2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Navigating Congested Environments with Risk Level Sets

Alyssa Pierson[1], Wilko Schwarting[1], Sertac Karaman[2], and Daniela Rus[1]

Abstract— **In this paper, we address the problem of navigating in a cluttered environment by introducing a congestion cost that maps the density and motion of objects to an occupancy risk. We propose that an agent can choose a "risk level set" from this cost function and construct a planning space from this set. In choosing different levels of risk, the agent adjusts its interactions with the other agents. From the assumption that agents are self-preserving, we show that any agent planning within their risk level set will avoid collisions with other agents. We then present an application of planning with risk level sets in the framework of an autonomous vehicle driving along a highway. Using the risk level sets, the agent can determine safe zones when planning a sequence of lane changes. Through simulations in Matlab, we demonstrate how the choice of risk threshold manifests as aggressive or conservative behavior.**

I. Introduction

As autonomous vehicles become more prevalent in today's society, they must learn to operate in congested and cluttered environments. To successfully navigate a dynamic environment, it is important to first quantify the level of congestion in the environment. Once the environmental congestion is known, the agent must choose a control law such that they avoid collisions with the other obstacles and agents in the environment. This paper proposes using "risk level sets" to navigate a cluttered environment. We first introduce a cost function that quantifies the level of congestion from both the location of other obstacles and agents, as well as their movement through the environment. This cost function allows us to measure occupancy for points in the environment and map it to a measure of risk for an agent attempting to navigate to that point. We allow the agents to choose a risk threshold from calculated level sets of the cost function. By construction, a conservative agent chooses a lower risk threshold, while an aggressive agent chooses a higher risk threshold. Agents are allowed to make any control action within this set, and in doing so, they are guaranteed to avoid collisions with other agents and obstacles. Our approach is useful to a number of applications in multi-agent systems. Using risk level sets reduces the control space for an agent, which guarantees collision avoidance. This is particularly useful in applications that require exploring a cluttered environment, or navigating in crowds or traffic. In situations where the number of obstacles may change, our approach scales with the density of the environment, both providing a metric of congestion to the agent and adjusting the control space.

A main focus of this paper is how these risk level sets may be applied to autonomous vehicles. For self-driving cars to fully integrate into our daily lives, they must interact with other human drivers and respond to everyday traffic scenarios. In the case of highway navigation, an autonomous vehicle will need to adapt to varying levels of traffic and congestion along the road. We use the problem of an autonomous vehicle planning a sequence of lane changes along a highway to demonstrate our planning algorithm. Using our congestion cost, the vehicle quickly assesses the level of congestion on the road. To plan the sequence of lane changes, we map the cost to a weighted graph along a planning horizon, and use Djikstra's Algorithm to find the fastest route through traffic while avoiding collisions with other cars. Using our risk level set formulation, we also demonstrate how an agent's choice of risk threshold manifests as varying behavior; conservative drivers with lower risk thresholds will make fewer lane changes, while aggressive drivers with a higher risk threshold attempt more lane changes.

A. Related Work

We focus on two core concepts: planning with guaranteed collision avoidance, then applying this to planning sequences of lane changes. Within multi-agent systems, there is extensive work on collision avoidance [1], but most assume simple approximations of the vehicle dynamics. In [2], the authors compare the reachable sets of vehicles and avoid collisions by choosing actions that avoid other vehicles while satisfying an objective function. While this works well for a small number of vehicles with reliable communication, our problem requires planning around large numbers of obstacles without communication to the human drivers. Another approach is to use a "global dynamic window" for high speed navigation [3], [4], which checks admissible velocities for motion planning against obstacles in view, however, this work is limited to static obstacles. In [5], the authors discretize a non-convex environment into polygons for navigating around static obstacles.

Our paper also draws inspiration from recent work in barrier functions for safety. In [6], the authors unify control barrier functions and Lyapunov functions in order to provide actuator bounds that guarantee safe performance. Further extensions look at other dynamics, as well as encoding collision avoidance and control guarantees for distributed systems with a single function [7], [8]. Here, we define a cost function that allows us to choose various risk thresholds for the agents.

In Section IV, we discuss how the risk level sets can be used for autonomous vehicles, and present an example of navigating in traffic along a highway. We build upon the

[1]Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA [apierson, wilkos, rus]@mit.edu

[2]Laboratory of Information and Decision Systems (LIDS), MIT, Cambridge, MA, USA sertac@mit.edu

This work supported in part by NSF Grant IIS-1724058, the Office of Naval Research (ONR) Grant N00014-12-1-1000, and by Toyota Research Institute (TRI). This article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity. Their support is gratefully acknowledged.

author's previous work, wherein lane changes among a group of vehicles are planned recursively [9]. In designing lane change algorithms, there are a multitude of approaches for autonomous vehicles. One method is to model the behavior by traffic flow, setting rules that correspond to flux [10]. Similarly, in [11], each lane has a pre-defined maximum speed, and cars "hop" between lanes to lower the probability of getting caught behind a slower vehicle. Before making any change, a maneuver must be deemed safe. One method is to determine a safe gap for merging [12], [13], however, these gaps may change as the density of traffic and driver personalities change [14]. Other methods attempt to mimic human behavior [15], or use reinforcement learning on lane switching [16]. To address risk in maneuvers, one approach is to use Scenario Model Predictive Control to account for uncertainty [17], while others explore a series of trajectories and choose the least-risky approach [18]. In this paper, we account for risk by defining a threshold to avoid collisions, and then allowing the agents to choose a level set of that risk, which scales with the density of the environment.

The remainder of the paper is organized as follows: in Section II, we present our problem formulation and assumptions. Section III defines the risk level sets as a series of thresholds, as well as a proof of collision avoidance given these risk thresholds. We apply these concepts to the problem of highway navigation for autonomous vehicles in Section IV, and Section V details our simulation results of the highway naviagtion scenarios. We present our conclusions in Section VI.

