# Robot Motion Planning: A Game-Theoretic Foundation

Steven M. LaValle, *Stanford University, Stanford, CA USA*

*This paper proposes a dynamic game-theoretic framework that is used as an analytical tool and unifying perspective for a wide class of problems in motion planning. This approach is inspired by the foundation laid by configuration-space concepts for basic path planning. In the same manner that configuration-space concepts led to substantial progress in path planning, game-theoretic concepts provide a more general foundation which can incorporate any of the essential features of path planning, sensing uncertainty, decision theory, bounded-uncertainty analysis, stochastic optimal control, and traditional multiplayer games. By following this perspective, new modeling, analysis, algorithms, and computational results have been obtained for a variety of motion planning problems including those involving uncertainty in sensing and control, environment uncertainties, and the coordination of multiple robots.*

## 1 Introduction

It is widely accepted that the configuration-space (C-space) representation has provided a powerful, unified foundation for the development and analysis of motion planning algorithms. In spite of this success, there has been little attempt to obtain further benefits by broadening this foundation into a common mathematical structure that encompasses many important, well-studied extensions of the basic planning problem. This paper proposes such a foundation by combining decision-theoretic concepts from areas such as dynamic game theory and stochastic optimal control with C-space concepts. The intent is not to provide an alternative formulation of motion planning, but instead to present an expanded foundation that is built on previous geometric concepts, while characterizing and unifying a broader class of problems.

The *basic* motion planning problem has been to determine a continuous, collision-free path that connects an initial configuration to a goal configuration. This problem was created by modularizing robotic tasks to isolate path planning from lower-level trajectory tracking. If the modularization is completely removed, many robotic tasks can be considered as a nonlinear control problem for which there are complicated constraints on the state space (which include the constraints due to static obstacles). Although basic motion planning has found many direct applications, there are fundamental limitations that have motivated many specific approaches to handle difficult extensions of the basic problem. Let *general* motion planning include complications such as sensing uncertainties, prediction uncertainties, nonholonomy, dynamics, performance criteria, and multiple robots with independent goals.

Several benefits arose from the use of the C-space representation for basic motion planning. The comparison of seemingly disparate approaches, such as cell decomposition methods, roadmap methods, and artificial potential field methods, was greatly facilitated through the use of C-space representations [22],[29]. Concepts such as *completeness* and *resolution completeness* were formulated in terms of configuration space, and hence applied to a wide class of problems and algorithms. Planning algorithms could also be generalized by utilizing the common C-space foundation. For instance, a randomized potential field planner was applied with only minor adaptations to a variety of problems ranging from multiple rigid robots to high degree-of-freedom manipulators [4]. Wide applicability is obtained because all problems are reduced to C-space terms.

The same basic philosophy can be preserved for general motion planning by using a broader mathematical foundation that provides the same types of benefits that configuration-space concepts provided for the basic planning problem. It is important to note, however,

that this paper emphasizes a *mathematical foundation* as opposed to a particular model or computational approach. Algorithms and computed examples are presented in this paper for the purpose of demonstrating the power of this foundation, to encourage its use in future motion planning research. More details on the specific algorithms, analysis, and computed examples appear in [24] and [25]-[28], [37].

## 2 Mathematical Formulation

Before considering a formulation of general motion planning, first consider making small extensions to the basic motion planning problem. The basic problem is to find a continuous path $x : [0, t_f] \rightarrow \mathcal{C}_{free}$ such that $x(0) = q_{init}$ and $x(t_f) = q_{goal}$. Recall that $\mathcal{C}_{free}$ implicitly incorporates all of the constraints due to the robot geometry and static obstacles in the workspace.

Suppose that there are nonholonomic constraints. To facilitate upcoming concepts, let $\mathcal{C}_{free}$ be renamed as a generic state space, $X = \mathcal{C}_{free}$. It is well known that the nonholonomic constraints can be expressed as $\dot{x} = f(x(t), u(t))$, which constrains the allowable vector fields on $X$. Instead of directly choosing $x(t)$, one is forced to interact with the system using the input (or action) $u(t)$. This occurs, for example, when manipulating an object through pushing [31]. If $f(x(t), u(t)) = u(t)$, the original nonholonomic, basic motion planning problem is obtained since any desired, collision-free path in the state space can be obtained by selecting an appropriate input.

Suppose that optimality with respect to some criterion, such as path length or execution time, is important. A *loss functional* can be defined that evaluates any state trajectory and input:

$$L(x(\cdot), u(\cdot)) = \int_0^{t_f} l(x(t), u(t))dt + Q(x(t_f)). \quad (1)$$

The integrand $l(x(t), u(t))$ allows the specification of a cost that will accumulate during execution and will depend in general on the state trajectory and the input. The final term $Q(x(t_f))$ can indicate the importance of achieving the goal. The basic problem can be considered as a special form of optimal control [13]. Suppose $l(x(t), u(t)) \equiv 0$, and $Q(x(t_f)) = 0$ if $x(t_f) = q_{goal}$ and $Q(x(t_f)) = 1$ otherwise. This corresponds to the original case in which optimality is not important. The space of possible inputs to the system in this case is partitioned into two classes: those that lead to the goal region, and those that fail.

Next, consider moving to a mathematical structure for the general motion planning problem, which is based on concepts from dynamic noncooperative game theory [1] and stochastic optimal control [20]. This structure will be formulated in discrete time to ease the specification of uncertainty aspects; however, continuous time can alternatively be used with some minor modifications. Thirteen components are first listed, and a discussion of each in relation to motion planning follows.

1. An index set, $\mathbf{N} = \{1, 2, \ldots, N\}$, of $N$ decision makers

2. An index set, $\mathbf{K} = \{1, 2, \ldots, K\}$, that denotes the *stages* of the game

3. A set, $X$, called the *state space*. The state of the game, $x_k$, at stage $k$, belongs to $X$.

4. A set, $U_k^i$, defined for each $k \in \mathbf{K}$ and $i \in \mathbf{N}$, which is called the *action set* of the $i^{th}$ decision maker at stage $k$. The *action*, $u_k^i$, at stage $k$, belongs to $U_k^i$.

5. A set, $\Theta_k^a$, defined for each $k \in \mathbf{K}$, which is called the *control action set for nature* at stage $k$. The *control action for nature*, $\theta_k^a$, at stage $k$, belongs to $\Theta_k^a$.

6. A function, $f_k : X \times U_k^1 \times \ldots \times U_k^N \times \Theta_k^a \rightarrow X$, defined for each $k \in \mathbf{K}$ so that

$$x_{k+1} = f_k(x_k, u_k^1, \ldots, u_k^N, \theta_k^a), \quad (2)$$

   is a *state transition equation.*

7. A set, $Y_k^i$, defined for each $k \in \mathbf{K}$ and $i \in \mathbf{N}$, and called the *sensor space* of the $i^{th}$ decision maker at stage $k$, to which the sensed observation $y_k^i$ belongs at stage $k$.

8. A set, $\Theta_k^{s,i}$, defined for each $i \in \mathbf{N}$ and $k \in \mathbf{K}$, which is called the *sensing action set for nature* at stage $k$. The *sensing action for nature*, $\theta_k^{s,i}$, at stage $k$, belongs to $\Theta_k^{s,i}$.

9. A function, $h_k^i$, defined for each $k \in \mathbf{K}$ and $i \in \mathbf{N}$, so that
$$y_k^i = h_k^i(x_k, \theta_k^{s,i}), \tag{3}$$
which is the *observation equation* of the $i^{th}$ decision maker concerning the value of $x_k$.

10. A finite set, $\eta_k^i$, defined for each $k \in \mathbf{K}$ and $i \in \mathbf{N}$ as a subset of all actions and observations made by decision makers at any previous stage, $\{u_1^1, \ldots, u_{k-1}^N, y_1^1, \ldots, y_k^N\}$.

11. A set of all possible values for $\eta_k^i$, denoted by $N_k^i$, which is called the *information space* for the $i^{th}$ decision maker at stage $k$.

