simulation results will be provided to demonstrate the effectiveness of the proposed computational methodology and compared with the traditional Monte-Carlo simulation method to validate the analytical results. This study suggests that the time needed to find the solution using the proposed decision engine can be reduced by three to four orders of magnitude, compared with the Monte-Carlo simulation method.**

Keywords—autonomous vehicles, probabilistic decision engine; uncertainty; path planning; real-time solution acceleration.

I. INTRODUCTION

In the past years, autonomous vehicles, such as self-driving cars, drones, and unmanned aerial systems (UASs), have found increasing applications in commercial and other missions [1]-[5]. An important feature of these systems is that they possess a variety of autonomy capabilities such as sensing, reasoning, and action skills; however, these systems typically operate in an uncertain or dynamically-changing environment. Even though, they are required to have a capability of robust operation for an extensive period of time with minimal or no human operator intervention. To address this, a variety of navigation and path planning algorithms have been widely investigated [6]-[10].

Navigation or route planning under uncertain conditions is very challenging but important for these autonomous vehicles. To improve the system performance, it is necessary to capture these uncertain factors and consider them in the planning and operation globally; however, a solid theoretical foundation is lacking that accounts for the uncertainty throughout the entire process. To account for the uncertainty in the autonomous vehicles and their dynamic environment, probabilistic methods to the analysis, modeling and optimization are urgently needed and beneficial [5]. However, the probabilistic optimization in these autonomous planning and operation problems is highly challenging due to the complexity of probabilistic algorithms and the long time needed to find the probabilistic solution.

Traditional methods use Monte Carlo simulation to obtain statistical results in a large number of scenarios and derive the probability distribution based on these simulation results [11]-[12]; however, these methods are very time-consuming and thus make the real-time operation very difficult or intractable. This paper presents an analytical, probabilistic decision engine that can serve as the core for the navigation of autonomous vehicles under uncertain and dynamic environment. Sensor data that are collected and processed locally and the real-time traffic flow data, as well as simultaneous localization and mapping (SLAM) algorithm, can be used together to generate location information, routing networks and cost knowledge. The probabilistic decision engine takes a network connection matrix (based on maps and graph theory) and a cost matrix (whose entries are the cost's mean values and its probability distributions) as its input and generates the probability distributions of optimal routes as its output, at any decision step, e.g., every 100 milliseconds.

The rest of the paper will be organized as follows. First, the operational principle of the entire probabilistic decision engine will be overviewed in Section II. Then, the implementation methods will be formulated for stochastic navigation, where the solution procedure and challenges in solving the problem are discussed. The main focus will be presenting the three main components in Section III. Representative simulation results will be provided in Section IV to demonstrate the effectiveness of the proposed computational solver and be compared with the traditional Monte-Carlo simulation method to validate the analytic results. This study suggests that the time needed to find the solution using the proposed decision engine can be greatly reduced, compared with the Monte-Carlo method. Section V will conclude the paper and provide information about future research directions.

This work was sponsored by the "Ohio Research Scholar" funding of the "Ohio Third Frontier" Program.

II. Proposed Probabilistic Decision Engine and Computational Framework

A. Graph-Based Computational Environment

An initial probabilistic computational framework for real-time planning and operation of autonomous vehicles under uncertainty was first proposed in [13]. This method considers three levels of planning: navigation - which decides the best possible path to take in the near future, path or motion planning - which decides the immediate path constrained by or subject to obstacles or high costs, and motion control - which aims to determine control actions for the motion actuators. Preliminary work on the three aspects has been reported in [13]-[15].

When planning for the movement of autonomous vehicles, an important requirement is to understand its own location and its environment. The graph-based modeling can be used to capture the spatial geometry of the environment using maps and 2-D or 3-D grids [8]. Typically, a variety of sensors (such as cameras, radar, LIDAR, etc.) and GPS receivers can collect large volumes of data and the map and localization information can then be derived from these data sets [2], where data fusion technologies are applied to develop the models. Additional work is under way to build the framework that fuses sensor data and transforms the extracted information to graph-based knowledge representation and will be reported in the future.

B. Overview of Proposed Probabilistic Decision Engine

The proposed probabilistic decision engine is illustrated in Fig. 1, where the blocks represent the performed actions or processes, and the circles indicate the resultant outcomes. The proposed probabilistic decision engine can serve as the core component for the navigation of autonomous vehicles under uncertain and dynamic environment. While data can be collected from various sensors onboard the vehicles and locally processed in real time, a simultaneous localization and mapping (SLAM) system can be used to generate the vehicle's location knowledge and routing networks. From real-time traffic data, it is possible to build a model for the traffic flow and congestion conditions and the associated driving costs such as traveling time or fuel costs, which can be represented by a connection/cost network based on maps and graph theory. In navigation of autonomous vehicles, the specific location in the environment is represented by a finite number of nodes and the network shows the nodes and connections among them. A nonnegative numerical value can be assigned to each edge (or branch) which denotes the cost moving from one node to another. An example of such network will be shown later in Fig. 6.