II. Problem Formulation

In this section, we present our formulation of occupancy cost, then state our assumption of self-preserving agents. Our approach treats all objects and agents in the environment as dynamic obstacles. Consider an environment $Q \subset \mathbb{R}^N$, with points in Q denoted q. For n agents in the environment, we denote the position of each agent as x_i, for $i \in \{1, ..., n\}$. We define the ego agent as the planning agent, with position denoted x_e. We assume agents have double-integrator dynamics, such that

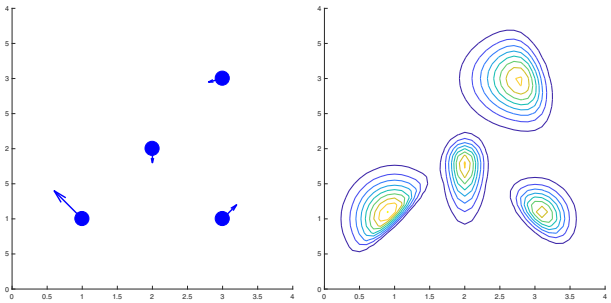
$$\ddot{x}_i = u_i,$$

where u_i is the control input for each agent. We also assume that all agents have maximum speed and acceleration, such that $\|\ddot{x}_i\| < a_i$ and $\|\dot{x}_i\| < v_i$, where a_i and v_i are the maximum acceleration and velocity, respectively.

We introduce an occupancy risk cost, H, which captures both the density of agents at a location, as well as their intended direction of motion. The cost is calculated based on all agents in the environment, defined

$$H(q, x, \dot{x}) = \sum_{i=1}^{n} \frac{\exp\left(-(q - x_i)^T \Omega (q - x_i)\right)}{1 + \exp\left(-\alpha \dot{x}_i^T (q - x_i)\right)}, \quad (1)$$

where Ω is the diagonal matrix of the inverse square of the standard deviation. In $\mathbb{R}^2$, $\Omega = \text{diag}\{\frac{1}{\sigma_x^2}, \frac{1}{\sigma_y^2}\}$. Note that the construction of H is a Gaussian peak multiplied by a logistic function. The Gaussian peak represents an estimate of an agent's position, while the logistic function skews this estimate in the direction of the velocity. By skewing the position by the velocity, the occupancy cost H is higher along points in the direct path of the agent's motion, and lower otherwise. Figure 1 illustrates the mapping from object positions and velocities to contours of H.



(a) (b)

Fig. 1: (a) Agents in environment with velocity direction indicated by blue lines. (b) Contour of H, noting the skew in the direction of the velocity.

We use the cost function as a proxy for the intention of the other agents in the system. In highly-cluttered and congested environments, it is impractical for the ego agent to know the control laws of all other agents in the system. Instead, we propose an algorithm where the ego agent only needs to calculate H and from this, can generate a conservative control plan that avoids collision with other agents. To calculate H, we assume the ego agent is able to observe the state (position) of the other agents, and can also estimate the velocity and acceleration of the other vehicles from prior observations. Since H is defined in the local reference frame of the ego agent, these measurements only need to be relative. We assume the other agents in the system are self-preserving, formalized in Assumption 1.

Assumption 1. *All agents in the environment are "self-preserving," defined as avoiding collisions with other agents when possible. To avoid collisions, the agents require a minimum stopping distance, as determined by their dynamics. If two agents' planned paths intersect, the agents will stop before colliding, given sufficient stopping distance.*

For our assumption of "self-preserving" agents, we only assume all agents have sufficient stopping distance at any given time to avoid collisions with other obstacles and agents. We do not assume an advanced collision-avoidance planner of any agent, and instead incorporate the stopping distance into determining our thresholds for the congestion cost. For each agent, from its maximum velocity and acceleration, we can find the stopping radius r_i needed for each agent to avoid collisions. While this assumption is valid in many scenarios, it does not allow for malicious agents in the system who may intentionally try to collide with other agents. Future work will explore this and other relaxations of the self-preserving assumption. The following section details how we determine risk thresholds for the agents under the self-preserving assumption.

III. Determining Risk Threshold

Under the assumption of self-preserving agents, we find the appropriate risk thresholds for the agents to guarantee

collision avoidance. We propose that there is some cost threshold that, if an agent stays below that threshold, is guaranteed to avoid collisions. The following section defines the various risk thresholds for both collision avoidance and planning costs. Let H_c define the cost at which a collision occurs. For self-preserving agents, the goal is for an agent to never exceed this cost. Given the dynamics of the system, we define $H_T < H_c$ as a maximum risk threshold, wherein if the agent stays within this cost, they will always have sufficient stopping distance to avoid collisions. Finally, we define $H_P \leq H_T$ as the planning threshold for an agent. If an agent chooses the maximum allowable risk, then $H_P = H_T$, but for any agent choosing a lower risk, $H_P < H_T$.

We construct a risk level set within the environment by choosing all values of $H < H_P$, defined

$$L_{\bar{p}} = \{q \mid H(q - x_e, \dot{x}_i) \leq H_P\}. \tag{2}$$

We will demonstrate that for an agent planning actions within its risk level set $L_{\bar{p}}$, it will avoid collisions with all other agents and obstacles. To do so, we present a series of proofs that define the various risk thresholds of the system. Using two agents, we first solve for the maximum cost before collision H_c in Lemma 1. Next, we define the planning threshold H_T in Lemma 2. To construct the planning set, we require that the planning threshold $H_P \leq H_T$. In Theorem 1, we show that if an agent starts inside $L_{\bar{p}}$ and plans all future actions within their risk level set, the cost will never exceed H_c, thus avoiding collisions. From the two-agent system, we extend these results to multiple agents and obstacles in the system. We also discuss the situation when the risk level set $L_{\bar{p}}$ is empty, and the implications on the agent. This only occurs when agents in the environment have varying planning thresholds or when an agent is braking to avoid collision. However, from the assumption of self-preserving agents, collisions will still be avoided given the design of the risk thresholds.