12. A set, $\Gamma_k^i$, of mappings $\gamma_k^i : N_k^i \rightarrow U_k^i$, which are the *strategies* available to the $i^{th}$ decision maker at stage $k$. The combined mapping $\gamma^i = \{\gamma_1^i, \gamma_2^i, \ldots, \gamma_K^i\}$ is a *strategy* for the $i^{th}$ decision maker, and the set $\Gamma^i$ of all such mappings $\gamma^i$ is the *strategy space* of the $i^{th}$ decision maker. A *game strategy*, $\gamma$, represents a simultaneous specification of the strategy for each decision maker, and the space of game strategies is denoted by $\Gamma = \Gamma^1 \times \cdots \times \Gamma^N$.

13. An (extended) real-valued functional $L^i : (X \times U_1^1 \times \ldots \times U_1^N) \times (X \times U_2^1 \times \ldots \times U_2^N) \times \ldots \times (X \times U_K^1 \times \ldots \times U_K^N) \times \Theta \rightarrow \Re^+$, defined for each $i \in \mathbf{N}$, and called the *loss functional* of the $i^{th}$ decision maker. The Cartesian product of all of nature's action spaces is represented here as $\Theta$.

**State transitions and control** Item 1 defines the decision makers, which each typically refers to an independent, controllable robot. In general, however, any agent that is capable of making decisions and interfering with the other decision makers can be considered as a decision maker.

Item 2 defines stages that correspond to times at which decisions are made. For standard discrete-time analysis, decisions are made at each $\Delta t$ time increment. The limiting case of $K = \infty$ can be defined. In general, decision making at regular intervals is not required. Suppose for instance that the decisions correspond to very high-level operations, which may have unpredictable completion times. This case is discussed in more detail in [37], for modeling the completion of a fine-motion operation. A continuum of stages can alternatively be considered, which results in a continuous-time differential game (e.g., [18]).

The state space is defined in Item 3. At the very least, the state space can be used to represent the free configuration space, $\mathcal{C}_{free}$. In the case of multiple robots, it can represent the composite configuration space that is formed by taking the Cartesian product of the configuration spaces of the individual robots. In general, however, the state space could incorporate additional information. For instance, dynamics can be included by expanding the state space to include configuration time derivatives. This corresponds to the standard use of state space representations in optimal control theory. The state space can also include any parameters that can be completely or partially controlled through the operation of the robot(s). In one application [28] the state space includes *environment modes* that characterize varying conditions in the environment that potentially affect the robot.

Item 4 defines the set of actions that are available to each decision maker at each stage.

Item 5 is used to model sources of uncertainty. Two common representations of uncertainty have been applied to motion planning problems. With a *nondeterministic* (or bounded-set) representation parameter uncertainties are restricted to lie within a specified set. A motion plan is then generated that is based on *worst-case* analysis (e.g., [6], [12], [23], [30]). With a *probabilistic* representation the parameter uncertainties are characterized with a probability density function (pdf). This often leads to the construction of motion plans through *average-case* or *expected-case* analysis (e.g., [5], [15]).

One key aspect of the proposed mathematical foundation is a general capacity to model uncertainties. This is accomplished by introducing a decision maker referred to as *nature*. It will be assumed that no one has complete control over actions that are chosen by nature; however, models can be constructed to partially predict nature's actions. Nature can introduce nondeterministic or probabilistic uncertainties into the game by applying either *control actions* or *sensing actions*. Item 5 defines the set of control actions that are available to nature, and Item 8 will define the set of sensing actions that are available to nature.

Item 6 defines how changes in state are effected. The state $x_{k+1}$, at stage $k+1$, is obtained as a function of

the previous state $x_k$ and the actions chosen by all decision makers, including nature. If nature is omitted from the state transition equation, then perfect prediction of future states is possible, given the actions of the decision makers. Under nondeterministic uncertainty, a set of possible future states can be derived from the state transition equation as:
$F_{k+1}(x_k, u_k^1, \ldots, u_k^N) =$

$$\{f(x_k, u_k^1, \ldots, u_k^N, \theta_k^a) \in X | \theta_k^a \in \Theta_k^a\}. \quad (4)$$

Under probabilistic uncertainty, it is assumed that $p(\theta_k^a)$, is known. By using the state transition equation, the next state is represented by a pdf, $p(x_{k+1}|x_k, u_k^1, \ldots, u_k^N)$.

**A control example** As an example of a state transition equation with uncertainty, consider characterizing the uncertainty model that is used for motion control in preimage planning research (e.g., [12], [23], [30]). Suppose there is a single decision maker that is a polygonal robot translating in the plane amidst polygonal obstacles. The action set defines commanded velocity directions, which can be specified by an orientation, yielding $U = [0, 2\pi)$ (for the case of a single decision maker, the superscripts will be dropped). The robot will attempt to move a fixed distance $\|v\|\Delta t$ (expressed in terms of a constant velocity modulus, $\|v\|$) in the direction specified by $u_k$. The action space of nature is a set of angular displacements $\theta_k^a$, such that $-\epsilon_\theta \leq \theta_k^a \leq \epsilon_\theta$, for some maximum angle $\epsilon_\theta$. Under nondeterministic uncertainty, any action $\theta_k^a \in [-\epsilon_\theta, \epsilon_\theta]$ can be chosen by nature. When using probabilistic uncertainty, $p(\theta_k^a)$ could be a continuous pdf, which is zero outside of $[-\epsilon_\theta, \epsilon_\theta]$. If the robot chooses action $u_k$ from state $x_k$, and nature chooses $\theta_k^a$, then $x_{k+1}$ is given by

$$f(x_k, u_k, \theta_k^a) = x_k + \|v\|\Delta t \begin{bmatrix} cos(u_k + \theta_k^a) \\ sin(u_k + \theta_k^a) \end{bmatrix}. \quad (5)$$

**Sensing uncertainty** Items 7 through 11 characterize the information that can be used for the basis of decision making when there is not direct access to the state. This can be considered as a general form of the sensing problem in robotics. Each decision maker at each stage has a *sensor space*, $Y_k^i$ (or observation space), which encodes information regarding the state that is observed during stage $k$. This type of projection is used in optimal control theory to define system outputs, and has also been considered in robot sensing problems (see, for instance, [8]). In addition to a projection from the state space to the sensor space, this information is potentially corrupted by a sensing action, $\theta_k^{s,i}$, of nature, which is chosen from $\Theta_k^{s,i}$.

Under nondeterministic uncertainty, the possible current states from a single sensor observation are

$$F_k^i(y_k^i) = \{x_k \in X | y_k^i = h_k^i(x_k, \theta_k^{s,i}), \theta_k^{s,i} \in \Theta^{s,i}\}. \quad (6)$$

Under probabilistic uncertainty the current state, assuming only a single observation, is represented by a pdf, $p(x_k|y_k)$.

The sensing model can be generalized to include state history, $y_k^i = h_k^i(x_1, \ldots, x_k, \theta_k^{s,i})$.

**A sensing example** Consider representing the sensing model used in [5], [12], [23]. Suppose that a single robot is equipped with a position sensor and a force sensor. Assume that the position sensor is calibrated in the configuration space, yielding values in $\Re^2$. The force sensor provides values in $[0, 2\pi) \cup \{\emptyset\}$, indicating either the direction of force when the robot is in contact with an obstacle, or no force (represented by $\emptyset$) when the robot is in the free space.

Independent portions of the observation equation are considered: $h^p$ for the position sensor and $h^f$ for the force sensor (which together form a three-dimensional vector-valued function). The sensing action of nature, $\theta_k^s$, are partitioned into subvectors $\theta_k^{s,p}$ and $\theta_k^{s,f}$, which act on the position sensor and force sensor, respectively. The observation for the position sensor is $y_k^p = h^p(x_k, \theta_k^{s,p}) = x_k + \theta_k^{s,p}$. Under nondeterministic uncertainty, $\theta_k^{s,p}$ could be any value in $\Theta_k^{s,p}$. If probabilistic uncertainty is used, a pdf is presented, such as

$$p(\theta_k^{s,p}) = \begin{cases} \frac{2}{\pi\epsilon_p^2} & \text{for } \|\theta_k^{s,p}\| < \epsilon_p \\ 0 & \text{otherwise} \end{cases}. \quad (7)$$

In (7) a radius $\epsilon_p$ is specified, and $\theta_k^{s,p}$ is two-dimensional.