Based on available sensor data and processing algorithms, the probabilistic decision engine takes in a network connection matrix and a cost matrix (whose entries are the mean values and probability distributions of the costs associated with each edge) as its input and generates the probability distributions of the optimal routes as its output. Basically, there are three integral components in the proposed probabilistic decision engine: a stochastic network standardization module, a stochastic network decomposition module and a probabilistic computational solver, as shown in Fig. 1. Staring from the original connection and cost networks, a deterministic network reduction method based on path planning algorithms is used to derive a standard, reduced network, augmented by a stochastic network reduction process. The standardized network is then decomposed into a series of stochastic subnetworks by using the convolution, probability distribution function (PDF) shifting and PDF reshaping techniques. A purely-analytical probabilistic solver is finally used to solve the stochastic decision problem. The output will be sent to the lower layers for path planning and motion control.

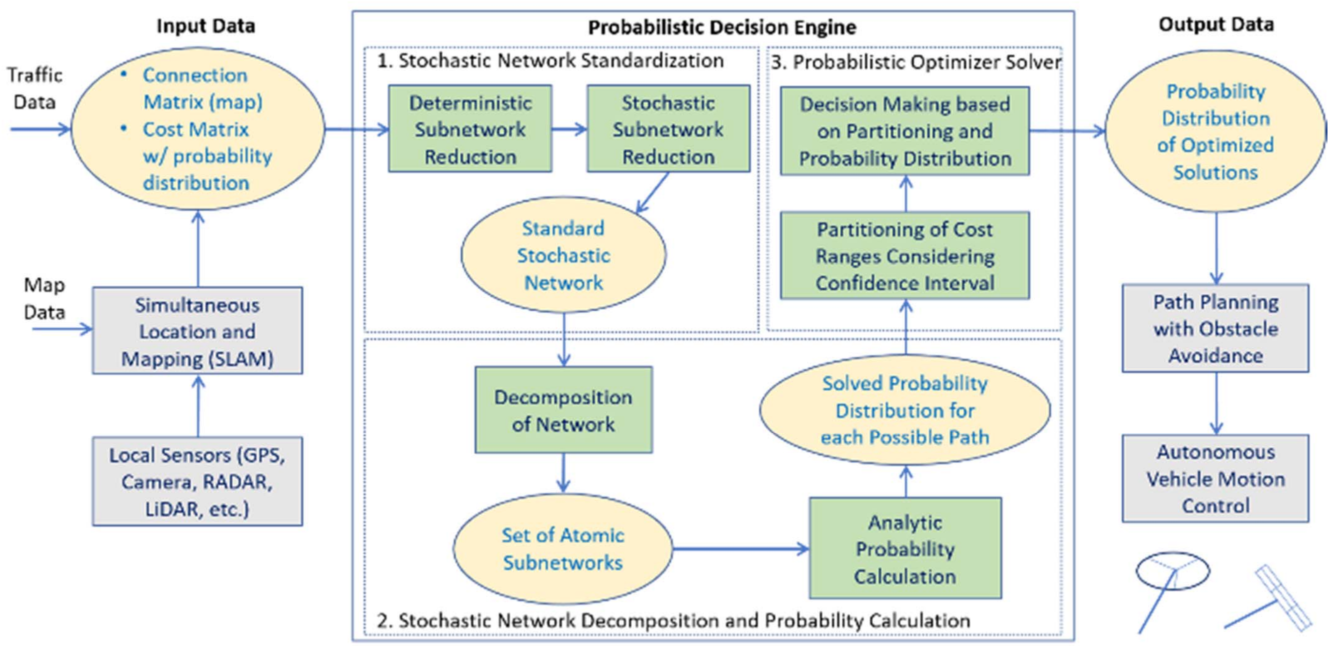


Fig. 1. Illustration of the proposed probabilistic decision engine consisting of a stochastic network standardization module, a stochastic network decomposition module and a probabilistic optimization kernel. The probabilistic decision engine takes in a network connection matrix and a cost matrix as its input and generates the probability distributions of the optimal routes as its output. The output will be sent to the lower layers for path planning and motion control of autonomous vehicles.

III. Main Components of Proposed Probabilistic Decision Engine

A. Stochastic Network Standardization

A standard network topology and cost matrix will be helpful in formulating the optimization problem as well as algorithm development. To achieve this, a standardization process will be first performed for any given network under consideration to reduce the complexity. As a stochastic cost network (or graph or matrix) contains some deterministic portions, it is desired to reduce both the deterministic and stochastic portions to smaller subnetworks, respectively, and combine them together to obtain a reduced cost network. In case of large networks, the portions far away from the location under current consideration may usually be assumed deterministic, since the impact of the uncertainty of those portions on the immediate solution is not significant and thus negligible. In the proposed framework, a format of the standard, reduced network is defined, as shown in Fig. 2. The number of stochastic routes determines the order of the network, and thus the complexity of the probabilistic optimization algorithm.

In this standardization process, the deterministic portion of the original network can first be reduced to a simplified equivalent subnetwork by using rules and deterministic optimization. The objective of this level of optimization is to find the deterministic path with the minimum cost that connects the starting node to the destination node in the graph. Assuming that all edges have nonnegative costs, Dijkstra's algorithm can be used to determine the optimal path in the network with both unidirectional or bidirectional edges [17]-[19]. A brief description of this algorithm was presented in [13]. The stochastic portion can be reduced by performing the convolution or other probability calculations. Combining the reduced deterministic subnetwork with the reduced stochastic portion, it is possible to derive the resultant reduced network as shown in Fig. 2. As there is no possible path between any two nodes, the respective edge is removed from the network. The optimization problem is thus reduced to a much smaller problem with fewer random variables, greatly reducing the solution time. This standardization process renders the remaining calculation modular and scalable, which makes the approach generally applicable to different situations.