A. Two-Agent System

Consider a two-agent system comprising the ego agent, located at x_e, and another agent, located at x_i. Without loss of generality, we can express the dynamics of the system such that the ego agent appears stationary at the origin. In this modified reference frame, we define v_i as the velocity of x_i, and a_i as the acceleration of x_i. Using these dynamics, we define R_b as the braking distance of x_i when traveling at its maximum velocity, $\dot{x}_i$, and r_c as the safety radius of the vehicle enclosing its extent. A collision is thus defined when $\|d\| \leq r_c$.

To choose a control action, we need the cost calculated at the location of the ego agent. Let $d = (x_e - x_i)$, expressed in the modified coordinate frame such that x_e is the origin. We write the cost from (1) for the two-agent system as

$$H(d, \dot{x}_i) = \frac{\exp\left(-d^T \Omega d\right)}{1 + \exp\left(-\alpha \dot{x}_i^T d\right)}. \tag{3}$$

For our ego agent to avoid collisions, we know the distance between the two agents must satisfy $\|d\| > r_c$. In Theorem 1, we show that by the properties of H, collision avoidance also means the cost H never exceeds a maximum cost H_c. From r_c, the safety radius of the vehicle, the following Lemma summarizes the calculation of H_c.

Lemma 1. *For a two-agent system with cost H defined in* (3)*, the maximum cost before collision H_c is defined*

$$H_c = \frac{\exp\left(-\frac{r_c^2}{\sigma_m^2}\right)}{1 + \exp\left(-\alpha v_i r_c\right)}, \tag{4}$$

where $\frac{1}{\sigma_m^2} = \min\left\{\frac{1}{\sigma_x^2}, \frac{1}{\sigma_y^2}\right\}$.

Proof. To calculate the maximum value of H_c before collision, we evaluate H when $\|d\| = r_c$. Let u_i represent the unit vector $u = \frac{d}{\|d\|}$. The safety radius can be written $r_c u$. Substituting this into the expression of cost (3),

$$H(r_c u, \dot{x}_i) = \frac{\exp\left(-r_c^2 u^T \Omega u\right)}{1 + \exp\left(-\alpha r_c \dot{x}_i^T u\right)}.$$

Note that the value of $u^T \Omega u$ depends on the direction of u. Define $\frac{1}{\sigma_m^2} = \min\left\{\frac{1}{\sigma_x^2}, \frac{1}{\sigma_y^2}\right\}$. We know

$$\frac{1}{\sigma_m^2} \leq u^T \Omega u,$$

thus $\exp\left(-\frac{r_c^2}{\sigma_m^2}\right) \geq \exp\left(-r_c^2 u^T \Omega u\right)$. The value of H_c will depend on the value of $\left(1 + \exp\left(-\alpha r_c \dot{x}_i^T u\right)\right)$ in the denominator. To find the largest value of H_c, we'll need to find the smallest value of the denominator. Note the value of $\exp\left(-\alpha r_c \dot{x}_i^T u\right)$ also depends on the relationship between $\dot{x}_i$ and u. If x_i is moving away from x_e, then $\dot{x}_i^T u < 0$, which drives the overall value of H toward zero. We see that $H(r_c u, \dot{x}_i)$ increases when $\dot{x}_i = vu$, where v is the magnitude of the velocity. Given a maximum velocity v_i, we find

$$H(r_c u, \dot{x}_i) \leq \frac{\exp\left(-\frac{r_c^2}{\sigma_m^2}\right)}{1 + \exp\left(-\alpha r_c v_i\right)},$$

Since H_c is the maximum value of $H(r_c u, \dot{x}_i)$, we find

$$H_c = \frac{\exp\left(-\frac{r_c^2}{\sigma_m^2}\right)}{1 + \exp\left(-\alpha r_c v_i\right)},$$

which is equivalent to (4). □

From Lemma 1, we see that the maximum cost H_c occurs when an agent is traveling at its maximum velocity at its safety radius. For vehicles with single-integrator dynamics that can stop instantaneously, our agents could generate a level set based on this threshold and avoid collisions. However, if the agents cannot stop instantaneously, then a cost of H_c would cause a collision. Instead, we also need to find the value of cost that prevents collisions given a minimum braking distance. For a braking distance R_b, Lemma 2 defines the maximum threshold cost H_T. We use H_T when choosing our planned risk threshold $H_P \leq H_T$.

Lemma 2. *For the two-agent system with cost H defined in* (3)*, the maximum threshold to avoid collisions H_T is defined*

$$H_T = \frac{\exp\left(-\frac{R_b^2}{\sigma_u^2}\right)}{2} \tag{5}$$

where $\sigma_u = \max\{\sigma_x, \sigma_y\}$.

Proof. Similar to Lemma 1, we can find the maximum threshold cost by looking at the requirements to guarantee the safe braking distance. Recall that R_b is the braking distance of x_i, such that it will never collide with x_e. To guarantee the self-preservation, the threshold cost will be the minimum value of H at this distance $\|d\| = R_b$. For the unit vector $u = \frac{d}{\|d\|}$, we can express

$$H(R_b u, \dot{x}_i) = \frac{\exp\left(-R_{b_i}^2 u^T \Omega u\right)}{1 + \exp\left(-\alpha R_b \dot{x}_i^T u\right)}.$$

