51 ROBOTICS

Dan Halperin, Lydia E. Kavraki, and Kiril Solovey

INTRODUCTION

Robotics is concerned with the generation of computer-controlled motions of physical objects in a wide variety of settings. Because physical objects define spatial distributions in 3-space, geometric representations and computations play an important role in robotics. As a result the field is a significant source of practical problems for computational geometry. There are substantial differences, however, in the ways researchers in robotics and in computational geometry address related problems. Robotics researchers are primarily interested in developing methods that work well in practice and can be combined into integrated systems. They often pay less attention than researchers in computational geometry to the underlying combinatorial and complexity issues (the focus of Chapter 50). This difference in approach will become clear in the present chapter.

We start this chapter with *part manipulation* (Section 51.1), which is a frequently-performed operation in industrial robotics concerned with the physical rearrangement of multiple parts, in preparation for further automated steps. Next, we consider the problem of *assembly sequencing* (Section 51.2), where multiple parts need to be assembled into a target shape. A significant part of this chapter is devoted to *motion planning* (Section 51.3), which is concerned with finding a collision-free path for a robot operating in an environment cluttered with obstacles. We put special emphasis on *sampling-based* techniques for dealing with the problem. We then review various extensions of this problem in Section 51.4. The three final and brief sections are dedicated to software for the implementation of sampling-based motion-planning algorithms, applications of motion planning beyond robotics and bibliographic sources.

GLOSSARY

Workspace W: A subset of 2D or 3D physical space: $W \subset \mathbb{R}^k$ ($k = 2$ or 3).

Body: Rigid physical object modeled as a compact manifold with boundary $B \subset \mathbb{R}^k$ ($k = 2$ or 3). B's boundary is assumed piecewise-smooth. We will use the terms "body," "physical object," and "part" interchangeably.

Robot: A collection of bodies capable of generating their own motions.

Configuration: Any mathematical specification of the position and orientation of every body composing a robot, relative to a fixed coordinate system. The configuration of a single body is also called a ***placement*** or a ***pose***.

Configuration space $\mathcal{C}$: Set of all configurations of a robot. For almost any robot, this set is a smooth manifold. We will always denote the configuration space of a robot by $\mathcal{C}$ and its dimension by m. Given a robot A, we will let $A(\boldsymbol{q})$ denote the subset of the workspace occupied by A at configuration $\boldsymbol{q}$.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

Number of degrees of freedom: The dimension m of $\mathcal{C}$. In the following we will abbreviate "degree of freedom" by ***dof***.

51.1 PART MANIPULATION

Part manipulation is one of the most frequently performed operations in industrial robotics: parts are grasped from conveyor belts, they are oriented prior to feeding assembly workcells, and they are immobilized for machining operations.

GLOSSARY

Wrench: A pair $[\boldsymbol{f}, \boldsymbol{p} \times \boldsymbol{f}]$, where $\boldsymbol{p}$ denotes a point in the boundary of a body B, represented by its coordinate vector in a frame attached to B, $\boldsymbol{f}$ designates a force applied to B at $\boldsymbol{p}$, and $\times$ is the vector cross-product. If $\boldsymbol{f}$ is a unit vector, the wrench is said to be a ***unit*** wrench.

Finger: A tool that can apply a wrench.

Grasp: A set of unit wrenches $\boldsymbol{w}_i = [\boldsymbol{f}_i, \boldsymbol{p}_i \times \boldsymbol{f}_i]$, $i = 1, \ldots, p$, defined on a body B, each created by a finger in contact with the boundary, ∂B, of B. For each $\boldsymbol{w}_i$, if the contact is frictionless, $\boldsymbol{f}_i$ is normal to ∂B at $\boldsymbol{p}_i$; otherwise, it can span the friction cone defined by the Coulomb law.

Force-closure grasp: A grasp $\{\boldsymbol{w}_i\}_{i=1,\ldots,p}$ such that, for any arbitrary wrench $\boldsymbol{w}$, there exists a set of real values $\{f_1, \ldots, f_p\}$ achieving $\Sigma_{i=1}^{p} f_i \boldsymbol{w}_i = -\boldsymbol{w}$. In other words, a force-closure grasp can resist any external wrenches applied to B. If contacts are nonsticky, we require that $f_i \geq 0$ for all $i = 1, \ldots, p$, and the grasp is called ***positive***. In this section we only consider positive grasps.

Form-closure grasp: A positive force-closure grasp in which all finger-body contacts are frictionless.