One of two possibilities is obtained for the force sensor: (1) a value in $[0, 2\pi)$, governed by $y_k^f = h^f(x_k, \theta_k^{s,f}) = \alpha(x_k) + \theta_k^{s,f}$, in which $x_k \in \mathcal{C}_{contact}$ (i.e., the position lies in the boundary of $\mathcal{C}_{free}$), and the true normal is given by $\alpha(x_k)$, or (2) an empty value,

$\emptyset$, when the robot is in $\mathcal{C}_{free}$. When the robot configuration lies in $\mathcal{C}_{contact}$ and probabilistic uncertainty is in use, then the pdf can be represented as

$$p(\theta_k^{s,f}) = \begin{cases} \frac{1}{2\epsilon_f} & \text{for } |\theta_k^{s,f}| < \epsilon_f \\ 0 & \text{otherwise} \end{cases}, \qquad (8)$$

for some positive prespecified constant $\epsilon_f < \frac{1}{2}\pi$.

**Information spaces** Items 10 and 11 characterize the history that is available for decision making. The relationship between sensor and action history and decision making has long been considered important in planning under uncertainty (e.g., [12], [22], [30]). Generally one would like to optimize the performance of a robot, while directly taking into account the complications due to limited sensing. By using the concept of information state, as considered in stochastic control and dynamic game theory, a useful characterization of this relationship is provided. When there is perfect state information, decisions can be made on the basis of state. However, with imperfect state information, the decisions are conditioned on information states. The information state concept is similar to the definition of knowledge states, considered in [10], and has also recently been proposed in [2].

In Item 10, the dimension of the information space can increase linearly with the number of stages; however, alternative representations are possible, and preferable in many cases. In the case of nondeterministic uncertainty, the information space can be alternatively represented as an algebra of subsets of $X$ that are obtained by performing set intersections that maintain consistency with the history. With probabilistic uncertainty, the information space can be alternatively represented as a pdf on $X$ that is obtained through the repeated application of Bayes' rule. Functional approximation can also be considered to produce a low-dimensional representation of the information space [24].

**The strategy concept** Item 12 defines a strategy for each decision maker. The computational goal is to design a strategy that will lead to the accomplishment of some robotic task. At a given stage, each decision maker conditions its actions on its information state. This represents a deterministic (or pure) strategy; however, a randomized (or mixed) strategy can alternatively be defined. In this case a pdf of the form $p(u_k^i|\eta_k^i)$ is specified as the strategy, and actions are chosen by sampling.

**Encoding preferences** Item 13 defines a loss functional for each of the decision makers, which guides the selection of strategies. The loss can generally be based on actions taken by any decision maker at any stage, and on the state trajectory. In this general form, the loss functionals can also depend on nature.

One form that is often used in discrete-time optimal control theory is the stage-additive loss functional (for a single decision maker):
$L(x_1, \ldots, x_{K+1}, u_1, \ldots, u_K) =$

$$\sum_{k=1}^{K} l_k(x_k, u_k) + l_{K+1}(x_{K+1}), \qquad (9)$$

in which $l_k(x_k, u_k)$ represents a cost that can accumulate (such as time, distance, or energy), and $l_{K+1}(x_{K+1})$ is a final cost that could, for instance, penalize a strategy that does not terminate in a goal region.

The general task is to determine strategies that optimize the losses in some appropriate sense. In the case of a single decision maker without nature, the task is to select a strategy that minimizes $L$. In the case of nondeterministic actions from nature, the task is to select a strategy that minimizes the worst-case loss. In the probabilistic case, one natural choice is to minimize the expected loss. For cases in which there are multiple, independent decision makers, a number of different concepts may be appropriate. For instance, in a cooperative game in which there is a certain amount of trust, *Pareto optimality* may be appropriate [34]. In a noncooperative setting, a Nash equilibrium condition might be appropriate [1]. This corresponds to a game strategy that minimizes the loss of each decision maker, given that the strategies of the other decision makers cannot be changed.

## 3 Synthesizing and Extending Motion Planning Concepts

By following this game-theoretic perspective, modeling, analysis, algorithms, and computed examples have so far been obtained for three classes of problems: (1)

motion planning under uncertainty in sensing and control [24], [25], [26]; (2) motion planning under environment uncertainties [24], [28]; and (3) multiple-robot motion planning [24], [27]. For the first problem class, a general method for determining feedback strategies is developed by blending ideas from dynamic game theory with traditional preimage planning concepts. This generalizes classical preimages to *performance preimages* and preimage plans to *motion strategies with information feedback*. For the second problem class, robot strategies are analyzed and determined for situations in which the environment is changing, but not completely predictable. For the third problem class, dynamic game-theoretic concepts are applied to motion planning for multiple robots that have independent goals. Several versions of the formulation have been considered: fixed-path coordination, coordination on independent configuration-space roadmaps, and centralized planning.

This section highlights some of the key concepts, to illustrate the the utility of the mathematical representation. Section 4 discusses some selected algorithm issues and presents some illustrative, computed examples.

**Modeling sources of uncertainty** Several types of uncertainty will be discussed for the single-robot case. It is straightforward to extend the discussion to multiple robots. All types are modeled with nature, which can be assumed to be either nondeterministic or probabilistic.

Suppose that $X = \mathcal{C}_{free}$, and let $\mathbf{q}_k$ denote the configuration (or state) at stage $k$. The state transition equation (2) can be specialized to $\mathbf{q}_{k+1} = f_k(\mathbf{q}_k, u_k, \theta_k)$. This represents a generalization of the control model that was given in (5), and represents uncertainty in *configuration predictability*. Suppose further that the observation equation is of the form $y_k = h_k(\mathbf{q}_k, \theta_k^s)$. This represents a generalization of the sensing model that was given in (8), and represents uncertainty in *configuration sensing*.

Other sources of uncertainty can be considered in addition to configuration uncertainties. Suppose, for example, that $\mathcal{C}_{free}$ is not exactly known, but is instead known to be one of several possibilities. In this case there is uncertainty in the robot's environment. A set $E$ can be used to index the alternatives, and a state space is defined as some subset $X \subset \mathcal{C} \times E$ [28]. Thus, for every $e \in E$, a different free configuration space can be obtained. Let $[\mathbf{q}_k \; e_k]$ represent the state at stage $k$. A state transition equation can be defined in two portions. Suppose that the future configurations are obtained deterministically from $\mathbf{q}_{k+1} = f_k'(\mathbf{q}_k, u_k)$, and future environments are obtained from $e_{k+1} = f_k''(e_k, \theta_k^a)$. In this case nature causes uncertainty in *environment predictability*. More generally, the future environments can be conditioned on the robot's configuration (which occurs, for instance in a manipulation problem) and the action, to yield $e_{k+1} = f_k''(x_k, \theta_k^a)$, in which $x_k = [\mathbf{q}_k \; e_k]$.

If the current environment is unknown, then there is uncertainty in *environment sensing*, which is a problem that has been considered from several different perspectives (e.g., [7],[9],[17],[38]). This can be modeled by defining $y_k = h_k(x_k, \theta_k^s)$, in which $x_k = [\mathbf{q}_k \; e_k]$.

In general, sensing and predictability uncertainties can be defined for any state space, including those that include dynamics. Also, a set of parameters could characterize variations in the model, and used to form models of uncertainty in predictability and sensing, in the same way that $E$ was used.

It has been assumed thus far that each decision maker knows all game components, including the loss functionals, of other decision makers. Another sensing model could be introduced that reflects imperfect information that each decision maker has about the game itself. Problems of this type are quite realistic, yet are very difficult to model [14], [16]. The information of each decision maker could be represented, for example, as a pdf over a set of possible games. To make appropriate decisions, each decision maker must speculate about the knowledge that other decision makers have regarding the game. This type of second-guessing can progress for an infinite number of layers, which leads to a formidable modeling task.