In Lemma 1, we found the maximum value of H to calculate H_c. Here, we find the minimum value to calculate H_T. Note that the value of $u^T \Omega u$ depends on the orientation and values of σ_x and σ_y. However, we know

$$H(R_b u, \dot{x}_i) \geq \frac{\exp\left(\frac{-R_{b,i^2}}{\sigma_u^2}\right)}{1 + \exp\left(-\alpha R_b \dot{x}_i^T u\right)},$$

for $\sigma_u = \max\{\sigma_x, \sigma_y\}$. Here, we observe that the value of H decreases as the denominator increases. Note that for some positive constant c, $\exp(-c) \leq 1$. The largest value of the denominator occurs when $\dot{x}_i = 0$. Any velocity in the direction of u would result in a larger value of H. Thus,

$$H(R_b u, \dot{x}_i) \geq \frac{\exp\left(\frac{-R_{b,i^2}}{\sigma_u^2}\right)}{2}.$$

By choosing the smallest value of $H(R_b, \dot{x}_i)$ as H_T, we find the expression in (5). □

We propose that to avoid collisions, the ego agents stays within $L_{\bar{p}}$, their risk level set, defined in (2) by some choice of $H_P \leq H_T$. By construction, we see that the chosen value of H_P correlates with how close the ego agent will get next to other agents. Intuitively, a smaller value of H_P corresponds to larger separations between the ego agent and other agents, hence a "less risky" maneuver. In Theorem 1, we demonstrate that for agents starting in $L_{\bar{p}}$ and only planning actions that stay below this risk level set, the cost will never exceed H_c.

Theorem 1. *For an ego agent starting with some initial cost $H < H_P$ and operating within the set of points defined by $L_{\bar{p}}$, the total cost will never exceed H_c.*

Proof. Consider a two-agent system, with ego vehicle x_e and other vehicle x_i, and all dynamics written such that the ego vehicle appears stationary at the origin. Given the initial cost is below H_T, we know that $\|d\| > R_b$, the braking distance for the other vehicle. To prove that the cost will never exceed H_c, we examine its dynamics. Recall

$$H(d, \dot{x}_i) = \frac{\exp\left(-d^T \Omega d\right)}{1 + \exp\left(-\alpha \dot{x}_i^T d\right)}.$$

Now consider the function

$$W(r) = \frac{\exp\left(-\frac{r^2}{\sigma_m^2}\right)}{1 + \exp\left(-\alpha v_{\max} r\right)},$$

where $v_{\max}$ is the maximum magnitude of velocity of $\dot{x}_i$ and r is the distance between the two agents, $r = \|d\|$. By inspection,

$$W(\|d\|) \geq H(d, \dot{x}_i),$$

and that for $\|d\| = r_c$, $W(r_c) = H_c$. The derivative of W is

$$\dot{W} = \frac{-2r\dot{r}\exp\left(-\frac{r^2}{\sigma_m^2}\right)}{\sigma_m^2\left(1 + \exp\left(-\alpha v_{\max} r\right)\right)} - \frac{\left(-\alpha v_{\max} r\dot{r}\right)\exp\left(-\alpha v_{\max} r - \frac{r^2}{\sigma_m^2}\right)}{r\left(1 + \exp\left(-\alpha v_{\max} r\right)\right)^2}.$$

From the derivative, $\dot{W}$ is not always decreasing as r increases. We can rewrite $\dot{W}$ as

$$\dot{W} = \left[\frac{-2}{\sigma_m^2} + \frac{\alpha v_{\max}\exp\left(-\alpha v_{\max} r\right)}{r\left(1 + \exp\left(-\alpha v_{\max} r\right)\right)}\right]\beta,$$

where

$$\beta = \frac{r\dot{r}\exp\left(\frac{-r^2}{\sigma_m^2}\right)}{\left(1 + \exp\left(-\alpha v_{\max} r\right)\right)} \geq 0.$$

The sign of $\dot{W}$ then depends on choosing an α such that

$$\frac{\alpha v_{\max}\exp\left(-\alpha v_{\max} r_c\right)}{\left(1 + \exp\left(-\alpha v_{\max} r_c\right)\right)} < \frac{2r_c}{\sigma_m^2}.$$

Thus, given the values of r_c, σ_m, and $v_{\max}$, we can choose α such that such that $\dot{W} < 0$ for $r > r_c$.

Since $W(r)$ is always greater than H for $\|d\| > r_c$, and $\dot{W} < 0$ for increasing r, then $H(d, \dot{x}_i) < H_c$ for any$\|d\| > r_c$. Thus, the cost $H(d, \dot{x}_i)$ will never exceed H_c unless the agents collide. For agents operating within $L_{\bar{p}}$, to stay in $L_{\bar{p}}$ requires only choosing points such that $H(\cdot) < H_P < H_c$, thus any agent starting in $L_{\bar{p}}$ will never choose an action that results in $H > H_c$, completing the claims of Theorem 1.

□

As shown in Theorem 1, an agent starting within $L_{\bar{p}}$ will never exceed the collision threshold H_c. If all agents in the system stay within their respective thresholds $L_{\bar{p},i}$, then all agents will never leave these sets. However, we do not assume all agents in the system use the same control laws, only that they are self-preserving. Due to this property, there are situations in which the ego agent experiences a cost greater than H_P, but never exceeding H_T. Consider the scenario of the ego agent directly following the other agent in the environment. If the other agent brakes suddenly, the ego agent will experience an increase in cost as it moves towards the other agent. However, since H_T is chosen to include a braking distance, then as soon as the ego agent sees $H > H_P$, it can begin braking to avoid collision and maintain $H < H_c$.