51.1.1 GRASPING

Grasp analysis and synthesis has been an active research area over the last decade and has contributed to the development of robotic hands and grasping mechanisms. A comprehensive review of robot grasping foundations can be found at [LMS14, Chapter 2].

SIZE OF A FORM/FORCE CLOSURE GRASP

The following results are shown in [MLP90, MSS87]:

- Bodies with rotational symmetry (e.g., disks in 2-space, spheres and cylinders in 3-space) admit no form-closure grasps.
- All other bodies admit a form-closure grasp with at most four fingers in 2-space and twelve fingers in 3-space.
- All polyhedral bodies have a form-closure grasp with seven fingers.
- With frictional finger-body contacts, all bodies admit a force-closure grasp that consists of three fingers in 2-space and four fingers in 3-space.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

TESTING FOR FORM/FORCE CLOSURE

A necessary and sufficient condition for force closure in 2-space (resp. 3-space) is that the finger wrenches span three (resp. six) dimensions and that a strictly positive linear combination of them be zero. Said otherwise, the null wrench (the origin) should lie in the interior of the convex hull H of the finger wrenches [MSS87]. This condition provides an effective test for deciding in constant time whether a given grasp achieves force closure. A related quantitative measure of the quality of a grasp (one among several metrics proposed) is the radius of the maximum ball centered at the origin and contained in the convex hull H [KMY92].

SYNTHESIZING FORM/FORCE CLOSURE GRASPS

Most research has concentrated on computing grasps with two to four nonsticky fingers. Optimization techniques and elementary Euclidean geometry are used in [MLP90] to derive an algorithm computing a single force-closure grasp of a polygonal or polyhedral part. This algorithm is linear in the part complexity. Other linear-time techniques using results from combinatorial geometry (Steinitz's theorem) are presented in [MSS87, Mis95]. Optimal force-closure grasps are synthesized in [FC92] by maximizing the set of external wrenches that can be balanced by the contact wrenches.

Finding the maximal regions on a body where fingers can be positioned independently while achieving force closure makes it possible to accommodate errors in finger placement. Geometric algorithms for constructing such regions are proposed in [Ngu88] for grasping polygons with two fingers (with friction) and four fingers (without friction), and for grasping polyhedra with three fingers (with frictional contact capable of generating torques) and seven fingers (without friction). Curved obstacles have also been studied [PSS$^+$97]. The latter paper contains a good overview of work on the effect of curvature at contact points on grasp planning.

DEXTROUS GRASPING

Reorienting a part by moving fingers on the part's surface is often considered to lie in the broader realm of grasping. Finger gait algorithms [Rus99] and nonholonomic rolling contacts [HGL$^+$97] for fingertips have been explored. A more general approach relying on extrinsic resources to the hand is considered in [DRP$^+$14].

51.1.2 CAGING

The problem of *caging* is concerned with surrounding an object by the fingers of a robotic hand such that the object may be able to move locally, but cannot escape the cage. This problem was first introduced in [RB99] with the motivation of simplifying the challenging task of grasping an object by first caging it. The relation between caging and grasping is studied in [RMF12]. This work characterizes caging configurations from which the object can be grasped without breaking the cage.

An important parameter of caging is the number of fingers involved in the construction of the cage—caging of complex objects typically requires more fingers than simple ones, and is also more complex to compute. Two-finger cagings were studied in [RB99, PS06, PS11, ABR12] for a polygonal object, and in [PS11, ARB14] for a polytope; three-finger cagings were the concern of [DB98, ETRP07, VS08, ARB15];

more complex settings are studied in [PVS08, RMF12]. Caging was also used in the context of manipulation (see Section 51.4.2), where several robots cage the object in order to manipulate it [SRP02, FHK08].

51.1.3 FIXTURING

Most manufacturing operations require fixtures to hold parts. To avoid the custom design of fixtures for each part, modular reconfigurable fixtures are often used. A typical modular fixture consists of a workholding surface, usually a plane, that has a lattice of holes where locators, clamps, and edge fixtures can be placed. Locators are simple round pins, while clamps apply pressure on the part.

Contacts between fixture elements and parts are generally assumed to be frictionless. In modular fixturing, contact locations are restricted by the lattice of holes, and form closure cannot always be achieved. In particular, when three locators and one clamp are used on a workholding plane, there exist polygons of arbitrary size for which no fixture design can be achieved [ZG96]. But if parts are restricted to be rectilinear, a fixture can always be found as long as all edges have length at least four lattice units [Mis91]. Algorithms for computing all placements of (frictionless) point fingers that put a polygonal part in form closure and all placements of point fingers that achieve "2nd-order immobility" [RB98] of a polygonal part are presented in [SWO00].