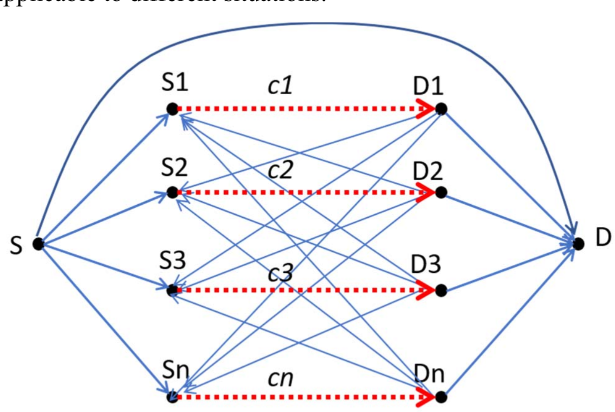


Fig. 2. illustrates the standardized network with starting point of S and destination of D. The number of stochastic routes determine the order of the entire network, and the complexity of the probabilistic optimzaition algorithm.

B. Stochastic Network Decomposition and Probability Calculation

Based upon Bayesian and Dempster-Shafer theories [16], the standard, reduced network will be decomposed into multiple atomic subnetworks. Fig. 3 briefly illustrates an example of the decomposition process, where one scenario is identified. This scenario has a possible path from S to D, which contains 3 deterministic edges (S-S1, D1-S3, and D3-D) and 2 stochastic edges (S1-D1, and S3-D3), and a further stochastic network reduction step will be performed to reduce the subnetwork to a possible stochastic path with cost probability distribution.

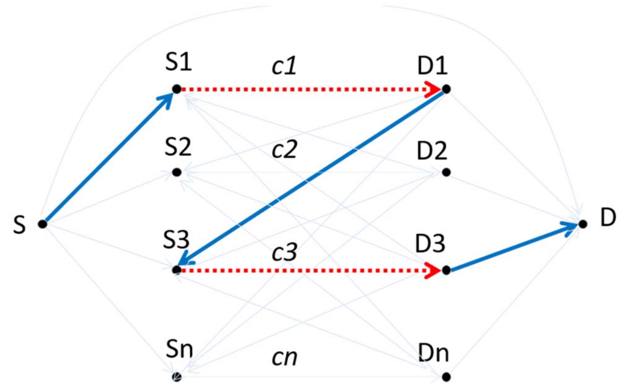


Fig. 3. Illustration of the decomposition process. One possible path from S to D is highlighted, which contains 3 deterministic edges (S-S1, D1-S3, and D3-D) and 2 stochastic edges (S1-D1, and S3-D3).

For each atomic subnetwork, the probability distribution of the total cost can be determined by using sequential convolution, probability density function (PDF) shifting and PDF reshaping techniques, since there are deterministic and stochastic edges in each scenario. Following the same procedure, all possible stochastic paths can be derived based on the connection graph. Fig. 4 illustrates an example of the resultant possible paths, where each scenario is independent and exclusive from others and the combination should comprise all scenarios of the routing options. In the decomposed network as shown in Fig. 4, the cost of the deterministic path is fixed, while the costs for stochastic paths are subject to combined probability distributions which are calculated from the convolution and pdf shifting/reshaping techniques. The details may be reported in a future publication.

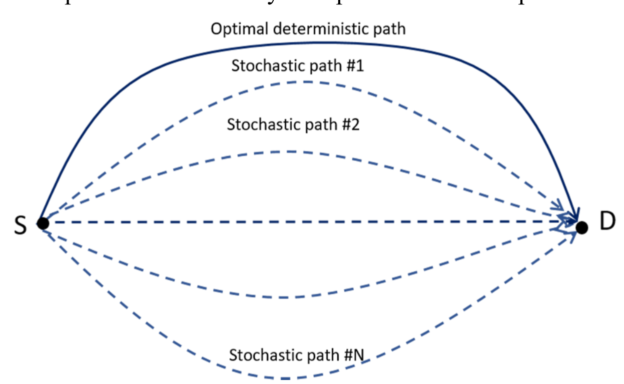


Fig. 4. Example scenarios of decomposed atomic subnetworks. There is only one deterministic optimal path from S to D and multiple (N) stochastic optimal paths, each of which may involve multple determinstaic edges and one or more (up to n) stochastic edges in the standard network.

C. Probabilistic Optimization Solver

This section focuses on the specific analytical methodology to find the probabilistic optimization solution. When running Monte-Carlo simulation, an important step is to generate a large number of samples for random variables which will be used to conduct experimental runs individually. However, the total time for these simulations would be reduced if some samples could be removed from consideration in the apparent “out of range” scenarios. As an example, adaptive bounding can be applied to the sampling process [13]. In the analytic solution, a probability discretization method is used instead of sampling, and the same adaptive bounding procedure can be used to make the computing faster. This process will be expanded into a future paper.