B. Multiple Agents

While the previous section provides collision avoidance guarantees for two agents, our goal is to find an algorithm to work in highly cluttered environments with many agents. Using the claims from Theorem 1 for two agents, this section extends those results to multiple agents in the environment. For two agents, we introduced H, which maps the position of the other agent to an occupancy cost in the environment,

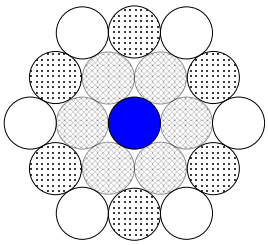


Fig. 2: Illustration of the hexagonal pattern of circle packing. The blue agent in the center is the ego agent.

allowing us to define a risk threshold and $L_{\bar{p}}$. When considering multiple agents, a simple approach is to consider all the agents as many pairwise interactions. For $i = \{1, ..., n\}$ other agents, let $d_i = (x_e - x_i)$, $H_{T,i}(d_i, \dot{x}_i)$ be the threshold between two agents, and $H_{c,i}$ be the collision cost. The risk level set is $L_{\bar{p}}$ can then be expressed as the intersection of all pairwise level sets,

$$L_{\bar{p}} = \bigcap_{i=1}^{n} L_{\bar{p},i}, \tag{6}$$

where $L_{\bar{p},i}$ is defined in (2). When defined in this fashion, Corollary 1 extends the claims of Theorem 1 to the multi-agent scenario.

Corollary 1. *For an ego agent x_e and a group of n agents with positions x_i, let $L_{\bar{p}}$ represent planning set of agent x_e. By planning within $L_{\bar{p}}$, agent x_e avoids collisions with all other n agents in the environment.*

Proof. By construction, $L_{\bar{p}}$ is the intersection of all pairwise risk sets. For a point q to be in this set, the individual agent cost $H_i(q - x_i, \dot{x}_i) < H_T$. By Theorem 1, when the ego agent plans within this risk threshold, the maximum pairwise cost will never exceed $H_{c,i}$, ensuring the ego agent does not collide with the agent x_i. With this pairwise approach, the ego agent avoids collisions with all agents. □

While Corollary 1 demonstrates a method for avoiding collisions in a pairwise fashion, to compute $L_{\bar{p}}$ as the intersection of all individual sets becomes intractable with many agents. Instead, we wish to compute a single cost across the environment, and find a single risk level set within that cost. Consider the cost as the sum of all agents in the environment. Let

$$H = \sum_{i=1}^{n} H_i(d_i, \dot{x}_i) = \sum_{i=1}^{n} \frac{\exp\left(-d_i^T \Omega d_i\right)}{1 + \exp\left(-\alpha \dot{x}_i^T d_i\right)}, \tag{7}$$

where n is the number of other agents in the field, and $d_i = (x_e - x_i)$, the pairwise distance between the ego agent x_e and other agent x_i. Here, we assume that Ω, α, $v_{\max}$, and the collision radius r_c are equal for all agents in the group. From the assumption that all agents have the same safety radius, we show that the level set of cost can be determined from a circle-packing problem. It turns out that for all agents with safety radius r_c, the hexagonal packing pattern is the densest arrangement of agents. In this pattern, at most six other agents of radius r_c are positioned around the ego agent, and another twelve agents are arranged within $2r_c$ distance from the ego agent, as illustrated in Figure 2.

From this hexagonal pattern in Figure 2, we can approximate the maximum cost H_c our ego agent may encounter, based on a worst-case scenario of all agents packed together. We find

$$\begin{aligned} H_c \approx & \sum_{i=1}^{6} \frac{\exp\left(-\frac{r_c^2}{\sigma_m^2}\right)}{1 + \exp\left(-\alpha v_{\max} r_c\right)} \\ & + \sum_{i=7}^{12} \frac{\exp\left(-\frac{(1.5r_c)^2}{\sigma_m^2}\right)}{1 + \exp\left(-1.5\alpha v_{\max} r_c\right)} \\ & + \sum_{i=13}^{18} \frac{\exp\left(-\frac{(2r_c)^2}{\sigma_m^2}\right)}{1 + \exp\left(-2\alpha v_{\max} r_c\right)} + ... \end{aligned} \tag{8}$$

In this summation, six agents are arranged at r_c, shaded with a crosshatch in Figure 2. Six more agents are a distance $1.5r_c$, shaded with dots, and six agents are a distance $2r_c$ away, shown in white in Figure 2. We can continue calculating this by adding more agents around the ego agent in the hexagonal pattern. Note the approximation of H_c is determined by the number of agents and the packing density. For practical purposes in highly-cluttered environments, H_c may be approximated by a fixed number of agents. Similarly, we can approximate H_T from (5) for multiple agents as

$$\begin{aligned} H_T \approx & \sum_{i=1}^{6} \frac{\exp\left(-\frac{R_b^2}{\sigma_u^2}\right)}{2} \\ & + \sum_{i=7}^{12} \frac{\exp\left(-\frac{(1.5R_b)^2}{\sigma_m^2}\right)}{2} + \sum_{i=13}^{18} \frac{\exp\left(-\frac{(2R_b)^2}{\sigma_m^2}\right)}{2} + ... \end{aligned} \tag{9}$$

For planning purposes, approximating H_T by truncating the series expansion will yield a more conservative estimate. For example, letting

$$H_P \leq \frac{3\exp\left(-\frac{R_b^2}{\sigma_u^2}\right)}{2}$$

will guarantee that $H_P < H_T$ for any group size of $n \geq 6$ other agents in the environment.

IV. Planning in Risk Level Sets

The previous section introduced our definition of risk level sets, and proposed that an agent can use the risk level sets to guarantee collision avoidance. In this section, we provide an example of one such planning algorithm. For this example, consider the problem of an autonomous vehicle planning a sequence of lane changes along a highway. The ego vehicle wants to minimize its travel time along a segment, but also must avoid collisions with the other cars. We use the risk level sets to both check for collision avoidance during individual lane changes, as well as provide a metric of congestion for overall route planning.