When the fixturing kit consists of a latticed workholding plane, three locators, and one clamp, the algorithm in [BG96] finds all possible placements of a given part on the workholding surface where form closure can be achieved, along with the corresponding positions of the locators and the clamp. The algorithm in [ORSW95] computes the form-closure fixtures of input polygonal parts using a kit containing one edge fixture, one locator, and one clamp.

An algorithm for fixturing an assembly of parts that are not rigidly fastened together is proposed in [Mat95]. A large number of fixturing contacts are first scattered at random on the external boundary of the assembly. Redundant contacts are then pruned until the stability of the assembly is no longer guaranteed.

Fixturing is also studied for more complex parts such as sheet metal with holes [GGB+04], deformable parts that are modeled as linearly-elastic polygons [GG05] and polygonal chains [CSG+07, RS12].

51.1.4 PART FEEDING

Part feeders account for a large fraction of the cost of a robotic assembly workcell. A typical feeder must bring parts at subsecond rates with high reliability. The problem of part-feeder design is formalized in [Nat89] in terms of a set of functions—called ***transfer functions***—which map configurations to configurations. The goal is then to find a composition of these functions that maps each configuration to a unique final configuration (or a small set of final configurations). Given k transfer functions and n possible configurations, the shortest composition that will result in the smallest number of final configurations can be found in $O(kn^4)$ [Nat89]. If the transfer functions are all monotone, the complexity is reduced to $O(kn^2)$ [Epp90].

Part feeding often relies on ***nonprehensile manipulation***. Nonprehensile manipulation exploits task mechanics to achieve a goal state without grasping and

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

frequently allows accomplishing complex feeding tasks with few dofs. It may also enable a robot to move parts that are too large or heavy to be grasped and lifted.

Pushing is one form of nonprehensile manipulation. Work on pushing originated in [Mas82] where a simple rule is established to qualitatively determine the motion of a pushed object. This rule makes use of the position of the center of friction of the object on the supporting surface. Given a part we can compute its ***push*** transfer function. The push function, $p_\alpha : S_1 \rightarrow S_1$, when given an orientation θ returns the orientation of the part $p_\alpha(\theta)$ after it has been pushed from direction α by a fence orthogonal to the push direction. With a sequence of different push operations it is possible to uniquely orient a part. The push function has been used in several nonprehensile manipulation algorithms:

- A planning algorithm for a robot that tilts a tray containing a planar part of known shape to orient it to a desired orientation [EM88]. This algorithm was extended to the polyhedral case in [EMJ93].
- An algorithm to compute the design of a sequence of curved fences along a conveyor belt to reorient a given polygonal part [WGPB96]. See also [BGOS98].
- An algorithm that computes a sequence of motions of a single articulated fence on a conveyor belt that achieves a goal orientation of an object [AHLM00].

A frictionless parallel-jaw gripper was used in [Gol93] to orient polygonal parts. For any part P having an n-sided convex hull, there exists a sequence of $2n-1$ squeezes achieving a single orientation of P (up to symmetries of the convex hull). This sequence is computed in $O(n^2)$ time [CI95]. The result has been generalized to planar parts having a piecewise algebraic convex hull [RG95]. It was shown [SG00] that one could design plans whose length depends on a parameter that describes the part's shape (called *geometric eccentricity* in [SG00]) rather than on the description of the combinatorial complexity of the part. For the parallel-jaw gripper we can define the ***squeeze*** transfer function. In [MGEF02] another transfer function is defined: the ***roll*** function. With this function a part is rolled between the jaws by making one jaw slide in the tangential direction. Using a combination of squeeze and roll primitives a polygonal part can be oriented without changing the orientation of the gripper.

Distributed manipulation systems provide another form of nonprehensile manipulation. These systems induce motions on objects through the application of many external forces. The part-orienting algorithm for the parallel-jaw gripper has been adapted for arrays of microelectromechanical actuators which—due to their tiny size—can generate almost continuous fields [BDM99]. Algorithms that position and orient parts based on identifying a finite number (depending on the number of vertices of the part) of distinct equilibrium configurations were also given in [BDM99, BDH99]. Subsequent work showed that using a carefully selected actuators field, it is possible to position and orient parts in two stable equilibrium configurations [Kav97]. Finally, a long standing conjecture was proved, namely that there exists a field that can uniquely position and orient parts in a single step [BDKL00]. In fact, two different such fields were fully analyzed in [LK01b, SK01]. On the macroscopic scale it was shown that in-plane vibration can be used for closed-loop manipulation of objects using vision systems for feedback [RMC00], that arrays of controllable airjets can manipulate paper [YRB00], and that foot-sized discrete actuator arrays can handle heavier objects under various manipulation strategies [LMC01]. The work [BBDG00] describes a sensorless technique for manipulating objects using vibrations.