Fig. 5 shows a general scenario for two random variables and a deterministic variable in parallel in considering a probability distribution of the optimal solution using curve bounding. When the value of a random variable is higher than the deterministic variable, this portion may be discarded due to apparently high costs, for instance, Region R_{U1} or R_{U2} in Fig. 5. When the value of a random variable is lower than the deterministic variable as well as the minimum value within a certain confidence interval (e.g., 99%), this portion may not be discretized for comparison but will be directly considered for the later probability calculation, for example, Region R_{L1} in Fig. 5. In the middle region (i.e., R_M), a detailed derivation will be done.

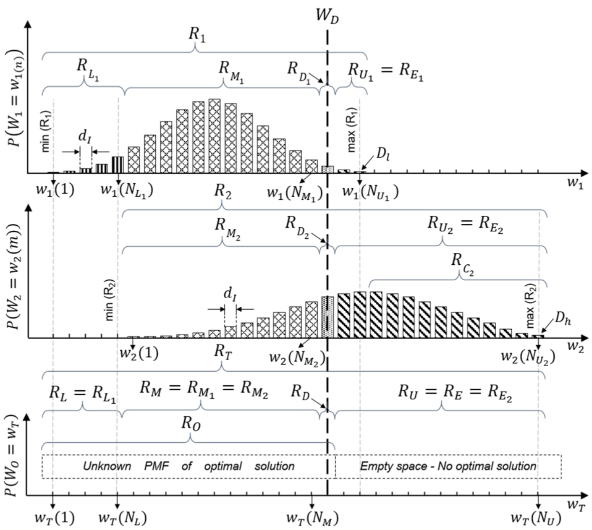


Fig. 5. General scenario of two random variables and a deterministic variable for consideration of pdf discretization and bounding.

IV. Case Study and Results

This section will use a specific example to demonstrate the application of the developed probabilistic decision engine in inferring the optimal traveling paths under uncertain conditions.

A. Problem Dscription

Fig. 6 shows the network or map under study, where the vehicle will move from Point 7 to Point 9. The solid lines represent the deterministic edges with constant costs and dashed lines indicate the edges with stochastic costs. This is a snapshot of the driving condition. The connection and cost matrices will be updated at any decision step, e.g., every 100 milliseconds.

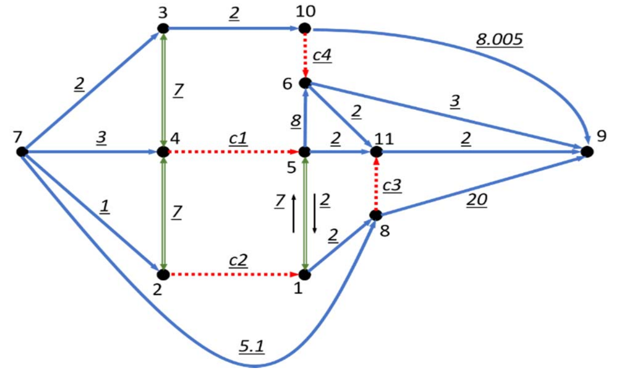
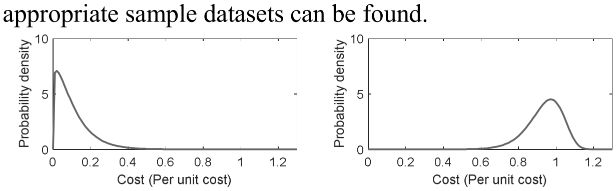
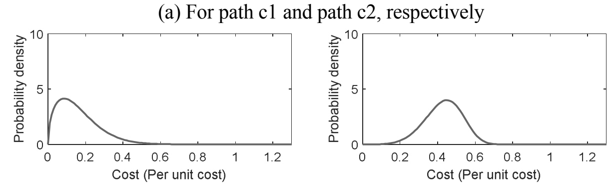


Fig. 6. Illustration of the network under study, where the vehicle will move from Node 7 to Node 9. The solid lines indicate the deterministic edges with constant costs and dashed lines represent the stochastic costs.

As an example, the Weibull distribution is considered here for modeling the travel costs or weights (e.g., travel time or fuel cost) of the stochastic edges [20], since the costs are positive and may vary over a large range. Fig. 7 shows the probability density curves of the costs for the stochastic edges in this example network. It is worthwhile to notice that many different types of probability distributions can be directly handled by the proposed decision engine as long as the analytic expressions of PDF or appropriate sample datasets can be found.



(a) For path c1 and path c2, respectively



(b) Path c3 and path c4 respectively

Fig. 7. Probability density curves of the costs for the stochastic edges in the example network.

B. Results for Network Standardization

Following the strategy in Section III-A, the network can be reduced to a much simpler network, as shown in Fig. 8.

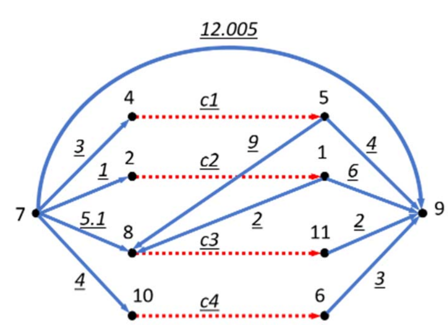


Fig. 8. Reduced network in the standardized form, which has one possible deterministic path from S to D, and 4 stochstic edges in the middle.