**Forward projections** In preimage planning research, the notion of a forward projection has been useful for characterizing robot execution when there is uncertainty in configuration predictability and sensing. This concept can be substantially generalized, and in Section 4 a computed example of a probabilistic forward projection is shown.

The forward projection in this section will character-

ize future states under the implementation of a strategy. Without uncertainties, this corresponds to providing the state trajectory that can be inferred from (2). Suppose that the strategy, $\gamma_k$ is fixed for all $k$, and that there is perfect current-state information available at all times. Under nondeterministic uncertainty, a subset of $X$ in which the system state will lie can be inferred. Consider the state at stage $x_{k+2}$, if $x_k$ is known. From (4), it is already known that $x_{k+1} \in F_{k+1}(x_k, u_k)$, and $u_k = \gamma_k(x_k)$. The nondeterministic action of nature at stage $k+1$ must next be taken into account to yield $F_{k+2}(x_k, \gamma) =$

$$\{f(x_{k+1}, u_{k+1}, \theta^a_{k+1}) \in X | x_{k+1} \in F_{k+1}(x_k, \gamma), \ \ \theta^a_{k+1} \in \Theta^a\}. \quad (10)$$

This defines the forward projection at stage $k+2$ in terms of the projection at stage $k+1$. By induction, forward projections can be constructed to any future stage.

Forward projections can be analogously constructed for probabilistic uncertainty. For instance, the pdf at stage $k+2$ is $p(x_{k+2}|x_k, \gamma) =$

$$\int p(x_{k+2}|x_{k+1}, \gamma_{k+1}(x_{k+1}))p(x_{k+1}|x_k, \gamma_k(x_k))dx_{k+1}. \quad (11)$$

These are the forward projections for the cases of nondeterministic uncertainty and probabilistic uncertainty, with a single robot that has uncertainty only in predictability (i.e., the current state is known). Other forward projections, which include sensing uncertainty and multiple robots, are presented along with additional computed examples in [24].

**Termination conditions** The decision to halt the robot has been given careful attention in manipulation planning research, particularly in cases that involve configuration-sensing uncertainty. A motion plan might bring the robot into a goal region (*reachability*), but the robot may not halt if it does not realize that it is in the goal region (*recognizability*) [12]. The notion of a termination condition has been quite useful for formulating robot plans that tell the robot when to halt, based on its current, partial information [12], [23], [30]. The same concept can be introduced in a game-theoretic formulation by defining a binary-valued mapping (as part of a strategy),

$$TC_k : N_k \rightarrow \{true, false\}, \quad (12)$$

and enforcing the constraint that if $TC_k = true$, then $TC_{k+1} = true$. The *true* condition indicates that the robot should halt, and can be considered as a special action that can be considered by a decision maker (and hence incorporated into a strategy that uses information feedback). This termination condition, in the determination of an optimal strategy, is equivalent to an *optimal stopping rule*, which appears in optimal control theory [20], [24].

**Performance preimages** Recall that a classical preimage yields the set of places in the configuration space from which a goal will be achieved under the application of a fixed motion command. This principle can be significantly generalized within the game-theoretic framework to yield a *performance preimage*.

Assume that a strategy encodes a termination condition in addition to motion control. Suppose that there is nondeterministic uncertainty, which is standard in preimage planning research. Consider some subset of the reals, $\mathcal{R} \subseteq \Re$. The *performance preimage on* $X$ is the subset of $X$ that is given by

$$\check{\pi}_x(\gamma, \mathcal{R}) = \{x_1 \in X | \check{L}(x_1, \gamma) \in \mathcal{R}\}, \quad (13)$$

in which $\check{L}(x_1, \gamma)$ represents the worst-case loss that could be obtained under the implementation of $\gamma$ with an initial state $x_1$. The set $\check{\pi}_x(\gamma, \mathcal{R}) \subseteq X$ indicates places in the state space from which if robot begins, the loss will lie in $\mathcal{R}$.

The state space, $X$, can be partitioned into *isoperformance classes* by defining an equivalence class $\check{\pi}_x(\gamma, \{r\})$ for each $r \in [0, \infty)$. For a 0-1 loss functional (zero if the goal region is achieved), $\check{\pi}_x(\gamma, \{0\})$ yields the classical preimage. With a general loss functional, and $\mathcal{R} = [0, m)$ a performance preimage is obtained that indicates all $x_1 \in X$ from which the goal will be achieved with a loss that is guaranteed to be less than $m$. If a termination condition is neglected, then $\check{\pi}(g, \{0\})$ yields a *backprojection* similar to that in [12].

Suppose probabilistic uncertainty is considered instead of nondeterministic uncertainty. The performance preimage becomes

$$\bar{\pi}_x(\gamma, \mathcal{R}) = \{x_1 \in X | \bar{L}(x_1, \gamma) \in \mathcal{R}\}, \quad (14)$$

in which $\bar{L}(x_1, \gamma)$ represents the *expected* loss that is obtained under the implementation of $\gamma$ from $x_1$. Suppose that $\mathcal{R} = [0, r]$ for some $r \geq 0$. The performance

preimage yields places in $X$ from which the expected performance will be less than or equal to $r$. If $\mathcal{R} = \{r\}$ for some point $r \geq 0$, then places in $X$ are obtained in which equal expected performance will be obtained. With a 0-1 loss functional and ignoring the termination condition, the performance preimages can give isoprobability curves which are equivalent to the probabilistic backprojections in [5].

Performance preimages can also be defined on the information space to account for sensing uncertainty, and for multiple robots [24].

**Decoupling multiple robots** Consider the problem of coordinating multiple robots that have independent goals. Approaches to multiple-robot motion planning are often categorized as *centralized* or *decoupled*. A centralized approach typically constructs a path in a composite configuration space, which is formed by combining the configuration spaces of the individual robots (e.g., [3],[36]). A decoupled approach typically generates paths for each robot independently, and then considers the interactions between the robots (e.g., [11],[19],[33]). The suitability of one approach over the other is usually determined by the tradeoff between computational complexity associated with a given problem and the amount of completeness that is lost.

A variety of multiple-robot coordination problems can be formulated by defining appropriate state spaces [24]. Suppose there are a collection of $N$ robots that share a common workspace and have free spaces $\mathcal{C}^1_{free}, \ldots, \mathcal{C}^N_{free}$. The state space can be defined as the Cartesian product

$$X = \mathcal{C}^1_{free} \times \mathcal{C}^2_{free} \times \cdots \times \mathcal{C}^N_{free}. \tag{15}$$

The subset of $X$ in which two or more robots collide is avoided in a successful motion plan. The dimensionality of this composite space has previously prompted many approaches that decouple the problem. Motion plans are more or less constructed for each robot independently, and then combined to coordinate the robots.

In [25], two additional state space definitions are used. For fixed-path coordination, it is assumed that a collision-free path $\tau^i : [0,1] \rightarrow \mathcal{C}^i_{free}$ is given for each robot, and the state space is defined as the Cartesian product $[0,1]^N$. Instead of a single collision-free path, suppose that each robot is given a network of collision-free paths, referred to as a *roadmap*. Let $\mathcal{R}^i$ denote a space that is formed by combining the domains of the roadmap paths for the $i^{th}$ robot. A roadmap coordination space can be defined as

$$X = \mathcal{R}^1 \times \mathcal{R}^2 \times \cdots \times \mathcal{R}^N. \tag{16}$$

In general, many other combinations of constrained spaces are possible to define the state, leading to a variety of ways to define decoupled planning problems.

**Multiple-robot optimality** Little concern has been given in previous research to optimality for multiple-robot motion planning problems. For a single robot, a scalar loss is optimized. Previous multiple-robot motion planning approaches that consider optimality project the vector of individual losses onto a scalar loss. As a result, these methods can fail to find many potentially useful motion plans.

There are many well-studied optimality concepts from game-theory and multiobjective optimization literature. An optimality concept will be briefly described for the multiple-robot planning problem that results in a small set of alternative strategies that are guaranteed to be less than or equivalent to (in terms of losses) than any other possible strategy.