Some research is devoted to the design of a *vibratory bowl feeder*, which consists of a bowl filled with parts surrounded by a helical metal track. Vibrations cause the parts to advance along the track, where they encounter a sequence of mechanisms, such as traps [ACH01, BGOS01] or blades [GGS06], which are designed to discard parts in improper orientations. Eventually, only parts that are oriented in the desired position reach to the top of the bowl.

OPEN PROBLEMS

A major open practical problem is to predict feeder throughputs to evaluate alternative feeder designs, given the geometry of the parts to be manipulated. In relation to this problem, simulation algorithms have been proposed to predict the pose of a part dropped on a surface [MZG$^+$96, ME02, PS15, Vár14]. In distributed manipulation, an open problem is to analyze the effect of discrete arrays of actuators on the positioning and orientation of parts [LMC01, LK01b].

51.2 ASSEMBLY SEQUENCING

Most mechanical products consist of multiple parts. The goal of assembly sequencing is to compute both an order in which parts can be assembled and the corresponding required movements of the parts.

GLOSSARY

Assembly: Collection of bodies in some given relative placements.

Subassembly: Subset of the bodies composing an assembly A in their relative positions and orientations in A.

Separated subassemblies: Subassemblies that are arbitrarily far apart from one another.

Hand: A tool that can hold an arbitrary number of bodies in fixed relative placements.

Assembly operation: A motion that merges s pairwise-separated subassemblies ($s \geq 2$) into a new subassembly; each subassembly moves as a single body. No overlapping between bodies is allowed during the operation. The parameter s is called the ***number of hands of the operation***. We call the reverse of an assembly operation ***assembly partitioning***.

Assembly sequence: A total ordering on assembly operations that merge the separated parts composing an assembly into this assembly. The maximum, over all the operations in the sequence, of the number of hands required by an operation is called the ***number of hands of the sequence***.

Monotone assembly sequence: A sequence in which no operation brings a body to an intermediate placement (relative to other bodies), before another operation transfers it to its final placement.

NUMBER OF HANDS IN ASSEMBLY

Every assembly of convex polygons in the plane has a two-handed assembly sequence of translations. In the worst case, s hands are necessary and sufficient for assemblies of s star-shaped polygons/polyhedra [Nat88].

There exists an assembly of six tetrahedra without a two-handed assembly sequence of translations, but with a three-handed sequence of translations. Every assembly of five or fewer convex polyhedra admits a two-handed assembly sequence of translations. There exists an assembly of thirty convex polyhedra that cannot be assembled with two hands [SS94].

COMPLEXITY OF ASSEMBLY SEQUENCING

When arbitrary sequences are allowed, assembly sequencing is PSPACE-hard. The problem remains PSPACE-hard even when the bodies are polygons, each with a constant number of vertices [Nat88].

When only two-handed monotone sequences are permitted, deciding if an assembly A can be partitioned into two subassemblies S and $A \setminus S$ such that they can be separated by an arbitrary motion is NP-complete [WKLL95]. The problem remains NP-complete when both S and $A \setminus S$ are required to be connected and motions are restricted to translations [KK95].

MONOTONE TWO-HANDED ASSEMBLY SEQUENCING

A popular approach to assembly sequencing is disassembly sequencing [HdMS91]. A sequence that separates an assembly into its individual components is first generated and next reversed. Most existing assembly sequencers can only generate two-handed monotone sequences. Such a sequence is computed by partitioning the assembly and, recursively, the resulting subassemblies into two separated subassemblies.

The ***nondirectional blocking graph*** (NDBG) is proposed in [WL94] to represent all the blocking relations in an assembly. It is a subdivision of the space of all allowable motions of separation into a finite number of cells such that within each cell the set of blocking relations between all pairs of parts remains fixed. Within each cell this set is represented in the form of a directed graph, called the directional blocking graph (DBG). The NDBG is the collection of the DBGs over all the cells in the subdivision.