As the previous section discussed, the ego agent uses the cost H to determine some planning threshold, H_P, less than the threshold cost H_T required for self-preserving agents. Figure 3 demonstrates various contours of H around obstacle cars. In Figure 3, the value of H_c corresponds to the smallest

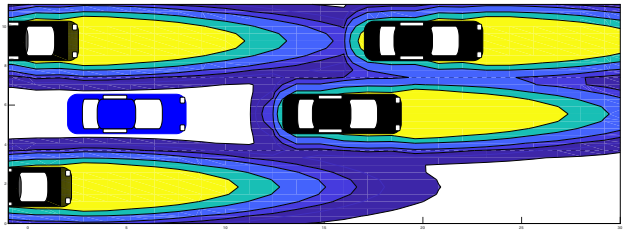


Fig. 3: Illustration of various contour levels. Each color represents a different level of H, with H_c denoted in yellow. As H decreases, the region around the car increases.

yellow region, while lower values of H create a larger region around the obstacle. By choosing a smaller H_P, the ego car leaves more room around the other cars, resulting in less risky maneuvers. Also note the contours of H are all skewed along the forward travel direction of the vehicle. Intuitively, when an agent is planning to change lanes, it needs to avoid cutting off a vehicle in front.

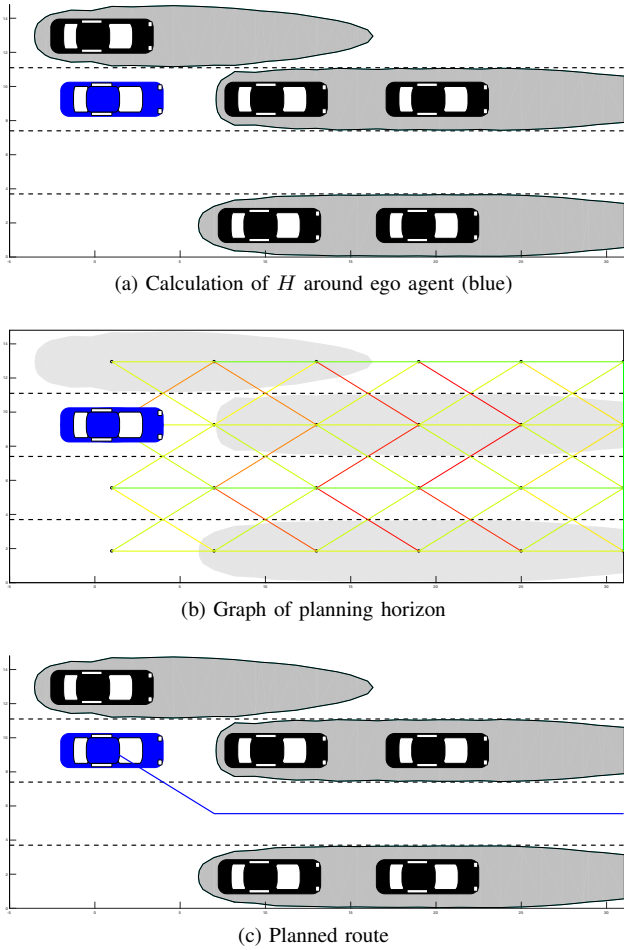
Our planner works in several stages, summarized in Algorithm 1. Using a short planning horizon, we discretize the horizon into a graph, with edges representing straight travel and lane changes. Using the calculated cost, we map the values of the cost function to weights along the edges. Using Djikstra's Algorithm, we find the shortest path along the planning horizon, representing the sequence of desired lane changes that accounts for travel time and risk in changing lanes. From the suggested sequence of lane changes, the ego agent only makes an individual lane change when it can do so without violating any risk thresholds. Since we update the planned route regularly, the path will change based on variations in traffic and relative lane speeds.

Algorithm 1 Sequential Lane Changes

1: Computer overall cost H (7)
2: Compute $L_{\bar{p}}$ (2)
3: Create graph within planning horizon
4: Map H and $L_{\bar{p}}$ to edge weights
5: Use Djikstra's to compute sequence of lane changes
6: Check that single lane change does not violate $L_{\bar{p}}$

Within our planner, the overall cost H is computed across the planning horizon from observed traffic. By construction, our ego vehicle does not need perfect localization of the other vehicles, but can instead use an estimated position skewed by the observed velocity. From H, the agent calculates $L_{\bar{p}}$, based on a chosen planning threshold H_P. In our Section V, we demonstrate in simulation how the choice of H_P affects the perceived aggressiveness of the ego vehicle.

To calculate the sequence of lane changes, we choose to discretize the horizon into a weighted graph and calculate the lowest-cost path. Along the planning horizon, nodes are placed in a grid pattern, centered in each lane and spaced a distance D horizontally. We classify the edges as either "straight travel" or as a"lane change." As the naming suggests, straight segments connect nodes sequentially within a lane, which the lane change nodes connect diagonally to the



(a) Calculation of H around ego agent (blue)