For each robot, assume there are no uncertainties and define a loss functional of the form
$L^i(x_{init}, x_{goal}, u^1, \ldots, u^N) =$

$$\int_0^T l^i(t, x^i(t), u^i(t))dt + \sum_{j \neq i} c^{ij}(x(\cdot)) + q^i(x^i(T)), \tag{17}$$

which maps to the extended reals, and

$$c^{ij}(x(\cdot)) = \begin{cases} 0 & \text{if } x(t) \in X_{valid} \text{ for all } t \\ \\ \infty & \text{otherwise} \end{cases} \tag{18}$$

and

$$q^i(x^i(T)) = \begin{cases} 0 & \text{if } x^i(T) = x^i_{goal} \\ \infty & \text{otherwise} \end{cases} . \tag{19}$$

The variables $x_{init}$ and $x_{goal}$ represent the initial and goal configurations for all of the robots.

The integrand $l^i$ represents a continuous cost function, which is a standard form that is used in optimal

control theory. It is additionally required, however, that

$$l^i(t, x^i(t), u^i(t)) = 0 \quad \text{if } x^i(t) = x^i_{goal}. \tag{20}$$

This implies that no additional cost is received while the $i^{th}$ robot "waits" at $x^i_{goal}$ until time $T$. The term (18) penalizes collisions between the robots. The subset $X_{valid} \subset X$ represents the (closed) set of all states at which no robots or obstacles are in collision. This has the effect of preventing any robots from considering game strategies that lead to collision. The term (19) represents the goal in terms of performance. If a robot, $\mathcal{A}^i$, fails to achieve its goal $x^i_{goal}$, then it receives infinite loss.

Suppose that the initial state is given. For each game strategy, $\gamma$, a vector of losses will be obtained. A partial ordering, $\preceq$, can be defined on the space of game strategies, $\Gamma$. For a pair of elements $\gamma, \gamma' \in \Gamma$ let $\gamma \preceq \gamma'$ if $L^i(\gamma) \leq L^i(\gamma')$ for every $i$. The minimal game strategies with respect to $\preceq$ are better than or equal to all other game strategies in $\Gamma$, and it is shown in [24] that very few minimal game strategies typically exist (ignoring those that produce equivalent losses).

These solutions can be generated using algorithms that are based on the dynamic programming principle. For the criterion (17) it is shown that minimal solutions are consistent with other well-established forms of optimality from optimization literature [24]. The minimal game strategies are equivalent to the *nondominated* strategies used in multiobjective optimization and *Pareto optimal* game strategies used in cooperative game theory. Furthermore, it can be shown that the minimal game strategies satisfy the Nash equilibrium condition from noncooperative game theory, which implies that for a game strategy $\gamma^* = \{\gamma^{1*} \ldots \gamma^{N*}\}$, the following holds for each $i$ and each $\gamma^i \in \Gamma^i$:

$$L^i(\gamma^{1*}, \ldots, \gamma^{i*}, \ldots \gamma^{N*}) \leq L^i(\gamma^{1*}, \ldots, \gamma^i, \ldots \gamma^{N*}). \tag{21}$$

**Moving obstacles and other nonstationary systems** It has been assumed so far in this section that the system is not time-varying. From a control perspective, this corresponds to a *stationary* problem. Optimal solutions to problems of this type depend only on state (or the information state with sensing uncertainty) and not on time.

By allowing time-varying models, many interesting motion planning problems can be defined. Suppose, for instance, that several moving obstacles exist in the workspace. For a single-robot problem, this leads to a time-varying free configuration space $\mathcal{C}_{free}(t)$ [22], which can be approximated in discrete time as

$$\mathcal{C}_{free}[k] = \bigcap_{t \in [(k-1)\Delta t, k\Delta t)} \mathcal{C}_{free}(t). \tag{22}$$

In general, many game items from Section 2 can encode time-dependent models.

## 4 Algorithms and Computed Examples

This section briefly discusses one of several algorithms that have been developed using this game-theoretic foundation. One purpose is to describe an approach that was inspired by numerical optimal control research, and was straightforward to develop, given the mathematical framework. This section also presents a variety of computed examples that were obtained using various algorithms, to indicate the broad applicability of the concepts. A more thorough presentation of algorithms and computed examples appears in [24].

**An algorithm that handles uncertainty in prediction** Suppose that there is one robot with probabilistic uncertainty in predictability, perfect configuration sensing (i.e., the information space reduces to $X$), and the models are not time-varying. The task is to compute a strategy that is optimal in the expected sense, which is a challenging extension of the basic motion planning problem. The expected loss obtained by starting from stage $k$ and implementing the portion of the optimal strategy $\{\gamma^*_k, \ldots, \gamma^*_K\}$ can be represented as

$$\bar{L}^*_k(x_k) = E\left\{\sum_{i=k}^{K} l_i(x_i, \gamma^*_i(x_i)) + l_{K+1}(x_{K+1})\right\}, \tag{23}$$

in which $E\{\}$ denotes expectation taken over the actions of nature.

The principle of optimality [20] states that $\bar{L}^*_k(x_k)$ can be obtained from $\bar{L}^*_{k+1}(x_{k+1})$ by the following recurrence:

$$\bar{L}^*_k(x_k) = \min_{\gamma_k \in \Gamma_k} \left\{ l_k(x_k, u_k) + \int \bar{L}^*_{k+1}(x_{k+1}) p(x_{k+1}|x_k, u_k) dx_{k+1} \right\}. \tag{24}$$

Note that the integral is taken over states that can be reached using the state transition equation.

An optimal strategy is determined by successively building approximate representations of $\bar{L}_k^*$. Each dynamic programming iteration can be considered as the construction of an approximate representation of $\bar{L}_k^*$. A discretized representation is used to construct a good approximation of the continuous function $\bar{L}_k^*$ over the entire state space. The value for $\bar{L}_k^*(x_k)$ is obtained by computing the right side of (24) for various values of $u_k$ and using linear interpolation. Other schemes, such as quadratic interpolation, can be used to improve numerical accuracy [21].

Note that $\bar{L}_K^*$ represents the cost of the optimal one-stage strategy from each state $x_K$. More generally, $\bar{L}_{K-i}^*$ represents the cost of the optimal $(i+1)$-stage strategy from each state $x_{K-i}$. For a motion planning problem, one is typically concerned only with strategies that require a finite number of stages before terminating in the goal region. For a positive $\delta \approx 0$ the dynamic programming iterations are terminated when $|\bar{L}_k^*(x_k) - \bar{L}_{k+1}^*(x_{k+1})| < \delta$ for all values in the state space. The resulting strategy is formed from the optimal actions and termination conditions in the final iteration. Note that no choice of $K$ is necessary. Also, at each iteration of the dynamic programming algorithm, only the representation of $\bar{L}_{k+1}^*$ is retained while constructing $\bar{L}_k^*$; earlier representations can be discarded.

To execute a strategy, the robot uses the final cost-to-go representation (which is called $\bar{L}_1^*$) in a way similar to the use of a navigation function [4], [35]. The optimal action can be obtained from any real-valued location $x \in X$ though the use of (24) (or the appropriate dynamic programming equation), interpolation, and the approximate representation of $\bar{L}_1^*$. A real-valued initial state is given. Thus, the robot is not confined to move along the quantization grid that is used for determining the cost-to-go functions. The application of the optimal action will yield a new real-valued configuration for the robot. This form of iteration continues until the termination condition is met.

Let $Q$ denote the number of cells per dimension in the representation of $\mathcal{C}_{free}$. Let $n$ denote the dimension of the state space. Let $|U|$ denote the number of actions that are considered. Let $|\Theta|$ denote the number of actions that are considered by nature. The space complexity of the algorithm is $O(Q^n)$. For each iteration of the dynamic programming, the time complexity is $O(Q^n|U|\ |\Theta|)$, and the number of iterations is proportional to the number of stages required for sample paths to reach the goal. The complexity is exponential in dimension, but efficient for fixed dimension. Execution times vary dramatically depending on the resolutions, but computation times typically range from a minute or two for a basic 2D problem up to several hours for a challenging 3D problem, on a typical workstation with little regard to code optimization. It is important to note, however, that this algorithm is not competing with known algorithms that solve the basic problem, since the algorithm described in this paper overcomes uncertainty in prediction and yields an optimal strategy.