We illustrate this approach for polyhedral assemblies when the allowable motions are infinite translations. The partitioning of an assembly consisting of polyhedral parts into two subassemblies is performed as follows. For an ordered pair of parts P_i, P_j, the 3-vector $\vec{d}$ is a ***blocking direction*** if translating P_i to infinity in direction $\vec{d}$ will cause P_i to collide with P_j. For each ordered pair of parts, the set of blocking directions is constructed on the unit sphere $\mathcal{S}^2$ by drawing the boundary arcs of the union of the blocking directions (each arc is a portion of a great circle). The resulting collection of arcs partitions $\mathcal{S}^2$ into maximal regions such that the blocking relation among the parts is the same for any direction inside such a region.

Next, the blocking graph is computed for one such maximal region. The algorithm then moves to an adjacent region and updates the DBG by the blocking relations that change at the boundary between the regions, and so on. After each time the construction of a DBG is completed, this graph is checked for strong connectivity in time linear in its number of edges. The algorithm stops the first time it encounters a DBG that is not strongly connected and it outputs the two subassemblies of the partitioning. The overall sequencing algorithm continues recursively with the resulting subassemblies. If all the DBGs that are produced during a partitioning step are strongly connected, the algorithm reports that the assembly does not admit a two-handed monotone assembly sequence with infinite translations.

Polynomial-time algorithms are proposed in [WL94] to compute and exploit NDBGs for restricted families of motions. In particular, the case of partitioning a polyhedral assembly by a single translation to infinity is analyzed in detail, and it is shown that partitioning an assembly of m polyhedra with a total of v vertices takes $O(m^2v^4)$ time. Another case studied in [WL94] is where the separating motions are infinitesimal rigid motions. Then partitioning the polyhedral assembly takes $O(mc^5)$ time, where m is the number of pairs of parts in contact and c is the number of independent point-plane contact constraints. (This result is improved in [GHH$^+$98] by using the concept of maximally covered cells; see Section 28.6.) Using these algorithms, every feasible disassembly sequence can be generated in polynomial time.

In [WL94], NDBG's are defined only for simple families of separating motions (infinitesimal rigid motions and infinite translations). An extension, called the ***interference diagram***, is proposed in [WKLL95] for more complex motions. In the worst case, however, this diagram yields a partitioning algorithm that is exponential in the number of surfaces describing the assembly. When each separating motion is restricted to be a short sequence of concatenated translations (for example, a finite translation followed by an infinite translation), rather efficient partitioning algorithms are available [HW96]. A unified and general framework for assembly planning, based on the NDBG, called the *motion space approach* is presented in [HLW00]. On the practical side, an exact and robust implementation of this framework for infinite-translation motions is discussed in [FH13]. We mention the works [CJS08, LCS09], which employ sampling-based techniques (see Section 51.3) in order to produce assembly sequences. The reader is referred to a recent review on the subject of assembly sequencing in [Jim13].

OPEN PROBLEM

1. The complexity of the NDBG grows exponentially with the number of parameters that control the allowable motions, making this approach highly time consuming for assembly sequencing with compound motions. For the case of infinitesimal rigid motion it has been observed that only a (relatively small) subset of the NDBG needs to be constructed [GHH$^+$98]. Are there additional types of motion where similar gains can be made? Are there situations where the full NDBG (or a structure of comparable size) must be constructed?

2. Most efficient algorithms for assembly planning deal with the two-handed case. Devise efficient algorithms for 3-(and higher)handed assembly planning.

51.3 MOTION PLANNING

Motion planning is aimed to provide robots with the capability of deciding automatically which motions to execute in order to achieve goals specified by spatial arrangements of physical objects. It arises in a variety of forms. The simplest form—the *basic path-planning problem*—requires finding a geometric collision-free path for a single robot in a known static workspace. The path is represented by a curve connecting two points in the robot's configuration space [LP83]. This curve must not intersect a forbidden region, the *C-obstacle region*, which is the image of the workspace obstacles. Other motion planning problems require dealing with moving obstacles, multiple robots, movable objects, uncertainty, etc.

In this section we consider the basic motion planning. In the next one we review other motion planning problems. Most of our presentation focuses on practical methods, and in particular, sampling-based algorithms. See Chapter 50 for a more extensive review of theoretical motion planning.

GLOSSARY

Forbidden space: The set of configurations $\mathcal{B} \subset \mathcal{C}$ in which the robot collides with obstacles or violate the mechanical limits of its joints.

Free space: The complement of the forbidden space in region in $\mathcal{C}$, $\mathcal{F} = \mathcal{C} \backslash \mathcal{B}$.