(b) Graph of planning horizon

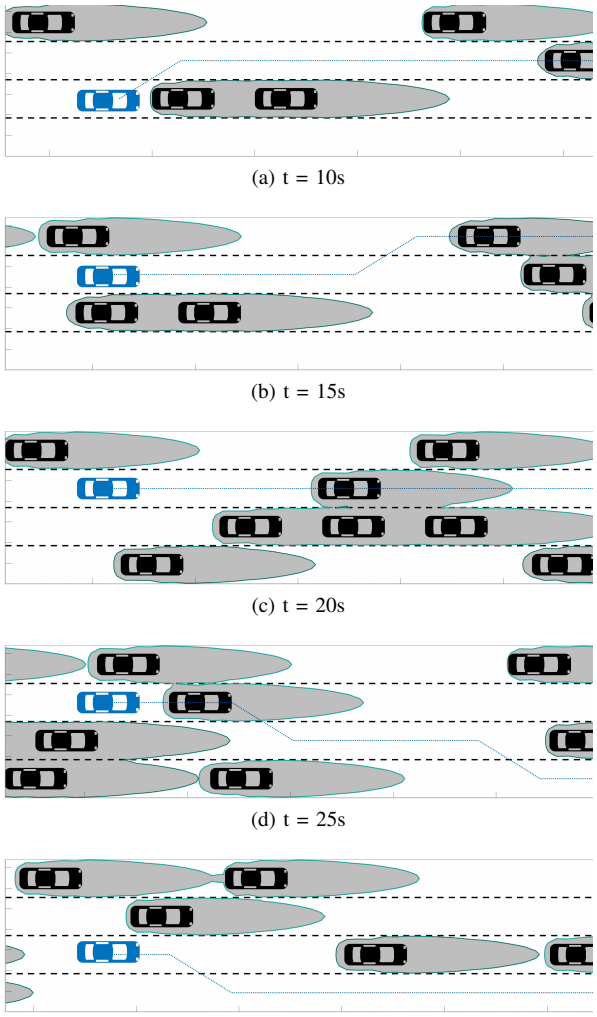
(c) Planned route

Fig. 4: Snapshot of planning algorithm with (a) values of $H > H_P$ shaded in gray, (b) the planning graph with the colors of the edges correspond to their relative weight, and (c) computed lane change for car.

adjacent lanes. To assign weights, the value of straight edges is based on whether the node is occupied (indicated by the cost function), and the difference in vehicle travel speed with some desired speed. For unoccupied nodes, straight segments are assigned a base value of B. For the lane changes, the edge weight is calculated on the occupancy between nodes, given by the cost H. For segments connecting unoccupied lane changes, the base value is $2B$, ensuring that a lane change will have a higher cost than a straight segment when no other cars are present. Figure 4 illustrates an example of the cost and the planned trajectory.

V. Simulations

To demonstrate the performance of our route-planning algorithm, we conducted simulations in Matlab. A video of the simulations is included with the submission of this paper. First, we present a single example of the lane change algorithm, illustrating how the car adjusts its route based on local cars and its planning horizon. Next, we present results summarizing the behavior in various traffic densities and risk thresholds. Finally, we comment on the extension to multiple cars simultaneously planning routes, and show that the cars



(a) t = 10s

(b) t = 15s

(c) t = 20s

(d) t = 25s

(e) t = 30s

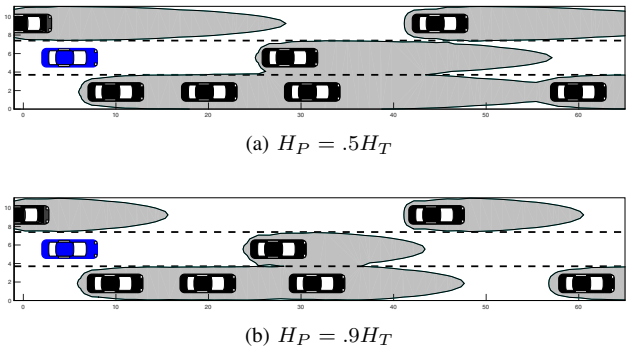
Fig. 5: Simulation of the ego car (blue) planning a path through traffic. The planned sequence of lane changes is shown by the dotted blue line. The contour of $H > H_P$ is shaded in grey.

naturally spread out and re-adjust based on the behavior of the other agents.

Figure 5 illustrates a car (blue) navigating a 4-lane highway. In the figure, the planned sequence of lane changes is drawn as the dashed blue line. The contour of $H > H_P$ is shaded grey, representing points outside $L_{\bar{p}}$. From Figure 5, we see the route changes as the agent moves through traffic and as openings appear on its planning horizon.

A. *Impact of Risk Threshold and Traffic Density*

To demonstrate how using different risk level sets affects performance, we compared several scenarios for a "conservative" driver and an "aggressive" ego agent. For each scenario, the ego car travels $2000m$ on a 4-lane segment of highway. To increase the number of lane changes, the ego car has a maximum desired speed of $40m/s$. The other cars are given a maximum speed based on their lane assignment, such that the fastest lane travels at $29m/s$ and the slowest lane travels at $17m/s$. Without any traffic, the expected travel time along



(a) $H_P = .5H_T$

(b) $H_P = .9H_T$

Fig. 6: The difference in $L_{\bar{p}}$ for the conservative and aggressive driver. The gray zone indicates values of $H > H_P$. Note the car in (b) has a much larger planning space $L_{\bar{p}}$ (white area) than the car in (a).

	$H_P = .9H_T$	$H_P = .5H_T$
$n = 100$	$56.9s$	$62.7s$
$n = 150$	$63.4s$	$68.1s$
$n = 200$	$67.2s$	$69.2s$

TABLE I: Average travel time for the ego car across 100 trials.

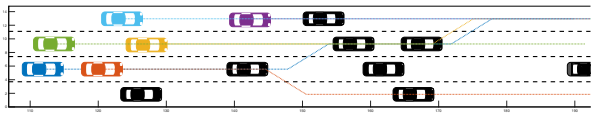
this segment for the ego car is 50 seconds, while the expected travel time for an obstacle car in the "fast" lane is 69 seconds. We benchmark the performance of the ego car by its travel time along the segment, which demonstrates how effective it is at weaving through traffic.

To compare the performance among each scenario, we examine both the average time to travel the distance, as well as the average number of lane changes. Table I summarizes the average time, and Table II summarizes the average number of lane changes. For each scenario, we ran 100 trials with randomized initial conditions. The ego car computes $L_{\bar{p}}$ from either $H_P = .9H_T$ for the "aggressive" driver, or $H_P = .5H_T$ for the "conservative" driver. Figure 6 illustrates the difference in $L_{\bar{p}}$ for the variations in H_P. In Figure 6, the higher value of H_P results in a gap to change lanes, whereas with the lower value of H_P, the car does not have a gap. We also vary the number of obstacle cars from $n = \{100, 150, 200\}$ to simulate moderate to high congestion.