Several variations of this algorithm, and other algorithms that apply to problems that were discussed in Section 3, are presented in [24].

**Computed examples** To indicate the broad applicability of the game-theoretic concepts, a variety of computed examples are presented. The scope of the mathematical framework is not limited to problems shown in this section; however, the examples were computed using algorithms that were developed by utilizing the game-theoretic framework [24].

Figures 1 and 2 show some computed results for problems that involve uncertainty in control (cases that additionally involve uncertainty in sensing are presented in [24]). The state transition equation and sensing models are the same as the examples given in Section 2.

Figures 1(a)-(c) show computed preimages for a classic peg-in-hole task with a fixed, downward motion command (i.e., $u_k$ represents a fixed angle in (5) ). Figure 1(a) shows a performance preimage (using a variation of the previously discussed computation technique) under nondeterministic uncertainty and a loss functional that returns 0 when the goal is achieved, and 1 otherwise. The curve shown in Figure 1(a) corresponds closely to the classical preimage that has been determined for this problem in previous manipulation planning research (e.g., [12], [22]). Figure 1(b) assumes probabilistic uncertainty, and shows probabilistic backprojections that are quite similar to those that appear in [5]. Figure 1(c) shows performance preimages for a case in which a Gaussian error model is used to

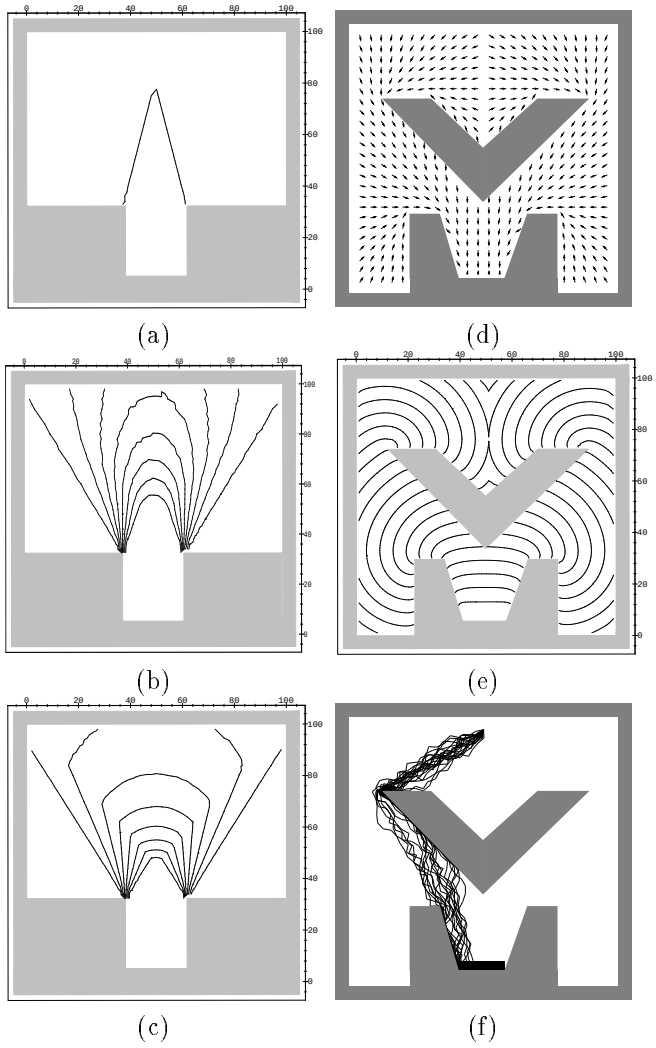


**Figure 1:** *Several computed performance preimages for the classic peg-in-hole problem: (a) a classical preimage; (b) a single-stage probabilistic preimage for a uniform state transition pdf; (c) a single-stage probabilistic preimage for a truncated Gaussian state transition pdf; and a computed optimal strategy for a different problem: (d) the state-feedback solution; (e) performance preimages; (f) simulated executions of the optimal strategy.*

represent the uncertainty in control, as opposed to a bounded uniform pdf as in Figure 1(b) and in [5].

Figure 1 shows a computed optimal strategy for a problem that involves probabilistic uncertainty in control and a loss functional that measures the time to achieve the goal. The goal is at the lower central part of the workspace. Figure 1(d) depicts the optimal strategy by showing the direction of the motion command $u_k = \gamma_k^*(x_k)$ at different locations in the state space. Figure 1(e) shows performance preimages under the implementation of the optimal strategy. Figure 1(f) shows 30 superimposed, simulated executions of the computed optimal strategy.

Figure 2 shows several stages of a computed forward projection under probabilistic uncertainty for a peg-in-hole problem and a fixed motion command. Initially, there is little uncertainty in configuration; however, as time progresses, the pdf on the state space becomes diffuse. Due to uncertainty reduction through compliance (which has been used in preimage planning research [32]), the pdf becomes flattened in the final stages. This type of simulation can provide useful information for experimenting with different uncertainty models and computed strategies.

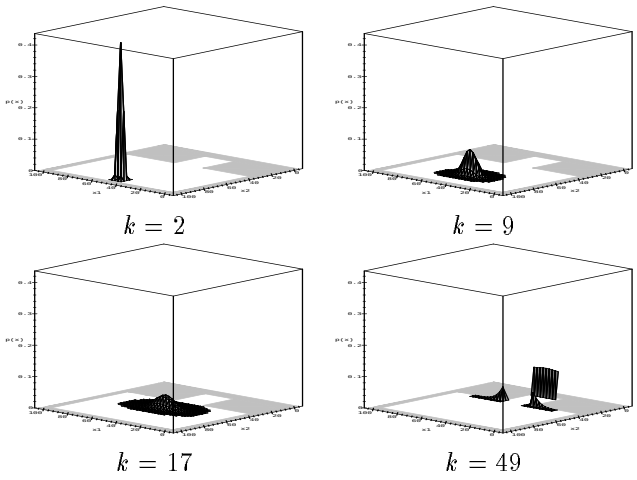


**Figure 2:** *The forward projection at several stages, with probabilistic uncertainty.*

Figures 3 and 4 show computed examples for problems that involve an environment that changes over time and is not completely predictable (more details appear in [28]). Figure 3(a) shows a problem for which there is a single rigid robot that can rotate in place or translate along its major axis. There are two doors that can become open or closed at various points in the future, and the behavior of the doors is modeled

with a Markov process. The state space for this problem is the Cartesian product of the configuration space of the robot and a set of four possible combinations of open and closed doors. Figures 3(b) and (c) show two simulated executions under the implementation of a computed strategy that minimizes the expected time to reach the goal. Different trajectories are taken in different executions because the openings and closings of doors vary; however, both behaviors are obtained from the same strategy. Figure 3(d) shows a problem in which there is a nonholonomic car robot that is capable of only moving in a forward direction and has a limited turning radius. There are two regions in the workspace that are designated as service areas. In this case, the robot interacts with the environment by processing service requests that can occur at various points in the future (again modeled with a Markov process). Figures 3(e) and (f) show two simulated executions under the implementation of the strategy that minimizes the expected time to reach the goal region while there are no outstanding requests. Figure 4 shows a problem that involves a three degree-of-freedom manipulator that delivers one of two possible parts from one of two sources to one of two destinations. The execution of a strategy that minimizes the expected time that parts wait to be processed is depicted, assuming Markov models for resuests of parts, sources, and destinations. Other problems of this type are studied in [37].

Figures 5 and 6 show computed examples for problems that involve multiple robots. Each robot is constrained to move with bounded velocity along an independent roadmap, and a minimal strategy is depicted. These strategies were computed by applying the principle of optimality to a partially ordered space of strategies [24].