Path: A continuous map $\tau : [0, 1] \to \mathcal{C}$. A path is also *free* if it lies entirely in $\mathcal{F}$.

Motion-planning problem: Compute a free path between two input configurations.

Path planning query: Given two points in configuration space find a free path between them. The term is often used in connection with algorithms that preprocess the configuration space in preparation for many queries.

Complete algorithm: A motion planning algorithm is complete if it is guaranteed to find a free path between two given configurations whenever such a path exists, and report that there is no free path otherwise. Complete algorithms are sometimes referred to as ***exact*** algorithms. There are weaker variants of completeness, for example, ***probabilistic completeness***.

SAMPLING-BASED PLANNERS

The complexity of motion planning for robots with many dofs (more than 4 or 5) has led to the development of sampling-based algorithms that trade off completeness against applicability in practical settings. Such techniques avoid computing an explicit geometric representation of the free space. Instead, in their simplest form, they generate a large set of configuration samples and discard samples that represent forbidden configurations of the robot. Then, an attempt is made to connect nearby configurations, given a metric which is defined over the configuration space. This construction induces a graph structure, termed a *roadmap*, with the property that a path connecting two vertices in the roadmap represents a free path in the configuration space which connects the corresponding configurations. Although such techniques are inherently incomplete, in many cases they can be shown to be *probabilistically complete*—guaranteed to find a solution with high probability if one exists, given a sufficient number of samples.

The implementation of sampling-based techniques relies on two main geometric components: A *collision detector* determines whether a given configuration is free or forbidden. Collision-detection calls typically dominate the running time of sampling-based planners. A *nearest-neighbor search* structure processes a collection of points to efficiently answer queries returning the set of nearest neighbors of a given configuration, or the set of configurations that lie within a certain distance from the given configuration. For further information on these components see Chapter 39 and Chapter 43. We provide references for software implementation of collision detection and nearest-neighbor search in Section 51.5 below.

A prime example of a sampling-based algorithm is the *probabilistic roadmap method* (PRM) [KŠLO96], which proceeds as follows. A collection of randomly-sampled configurations is generated, and the non-free configurations are identified using the collision detector, and consequently discarded. The remaining configurations represent the vertex set of the roadmap. Then, for every sampled configuration a collection of nearby configurations are identified using nearest-neighbor search. Finally, a simple (usually, straight-line) path is generated between pairs of nearby configurations, and the path is checked for collision by dense sampling along the path (or by more costly continuous collision-detection techniques). Paths that are identified as collision free are added as edges to the roadmap. Once a roadmap has been computed, it is used to process an arbitrary number of path-planning queries. The density of sampling can be selectively increased to speed up the roadmap connection.

Other sampling strategies, such as *rapidly-exploring random trees* (RRT) [LK01c] and *expansive-space trees* (EST) [HLM99] assume that the initial and goal configurations are given, and incrementally build a tree structure until these two configurations are connected. In the frequently used RRT, for example, in every step a random configuration is generated and the configuration that is the closest to the sample in the current tree is identified. In the most simplified version of the algorithm, the path between the two configurations is checked for collision. If it is indeed free, the sampled configuration is added as a vertex to the tree, along with the new edge connecting it to the tree. It should also be noted that RRT, EST, and several other sampling-based algorithms have found applications in cases where non-holonomic, dynamic, and other constraints are considered (see [KL16] and Section 51.4.4).

The results reported in [KLMR98, HLM99, LK01c, KKL98, LK04b] provide a probabilistic-completeness analysis of the aforementioned planners.

Some research has focused on designing efficient sampling and connection strategies. For instance, the Gaussian sampling strategy produces a greater density of milestones near the boundary of the free space $\mathcal{F}$, whose connectivity is usually more difficult to capture by a roadmap than wide-open areas of $\mathcal{F}$ [BOS99]. Different methods to create milestones near the boundary of $\mathcal{F}$ were obtained in [YTEA12]. A lazy-evaluation of the roadmap has been suggested in [BK00, SL03, SL02] while visibility has been exploited in [SLN00]. Sampling and connection strategies are reviewed in [SL02].

Standard sampling-based planners are unable to determine that a given problem has no solution. Several works deal with *disconnection proofs*, which are able to determine, in certain settings, that two given configurations reside in two distinct components of the free space [BGHN01, ZKM08, MBH12].

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

There is a large variety of applications beyond robot motion where sampling-based planners are useful. For a survey of few such applications, see Section 51.6.