The order/rank of the standard stochastic network is 4, i.e., the number of independent (combined) stochastic edges. This number does not change, while some deterministic edges have been merged. In the network standardization process, the node indexes of the original network can be tracked, as shown in Table 1. Ten nodes (vertices) are preserved, and the reduced network is smaller than the original network in terms of possible paths from the start point to the destination. It is valuable to note that the first deterministic path (i.e., Path 1), which contains multiple deterministic edges in the original network, has been found by Dijkstra's algorithm before the standardization.

TABLE 1 – NODE INDEX TRACKING ON THE STANDARDIZED NETWORK

Path Number	Nodes in Original Graph	Nodes in Standard Graph
1	[7,3,10,9]	[7,9]
2	[7,4,5,11,9]	[7,4,5,9]
3	[7,2,1,5,11,9]	[7,2,1,9]
4	[7,8,11,9]	[7,8,11,9]
5	[7,3,10,6,9]	[7,10,6,9]
6	[7,4,5,1,8,11,9]	[7,4,5,8,11,9]
7	[7,2,1,8,11,9]	[7,2,1,8,11,9]

C. *Results for Network Decomposition*

Following the decomposition process, the standard network can then be decomposed, and the atomic subnetworks are very simple and easy to solve. Each route is an atomic network and may contain multiple branches in the standard network and many edges in the original network. The 7 possible paths, and their cost distributions are shown in Fig. 9, where the high bars on the right side considers the PDF tails. It is worthwhile noting that the cost's probability distributions for the 6 stochastic edges are calculated from the original PDF functions of the 4 stochastic edges in the original network. Interestingly, the last two options can be excluded from the further consideration since the cost is always higher than that of the deterministic path.

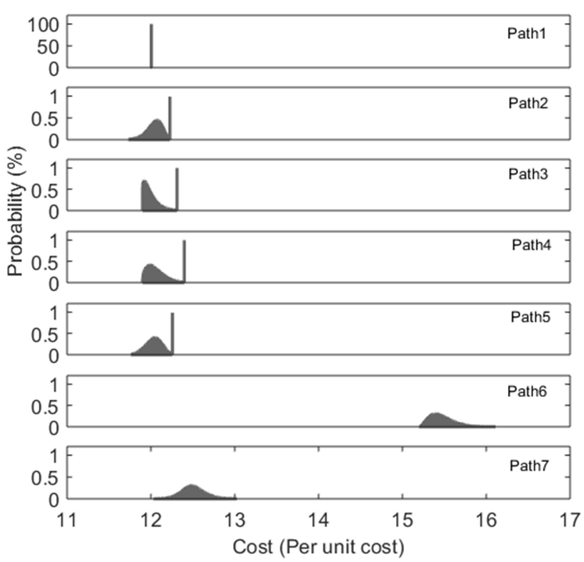


Fig. 9. Illustration of decomposed atomic subnetworks, i.e., 7 possible paths, and their cost distributions, where the high bars on the right side considers the PDF tails.

D. *Results for Probability Distribution of Optimized Solutions*

Fig. 10 shows the result about the probability distribution of potential optimal paths using the analytical approach described in Section III-C. The simulation is run and this result is obtained on a multi-core computing system (with Intel Xeon CPU E5-4657L, 2.40GHz, 12-Core Processor and 256GB memory). It takes less than 100 milliseconds to complete the solution. In the figure, we can see that path 3 (i.e., the 2nd stochastic path, 7-2-1-5-11-9) ranks as the top choice with a probability of 36.4%. The probability that the deterministic path (path 1, i.e., 7-3-10-9, in the original network) is optimal among all is around 10.42%. It is worth noting that the probabilistic distribution of the potential optimal paths are dependent on the original stochastic costs of the individual edges in the original network.

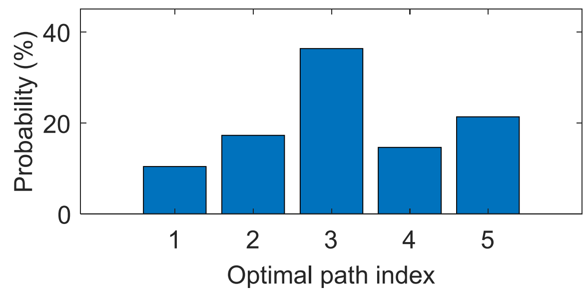


Fig. 10. Probability distribution of possible optimal paths.

Fig. 11 illustrates the cost distribution of the optimal paths within the cost range between approximately 11.73 and 12.005 per unit cost. The areas between the curves (i.e., integration of respective PDFs) are the respective probability values of the optimal paths. Since the cost of the deterministic path [7 9] is 12.005 per unit, it would become a better option when the stochastic cost in other parallel paths could be higher than this value. This explains why there exists no cost higher than that value, and therefore the bar for the 12.005 per unit cost interval looks very high, i.e., 10.42%. This mechanism will provide a feature that the stochastic optimal solution will be bounded by the deterministic optimal solution.