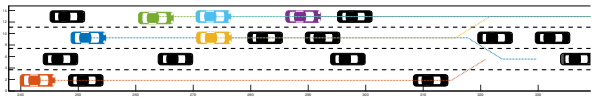
From Table I, for both values of H_P, we see that the average travel time increases as the congestion increases. By $n = 200$, the average travel time is almost equal to the expected travel time of the "fast" lane. We expect this result, as in dense traffic, the car does not have sufficient empty

	$H_P = .9H_T$	$H_P = .5H_T$
$n = 100$	13.4	9.7
$n = 150$	14.2	8.4
$n = 200$	12.8	7.0

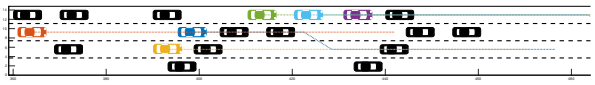
TABLE II: Average number of lane changes the ego car makes across 100 trials.



(a) t = 5s



(b) t = 10s



(c) t = 15s

Fig. 7: For 6 agents using the route planner, each car treats the other cars as obstacles. Over time, the cars adjust and disperse, based on lane changes occurring ahead.

lane space to travel at its faster desired speed and pass other cars. From Table II, we see a key difference between the aggressive and conservative driver, namely, the aggressive driver attempts more lane changes. This result is expected, as the higher H_P value corresponds to a larger $L_{\bar{p}}$ set, and as we saw in Figure 6, more opportunities to make lane changes.

B. Multiple Agents

Our algorithm is able to handle multiple cars simultaneously planning their individual routes. In this case, the other cars are still treated as dynamic obstacles, and when forward cars change lanes, the planner adjusts its path accordingly. Figure 7 shows 6 agents simultaneously planning routes along the same section of highway, with the active cars shown in various colors, and the other traffic shown in black. Notice that while the purple car in front stays in the fast lane, the rear cars perform more passing maneuvers and updates as the cars in front change lanes. Since this algorithm is based on the density, we see the cars spread out among the different lanes instead of all moving to the "fast" lane.

VI. Conclusions

This paper addresses the problem of navigating a cluttered environment by first introducing a congestion cost, then choosing planners that operate within "risk level sets" of this cost. The congestion cost maps the density and motion of the objects to an occupancy risk for the agent. In choosing different levels of risk, the agents adjust its interactions with other agents. We show that any agent planning within their risk level set avoids collisions with other agents. We demonstrate planning in congestion through the example of an autonomous vehicle planning a sequence of lane changes along a highway. Using the risk level sets, the agents determine safe zones to make lane changes. In simulations, we illustrate how a higher risk threshold results in more aggressive behavior of the vehicle, as well as an increase in number of lane changes as it weaves through traffic. Similarly, a lower risk threshold results in more conservative behavior with fewer lane changes. Future work will relax the assumption of self-preserving agents and examine how the agents interact in other environments. We will also examine advanced dynamics on the agents, and how level set calculations may change with more information on the reachable set of an agent.

References

[1] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 65–100, 2010.

[2] C. Schlegel, "Fast local obstacle avoidance under kinematic and dynamic constraints for a mobile robot," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 1. IEEE, 1998, pp. 594–599.

[3] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1, 1999, pp. 341–346.

[4] P. Ogren and N. E. Leonard, "A convergent dynamic window approach to obstacle avoidance," *Robotics, IEEE Transactions on*, vol. 21, no. 2, pp. 188–195, 2005.

[5] C. Belta, V. Isler, and G. J. Pappas, "Discrete abstractions for robot motion planning and control in polygonal environments," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 864–874, Oct 2005.

[6] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*, Dec 2014, pp. 6271–6278.

[7] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68 – 73, 2015, analysis and Design of Hybrid Systems ADHS.

[8] D. Panagou, D. M. Stipanovi, and P. G. Voulgaris, "Distributed coordination control for multi-robot networks using lyapunov-like barrier functions," *IEEE Transactions on Automatic Control*, vol. 61, no. 3, pp. 617–632, March 2016.

[9] W. Schwarting and P. Pascheka, "Recursive conflict resolution for cooperative motion planning in dynamic highway traffic," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct 2014, pp. 1039–1044.

[10] P. Wagner, K. Nagel, and D. E. Wolf, "Realistic multi-lane traffic rules for cellular automata," *Physica A: Statistical Mechanics and its Applications*, vol. 234, no. 3, pp. 687 – 698, 1997.

[11] D. Chowdhury, D. E. Wolf, and M. Schreckenberg, "Particle hopping models for two-lane traffic with two kinds of vehicles: Effects of lane-changing rules," *Physica A: Statistical Mechanics and its Applications*, vol. 235, no. 3, pp. 417 – 439, 1997.

[12] K. Ahmed, M. Ben-Akiva, H. Koutsopoulos, and R. Mishalani, "Models of freeway lane changing and gap acceptance behavior," *Transportation and traffic theory*, vol. 13, pp. 501–515, 1996. [Online]. Available: http://dx.doi.org/

[13] H. Jula, E. B. Kosmatopoulos, and P. A. Ioannou, "Collision avoidance analysis for lane changing and merging," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pp. 2295–2308, Nov 2000.

[14] P. Hidas, "Modelling lane changing and merging in microscopic traffic simulation," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 56, pp. 351 – 371, 2002.

[15] J. E. Naranjo, C. Gonzalez, R. Garcia, and T. de Pedro, "Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 438–450, Sept 2008.

[16] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *CoRR*, vol. abs/1610.03295, 2016. [Online]. Available: http://arxiv.org/abs/1610.03295

[17] G. Schildbach and F. Borrelli, "Scenario model predictive control for lane change assistance on highways," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 611–616.

[18] A. Lawitzky, D. Althoff, C. F. Passenberg, G. Tanzmeister, D. Wollherr, and M. Buss, "Interactive scene prediction for automotive applications," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, June 2013, pp. 1028–1033.