OPTIMALITY IN SAMPLING-BASED PLANNING

So far we have been concerned with finding *a* solution to a given motion-planning problem. In practical settings one is usually interested in finding a *high-quality* path, in terms of, for example, the path's length, clearance from obstacles, energy consumption, and so on. Initial efforts in this area include applying shortcuts on the returned solution to improve its quality [GO07], merging several solutions to produce an improved one [REH11]. Some works fine-tune existing sampling-based planners by modifying the sampling strategy [LTA03, US03, SLN00], or the connection scheme to new samples [US03, SLN00]. Other approaches include supplementing edges for shortcuts in existing roadmaps [NO04] and random restarts [WB08]. On the negative side, it was shown that existing methods can produce arbitrarily bad solutions [NRH10, KF11].

The breakthrough came in [KF11]. This work introduced PRM* and RRT*, variants of PRM and RRT, and proved that they are *asymptotically optimal*—the solution returned by these algorithms converges to the optimal solution, as the number of samples tends to infinity. This work establishes a bound on the connection radius of PRM, which ensures asymptotic optimality. In particular, the connection radius is a function of the total number of samples n and results in a roadmap of size $O(n \log n)$. The adaptation of RRT to an optimal planner requires to consider additional neighbors for connection and a rewiring scheme, which ensures that the resulting roadmap has a tree structure.

Since then, other asymptotically-optimal planners have been introduced. RRT$^{\#}$ [AT13] extends RRT, but through a different technique than RRT*, and has (empirically) better convergence rates than the latter. *Fast marching trees* (FMT*) [JSCP15] is another single-query planner which was shown to have faster convergence rates than RRT*. An improvement of FMT* is introduced in [SH15a]. A lazy version of PRM* is described in [Hau15]. Finally, we note that by using nearest-neighbor data structures tailored for these algorithms, additional speedup in the running times can be obtained [KSH15].

The guarantee of optimality comes at the price of increased space and time consumption of the planners. As a result, several techniques were developed to lower these attributes, by relaxing the optimality guarantees to *asymptotic near optimality*. This property implies that the cost of the returned solution converges to within a factor of $(1 + \varepsilon)$ of the cost of the optimal solution. Several methods were introduced to reduce the size of a probabilistic roadmaps via *spanners* [DB14, WBC15] or *edge contractions* [SSAH14]. The planner described in [SH14a] allows for continuous interpolation between the fast RRT algorithm and the asymptotically-optimal RRT*. A different approach is taken in [LLB14], where an asymptotically near-optimal algorithm is described, which is also suitable for systems with dynamics.

A recent work [SSH16b] describes a general framework for asymptotic analysis of sampling-based planners. The framework exploits a relation between such planners and standard models of *random geometric graphs*, which have been extensively studied for several decades and many of their aspects are by now well understood.

Most of the analysis for sampling-based algorithms is concerned with *asymptotic* properties, i.e., those that hold with high probability as the number of samples tends to infinity. A recent work [DMB15] provides bounds on the probability of finding a near-optimal solution with PRM* after a finite number of iterations. We also mention that the work on FMT* [JSCP15] studies a similar setting for this algorithm, albeit for a limited case of an obstacle-free workspace.

OTHER ALGORITHMS

Several non-sampling based methods have been proposed for path planning. Some of them work well in practice, but they usually offer no performance guarantee.

Heuristic algorithms often search a regular grid defined over the configuration space and generate a path as a sequence of adjacent grid points [Don87]. Early heuristics employed ***potential fields*** to guide the grid search. A potential field is a function over the free space that (ideally) has a global minimum at the goal configuration. This function may be constructed as the sum of an attractive and a repulsive field [Kha86]. More modern approaches use A*-flavored graph-search techniques that drastically reduce the amount of explored grid vertices, and in some cases can guarantee convergence to an optimal solution (see, e.g., [LF09, ASN$^+$16]). Other approaches combine discrete search and sampling-based planners. For example, SyCloP [PKV10] combines A*-flavored searched over a grid imposed on the workspace, and the result of the search is used to guide sampling-based planners.

One may also construct grids at variable resolution. Hierarchical space decomposition techniques such as octrees and boxtrees have been used for that purpose [BH95]. At any decomposition level, each grid cell is labeled EMPTY, FULL, or MIXED depending on whether it lies entirely in the free space, lies in the C-obstacle region, or overlaps both. Only the MIXED cells are decomposed further, until a search algorithm finds a sequence of adjacent FREE cells connecting the initial and goal configurations. A recent subdivision-based planner [WCY15] reduces the computation time by incorporating inexact predicates for checking the property of cells. This planner accepts a resolution parameter and can determine whether a solution determined by this parameter exists or not.