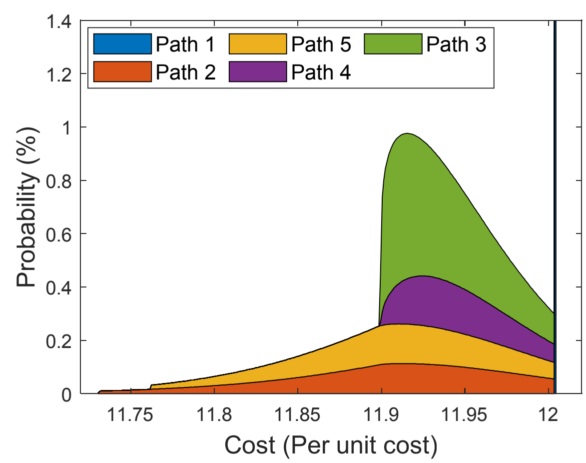


Fig. 11. Cost distribution of the optimal paths, where path 1 is the deterministic optimal path with a probability of 10.42%. The areas between the curves (i.e., integration of respective PDFs) are the respective probability values of the optimal paths.

In this study, deterministic optimization is also performed by considering the mean values of the probabilistic costs for the stochastic edges. It is found that the optimal solution with the highest probability in stochastic optimization is the same as what is obtained from deterministic optimization considering the expected mean values. Compared with the pure-deterministic approach which only uses constant costs or expected values of the stochastic costs, the probabilistic optimization solution provides more information such as the probability distribution of multiple possible optimal paths instead of a single path. As the uncertainty changes (e.g., the stochastic costs have different PDFs), this probability distribution output will also change. This will provide an adaptive snapshot of the dynamic environment, which is especially useful for the autonomous driving where decision making could be based on this information instead of only human driver's experience and instruction.

E. Validation and Timing Analysis

Fig. 12 presents the comparison of analytical results obtained above with the Monte-Carlo simulation results, where Case 2 is Monte-Carlo simulation with central sampling and Case 3 is with random sampling in the discrete cost slices. The slice width is 0.01 (per unit cost) and 300 random samples are taken from each stochastic edge. In this process, slices of equal width are taken to divide/discretize the cost into multiple units or intervals for computing the probability analytically. Table 2 compares the probability values of potential optimal paths calculated using the analytical and simulation methods, respectively.

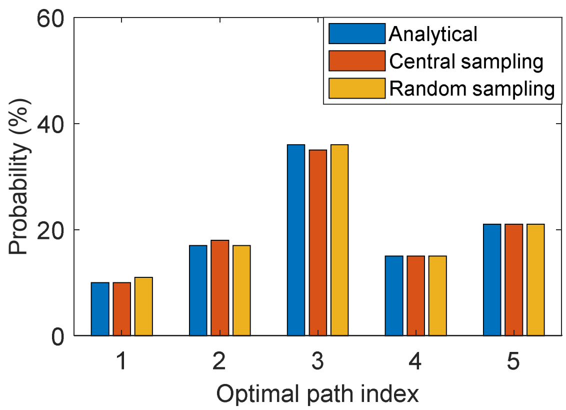


Fig. 12. Comparison of analytical results with Monte-Carlo simulation results, where Case 2 is Monte-Carlo simulation with central sampling and Case 3 is Monte-Carlo simulation with random sampling in the intervals.

TABLE 2 – PROBABILITY VALUES OF POSSIBLE OPTIMAL PATHS

Path index	Path				Occurrence chance		
					Analytical	Monte-Carlo 1	Monte-Carlo 2
1	7	9	0	0	10.42%	10.32%	10.52%
2	7	4	5	9	17.25%	18.21%	17.14%
3	7	2	1	9	36.36%	35.43%	36.44%
4	7	8	11	9	14.61%	14.75%	14.62%
5	7	10	6	9	21.36%	21.30%	21.28%

It can be seen from Fig. 14 and Table 2 that the Monte-Carlo simulation results match the analytical calculations very well, which also suggests that the network decomposition concept is valid for stochastic optimization. In this case, the run-time for Monte Carlo simulation is higher than 100 seconds, while it takes less than 100 milliseconds to complete the solution, as mentioned before. This study suggests that the time needed to find the solution using the proposed analytical decision engine can be reduced by three to four orders of magnitude, compared with the Monte-Carlo method.

F. Sensitivity Analysis

The sensitivity analysis was performed to understand how the changes in the parameters, such as the number of samples for Monte Carlo simulation, computational confidence interval and discrete slice width for analytic probability calculation, may affect the solution accuracy.

1) Impact of Number of Samples

Fig. 13 shows the probability distribution of the possible best paths. In addition to the baseline (Case 1), five additional cases are considered for the Monte Carlo simulation with different number of samples. Table 3 also lists the sensitivity analysis showing the relative error between the analytical method (baseline) and Monte Carlo simulation. It shows that the relative error becomes the lowest with the highest number of samples in the Monte Carlo simulation.

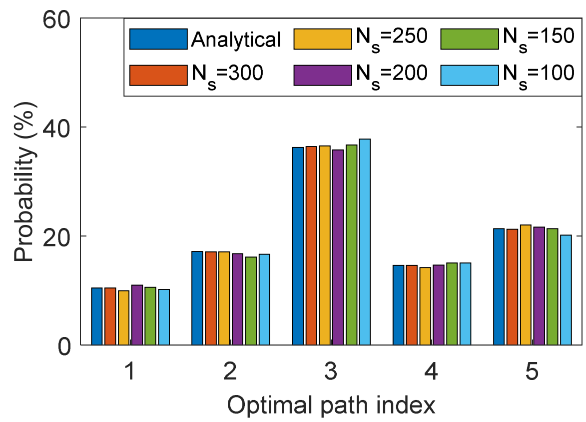


Fig. 13. Comparison of probability distribution of possible optimal paths under different numbers of samples for Monte-Carlo simulation.