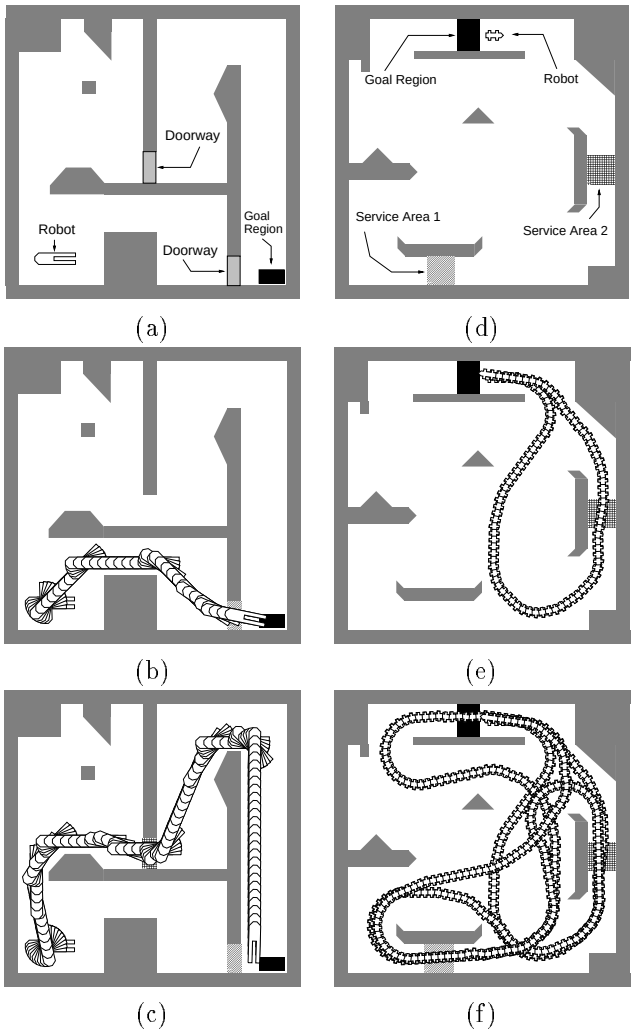


**Figure 3:** *Two examples that illustrate planning in a changing, partially-predictable environment.*

## 5 Conclusion

A dynamic game-theoretic framework has been proposed in this paper to serve as a mathematical foundation for a broad class of motion planning problems. Results obtained by following this perspective were summarized with the intent of indicating the general utility of this foundation. By no means is it intended to provide a general solution to a broad class of problems, but instead it provides a useful characterization upon which motion planning algorithms can be developed. In this way, it can serve the same purpose that configuration space concepts served for basic path planning problems.

This foundation can provide several key advantages for future research: (1) A common, unified structure facilitates the comparison of techniques. Just as configuration space concepts provided a precise, ideal formulation of basic path planning, the dynamic game-theoretic concepts provide a formulation of the ideal (or optimal) strategies that can be achieved. For many difficult problems, tradeoffs are inevitably made to im-

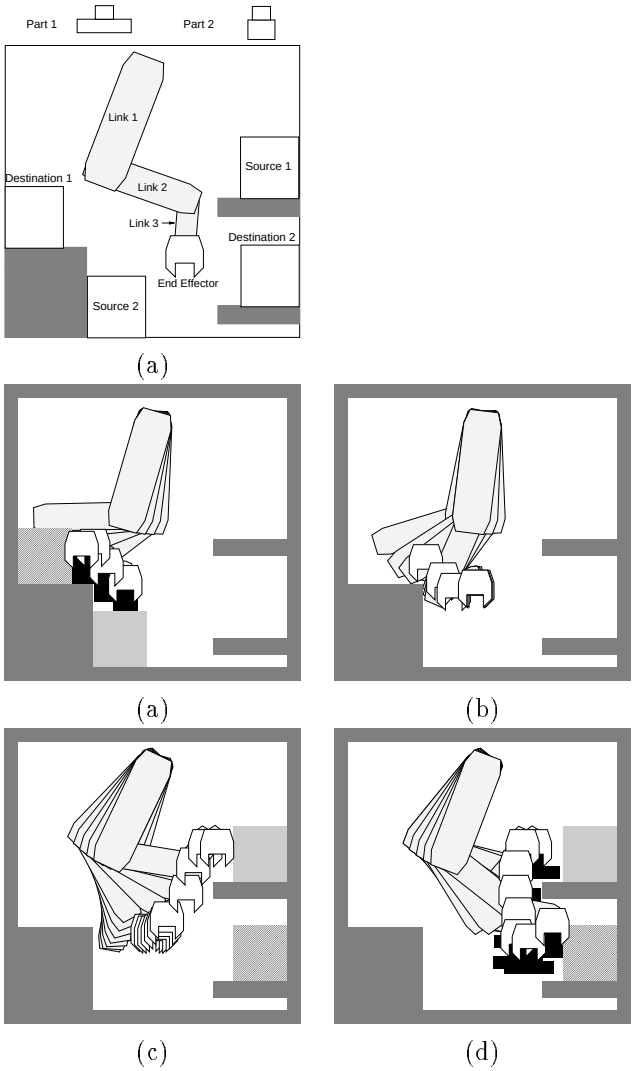


**Figure 4:** *A three degree-of-freedom manipulator with a constrained, rotating end-effector is in a workspace in which there are two parts, two sources, and two destinations that are modeled with a Markov process. The optimal strategy is shown.*

prove computational performance. As approximate or incomplete methods are proposed, it is useful for the purposes of analysis to have precise, ideal formulations. (2) Clear directions are provided along which the concepts and methods can be generalized. For example, the preimage and forward projection concepts have been shown to apply in very general settings by generalizing their definitions within the framework. This has provided a clear relationship between nondeterministic and probabilistic uncertainty models, and numerical navigation functions and preimages. (3) A variety of different models can be incrementally tested. One of the greatest difficulties in motion planning under uncertainties is determining appropriate models of uncertainty, while previous algorithms have often applied to very specific uncertainty models. The framework allows the substitutions of a variety of different models while many of the principles remain unchanged. This is particularly true of the numerical computation method briefly discussed in Section 4, which makes few restrictions on the models.

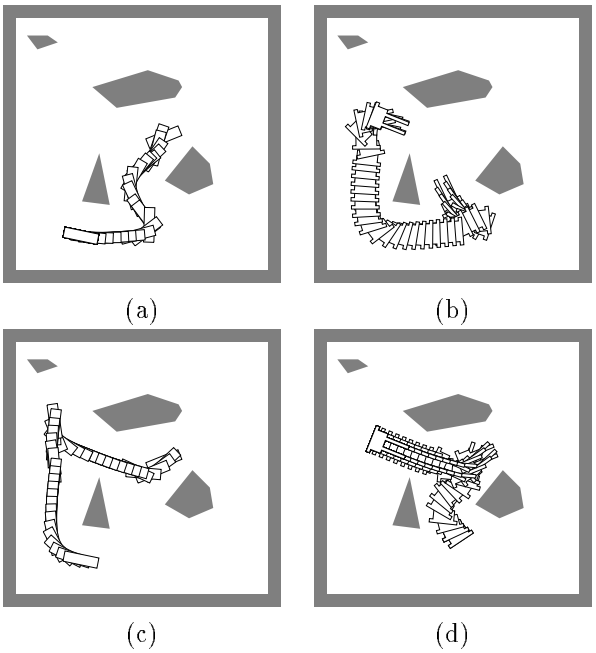


**Figure 5:** *Two strategies for a two-robot, roadmap-coordination problem.*

Although the framework has only been applied so far to three classes of motion planning problems, one important direction for future research will be to characterize and analyze additional problems. For example, problems that involve dynamics, sensing from a vision system, or complex manipulations, have yet to be considered. For many problems, specialized representations will undoubtedly be useful for developing algorithms. In many cases, useful concepts from the ex-

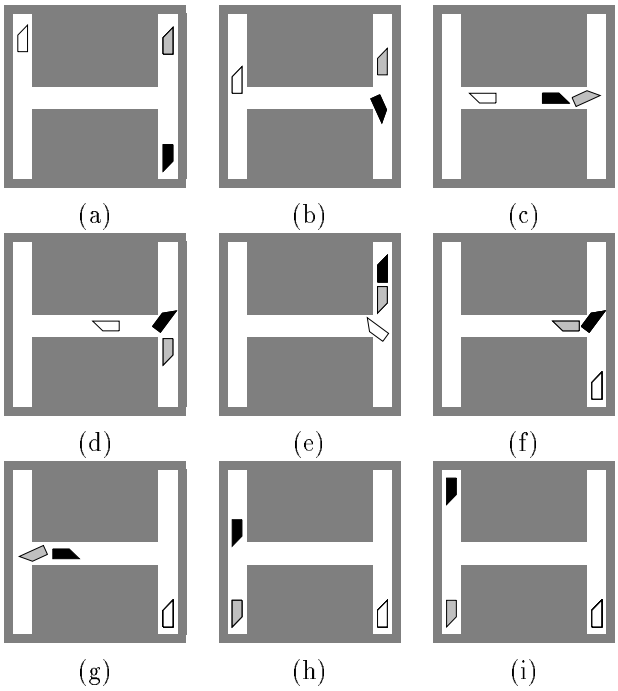


**Figure 6:** *A minimal solution strategy for three rotating robots on independent roadmaps.*

isting literature can be combined with the mathematical structure, such as in the case of using preimage planning research to develop the performance preimage. Such constructions are useful for developing algorithms, and are compatible with the dynamic game-theoretic concepts.

## Acknowledgments

I thank Narendra Ahuja, Tamer Başar, Bruce Donald, Mike Erdmann, Ken Goldberg, Dan Koditschek, Jean-Claude Latombe, Jean Ponce, and Mark Spong, for their helpful comments and suggestions regarding this research. James Kuffner made helpful suggestions regarding this manuscript. I especially thank Seth Hutchinson and Rajeev Sharma who made significant contributions to the individual applications. This work was sponsored at the University of Illinois by a Beckman Institute research assistantship, and a Mavis Fellowship. Jean-Claude Latombe's research lab at Stanford University provided additional support.

## References

[1] T. Başar and G. J. Olsder. *Dynamic Noncooperative Game Theory.* Academic Press, London, 1982.

[2] J. Barraquand and P. Ferbach. Motion planning with uncertainty: The information space approach. In *IEEE Int. Conf. Robot. & Autom.*, pages 1341–1348, 1995.

[3] J. Barraquand, B. Langlois, and J. C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Trans. Syst., Man, Cybern.*, 22(2):224–241, 1992.

[4] J. Barraquand and J.-C. Latombe. A Monte-Carlo algorithm for path planning with many degrees of freedom. In *IEEE Int. Conf. Robot. & Autom.*, pages 1712–1717, 1990.

[5] R. C. Brost and A. D. Christiansen. Probabilistic analysis of manipulation tasks: A computational framework. *Int. J. Robot. Res.*, 15(1):1–23, February 1996.

[6] J. F. Canny. On computability of fine motion plans. In *IEEE Int. Conf. Robot. & Autom.*, pages 177–182, 1989.

[7] B. R. Donald. *Error Detection and Recovery for Robot Motion Planning with Uncertainty.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1987.

[8] B. R. Donald and J. Jennings. Sensor interpretation and task-directed planning using perceptual equivalence classes. In *IEEE Int. Conf. Robot. & Autom.*, pages 190–197, Sacramento, CA, April 1991.

[9] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, 22(6):46–57, June 1989.

[10] M. Erdmann. Randomization for robot tasks: Using dynamic programming in the space of knowledge states. *Algorithmica*, 10:248–291, 1993.

[11] M. Erdmann and T. Lozano-Perez. On multiple moving objects. In *IEEE Int. Conf. Robot. & Autom.*, pages 1419–1424, 1986.

[12] M. A. Erdmann. On motion planning with uncertainty. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, August 1984.

[13] E. G. Gilbert and D. W. Johnson. Distance functions and their application to robot path planning in the presence of obstacles. *IEEE Trans. Robot. & Autom.*, 1(1):21–30, March 1985.

[14] P. J. Gmytrasiewicz, E. H. Durfee, and D. K. Wehe. A decision-theoretic approach to coordinating multi-agent interations. In *Proc. Int. Joint Conf. on Artif. Intell.*, pages 62–68, 1991.

[15] K. Y. Goldberg. *Stochastic Plans for Robotic Manipulation.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, August 1990.

[16] J. C. Harsanyi. Games with incomplete information played by Bayesian players. *Management Science*, 14(3):159–182, November 1967.

[17] H. Hu and M. Brady. A Bayesian approach to real-time obstacle avoidance for a mobile robot. *Autonomous Robots*, 1(1):69–92, 1994.

[18] R. Isaacs. *Differential Games.* Wiley, New York, NY, 1965.

[19] K. Kant and S. W. Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *Int. J. Robot. Res.*, 5(3):72–89, 1986.

[20] P. R. Kumar and P. Varaiya. *Stochastic Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1986.

[21] R. E. Larson and J. L. Casti. *Principles of Dynamic Programming, Part II.* Dekker, New York, NY, 1982.

[22] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, MA, 1991.

[23] J.-C. Latombe, A. Lazanas, and S. Shekhar. Robot motion planning with uncertainty in control and sensing. *Artif. Intell.*, 52:1–47, 1991.

[24] S. M. LaValle. *A Game-Theoretic Framework for Robot Motion Planning.* PhD thesis, University of Illinois, Urbana, IL, July 1995.

[25] S. M. LaValle and S. A. Hutchinson. An objective-based stochastic framework for manipulation planning. In *Proc. IEEE/RSJ/GI Int'l Conf. on Intelligent Robots and Systems*, pages 1772–1779, September 1994.

[26] S. M. LaValle and S. A. Hutchinson. Evaluating motion strategies under nondeterministic or probabilistic uncertainties in sensing and control. In *Proc. IEEE Int'l Conf. Robot. & and Autom.*, pages 3034–3039, April 1996.

[27] S. M. LaValle and S. A. Hutchinson. Optimal motion planning for multiple robots having independent goals. In *Proc. IEEE Int'l Conf. Robot. & and Autom.*, pages 2847–2852, April 1996.

[28] S. M. LaValle and R. Sharma. Motion planning in stochastic environments: Applications and computational issues. In *IEEE Int'l Conf. on Robotics and Automation*, pages 3063–3068, 1995.

[29] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Trans. on Comput.*, C-32(2):108–120, 1983.

[30] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor. Automatic systhesis of fine-motion strategies for robots. *Int. J. Robot. Res.*, 3(1):3–24, 1984.

[31] K. M. Lynch and M. T. Mason. Pulling by pushing, slip with infinite friction, and perfectly rough surfaces. *Int. J. Robot. Res.*, 14(2):174–183, 1995.

[32] M. T. Mason. Compliance and force control for computer controlled manipulators. In B. Brady *et al.*, editor, *Robot Motion: Planning and Control*, pages 373–404. MIT Press, Cambridge, MA, 1982.

[33] P. A. O'Donnell and T. Lozano-Pérez. Deadlock-free and collision-free coordination of two robot manipulators. In *IEEE Int. Conf. Robot. & Autom.*, pages 484–489, 1989.

[34] G. Owen. *Game Theory.* Academic Press, New York, NY, 1982.

[35] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential fields. *IEEE Trans. Robot. & Autom.*, 8(5):501–518, October 1992.

[36] J. T. Schwartz and M. Sharir. On the piano movers' problem: III. Coordinating the motion of several independent bodies. *Int. J. Robot. Res.*, 2(3):97–140, 1983.

[37] R. Sharma, S. M. LaValle, and S. A. Hutchinson. Optimizing robot motion strategies for assembly with stochastic models of the assembly process. *IEEE Trans. on Robotics and Automation*, 12(2):160–174, April 1996.

[38] A. Stentz. Optimal and efficient path planning for partially-known environments. In *IEEE Int. Conf. Robot. & Autom.*, pages 3310–3317, 1994.