TABLE 3 – RELATIVE ERROR BETWEEN ANALYTICL METHOD (BASELINE) AND MONTE CARLO SIMULATION WITH DIFFERENT NUMBERS OF SAMPLES

Number of samples	Path index					Sum
	1	2	3	4	5	
300	0.010	0.006	0.002	0.001	0.004	0.023
250	0.043	0.005	0.006	0.025	0.033	0.111
200	0.055	0.027	0.014	0.006	0.016	0.118
150	0.019	0.062	0.011	0.032	0.001	0.125
100	0.020	0.034	0.040	0.034	0.054	0.182

2) Impact of Confidence Interval

Fig. 14 shows the comparison of probability distribution of the possible optimal paths under different confidence intervals. The analysis is achieved through analytical calculations, and in the figure, the confidence interval of probability mass function of the optimal path weight is equal to ci^2, where ci=99%, 95% and 90%. This is because there are two stochastic edges in parallel with each other in some atomic subnetworks. While there are slight changes in the probability distribution, Table 4 shows that the analytical results with the highest (i.e., 99%) confidence interval match the Monte Carlo simulation best.

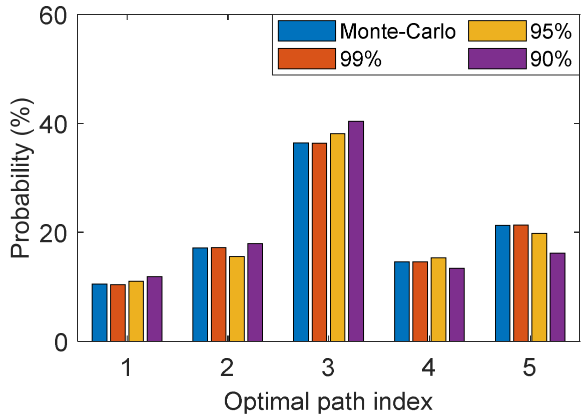


Fig. 14. Comparison of probability distribution of possible optimal paths under different confidence intervals.

TABLE 4 – RELATIVE ERROR COMPARISON FOR DIFFERENT CONFIDENCE INTERVALS

Confidence interval	Path index					Sum
	1	2	3	4	5	
99%	0.010	0.006	0.002	0.001	0.004	0.023
95%	0.052	0.090	0.047	0.049	0.069	0.306
90%	0.131	0.047	0.109	0.081	0.237	0.605

3) Impact of Distribution Slice Width

Fig. 15 compares the probability distribution of the possible optimal paths under different slice widths (i.e., numbers of slices) within the 99% confidence interval. The confidence interval of probability mass function of the final optimal path's weight is 0.99^2=0.9801 for all 5 cases. It seems that there are slight changes in the probability distribution; however, the relative error between the Monte Carlo simulation (baseline) and the analytical methods decreases as the number of slices increases (or the slice width decreases), as shown in Table 5.

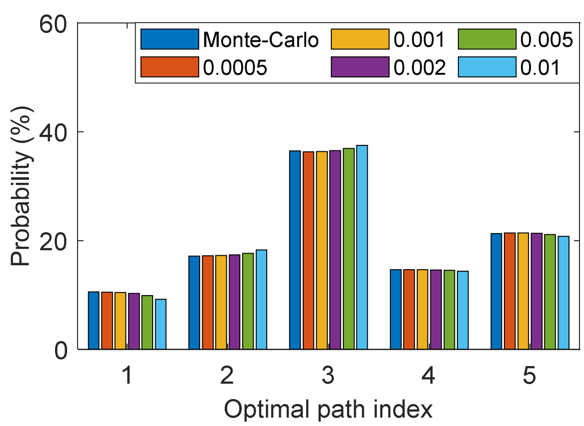


Fig. 15. Comparison of probability distribution of possible optimal paths under different slice widths (or numbers of slices) with 99% confidence interval for each stochastic variable.

TABLE 5 – RELATIVE ERROR COMPARISON FOR DIFFERENT SLICE WIDTHS

Interval width	Path index					Sum
	1	2	3	4	5	
0.0005	0.003	0.003	0.004	0.000	0.005	0.016
0.001	0.010	0.006	0.002	0.001	0.004	0.023
0.002	0.024	0.014	0.001	0.003	0.001	0.042
0.005	0.064	0.028	0.012	0.009	0.009	0.122
0.01	0.129	0.066	0.028	0.020	0.024	0.267

V. CONCLUSIONS AND FUTURE WORK

This paper has presented a probabilistic decision engine that can serve as the core for the navigation of autonomous vehicles under uncertain conditions. The probabilistic decision engine takes in a network connection matrix (based on maps and graph theory) and a cost matrix (with entries of the cost's mean values and probability distributions) as its input and then generates the probability distribution of optimal routes as its output. Basically, the proposed probabilistic decision engine consists of three main components including a stochastic network standardization module, a stochastic network decomposition module and a probabilistic computational solver (i.e., optimization kernel). In the presented probabilistic decision engine, a deterministic network reduction method based upon Dijkstra's algorithm is first used to derive a standard, reduced network, augmented by stochastic network reduction. The standard network is then decomposed into a series of stochastic subnetworks by using the sequential convolution and PDF shifting and PDF reshaping techniques. An analytical probabilistic solver is finally used to solve the stochastic decision-making problem.

In this paper, the operational principle and implementation methods of the entire probabilistic decision engine have been discussed in detail. These component algorithms are then used in an example navigation problem considering stochastic costs in some paths. Representative results have been provided to demonstrate the effectiveness of the proposed computational solver and been compared with the traditional Monte-Carlo simulation method to validate the analytic results. The optimal solution with the highest probability in stochastic optimization is found to be the same as what was obtained from deterministic optimization considering expected mean values, but stochastic optimization provides more information such as the probability distribution of multiple possible optimal solutions instead of a single solution. Timing and accuracy issues are discussed. This study suggests that the time needed to find the solution using the proposed decision engine can be reduced by three to four orders of magnitude, compared with the Monte-Carlo simulation method. The impact of number of samples, confidence interval and analytical slice width on the stochastic optimization solution is also studied.

The future work may include implementing the proposed probabilistic decision engine algorithm on real-time hardware such as FPGA, integrating the navigation algorithm with other path planning and motion control algorithms, and performing hardware-in-the-loop testing where a large number of stochastic scenarios can be easily generated in the simulation. A hardware prototype of the autonomous driving planer (or the "brain" of the vehicle) will also be developed and tested in the lab.

ACKNOWLEDGMENT

The funding support from the "Ohio Research Scholar" Program of "Ohio Third Frontier" Program is acknowledged.

REFERENCES

[1] Naval Research Advisory Committee, How Autonomy Can Transform Naval Operations, Report, Oct. 2012.

Available at https://www.nrac.navy.mil/docs/NRAC_Final_Report-Autonomy_NOV2012.pdf.

[2] R. Luo, M. Lin, R. Scherp, "Dynamic multi-sensor data fusion system for intelligent robots", IEEE Journal of Robotics and Automation, vol. 4 no. 4 pp. 386-396 1988.

[3] J. Velagic, B. Lacevic and N. Osmic, Nonlinear Motion Control of Mobile Robot Dynamic Model, Motion Planning, Xing-Jian Jing (Ed.), ISBN: 978-953-7619-01-5, InTech, 2008.

[4] N. Roy, W. Burgard, D. Fox and S. Thrun, "Coastal navigation-mobile robot navigation with uncertainty in dynamic environment", 1999 IEEE International Conference on Robotics and Automation, pp. 35-40, Detroit, MI, 1999.

[5] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, The MIT Press, Intelligent Robotics and Autonomous Agents series edition, Aug. 2005.

[6] J. Yu and S. M. LaValle, "Optimal Multirobot Path Planning on Graphs: Complete Algorithms and Effective Heuristics," in *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1163-1177, Oct. 2016.

[7] B. Bakker, Z. Zivkovic and B. Krose, "Hierarchical dynamic programming for robot path planning," 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2756-2761, 2005.

[8] Soonkyum Kim and M. Likhachev, "Path planning for a tethered robot using Multi-Heuristic A* with topology-based heuristics", 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015, pp. 4656-4663.

[9] Jean-Claude Latombe, Robot Motion Planning, Vol. 124. Springer Science & Business Media, 2012.

[10] P. J. From, J. T. Gravdahl, T. Lillehagen, and P. Abbeel, "Motion planning and control of robotic manipulators on seaborne platforms", Control Engineering Practice, pp. 809-819, Vol. 19, No. 8, 2011.

[11] A. Kaboli, M. Bowling and P. Musilek, "Bayesian Calibration for Monte Carlo Localization", AAAI, 2006.

[12] R. Y. Rubinstein, and D. P. Kroese. Simulation and the Monte Carlo Method. Vol. 10. John Wiley & Sons, 2016.

[13] Z. Jiang and A. Raziei, "Accelerating Probabilistic Optimization Solution to Autonomous Vehicles under Uncertain and Dynamic Environments", IEEE National Aerospace and Electronics Conference, July 2018.

[14] A. Raziei and Z. Jiang, "Dynamic Motion Planning of Autonomous Vehicles under Changing Conditions", IEEE National Aerospace and Electronics Conference, July 2019.

[15] A. Raziei and Z. Jiang, "Nonlinear Model Predictive Motion Control of Differential Wheeled Robots", IEEE National Aerospace and Electronics Conference, July 2018.

[16] R. R. Murphy, "Dempster-Shafer theory for sensor fusion in autonomous mobile robots," in *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 197-206, Apr 1998.

[17] S. M. LaValle, Planning Algorithms, Cambridge University Press, 2006.

[18] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, Introduction to Autonomous Mobile Robots, MIT Press, Second Edition, 2011.

[19] M. Sniedovich, "Dijkstra's Algorithm Revisited: the Dynamic Programming Connexion", Control and Cybernetics, Vol. 35, No. 3 pp. 599-620, 2016.

[20] https://www.weibull.com/.