We note that several optimizing planners that are based on optimization methods have been recently developed (e.g., [Kob12, SDH$^+$14]).

OPEN PROBLEMS

1. Analyze convergence rates for sampling-based algorithms (see [DMB15, JSCP15]).
2. Given a fixed amount of time, can one decide if it is best to use an optimizing sampling-based planner, or a non-optimizing sampling-based planner and then optimization methods to improve the resulting path?

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

51.4 EXTENSIONS OF THE BASIC MOTION PLANNING PROBLEM

There are many useful extensions of the basic motion-planning problem. Several are surveyed in Chapter 50, e.g., shortest paths, time-varying workspaces (moving obstacles), and exploratory motion planning. Below we focus on the following extensions: multi-robot motion planning, manipulation planning, task planning, and planning with differential constraints.

GLOSSARY

Movable object: Body that can be grasped, pushed, pulled, and moved by a robot.

Manipulation planning: Motion planning with movable objects.

Trajectory: Path parameterized by time.

Tangent space: Given a smooth manifold M and a point $p \in M$, the vector space $T_p(M)$ spanned by the tangents at p to all smooth curves passing through p and contained in M. The tangent space has the same dimension as M.

Nonholonomic robot: Robot whose permissible velocities at every configuration $\boldsymbol{q}$ span a subset $\Omega(\boldsymbol{q})$ of the tangent space $T_{\boldsymbol{q}}(\mathcal{C})$ of lower dimension. Ω is called the ***set of controls*** of the robot.

Feasible path: A piecewise differentiable path of a nonholonomic robot whose tangent at every point belongs to the robot's set of controls, i.e., satisfies the nonholonomic velocity constraints.

Locally controllable robot: A nonholonomic robot is locally controllable if for every configuration $\boldsymbol{q}_0$ and any configuration $\boldsymbol{q}_1$ in a neighborhood U of $\boldsymbol{q}_0$, there exists a feasible path connecting $\boldsymbol{q}_0$ to $\boldsymbol{q}_1$ which is entirely contained in U.

Kinodynamic planning: Find a minimal-time trajectory between two given configurations of a robot, given the robot's dynamic equation of motion.

51.4.1 MULTI-ROBOT MOTION PLANNING

In *multi-robot motion planning* several robots operate in a shared workspace. The goal is to move the robots to their assigned target positions, while avoiding collisions with obstacles, as well as with each other. The problem can be viewed as an instance of the single-robot case in which the robot consists of several independent parts (robots). In particular, suppose the problem consists of planning the motion of m robots, and denote by C_i the configuration space of the ith robot. Then it can be restated as planning the motion of a single robot whose configuration space is defined to be $C_1 \times \ldots \times C_m$. This observation allows to apply single-robot tools directly to the multi-robot case. However, such a naive approach overlooks the special characteristics of the problem at hand and tends to be inefficient both in theory and practice. Various techniques were specifically designed for this problem. Complete algorithms are covered in Chapter 50.2.3 ("coordinated motion planning"). Here we focus on heuristics and sampling-based techniques.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

We start with the *decentralized* approach, which asserts that conflicts between robots should be resolved locally and without central coordination. In the *reciprocal* method [SBGM11, BB15], the individual robots take into consideration the current position and the velocity of other robots to compute their future trajectories in order to avoid collisions. The work in [KEK10] takes a game-theoretic approach and develops a reward function that reduces time and resources which are spent on coordination, and maximizes the time between conflicts. Several works [KR12, GGGDC12] develop collision-avoidance heuristics based on the sociology of pedestrian interaction.

Centralized planners produce a global plan, which the individual robots need to execute. Such planners fall into the two categories of *coupled* and *decoupled* planners, which indicate whether the planners account for all the possible robot-robot interactions or only partially do so, respectively. Coupled planners typically operate in the combined configuration space of the problem, and as a result can usually guarantee *probabilistic completeness*. Due to the high dimensionality of the combined configuration space, *decoupled* planners typically solve separate problems for the individual robots and combine the sub-solutions into one for the entire problem [LLS99, BBT02, BO05, BSLM09, BGMK12].

We proceed to describe coupled planners. In [SL02] an evaluation of single-robot sampling-based algorithms is given, when directly applied to the multi-robot case. The works [HH02, SHH15] exploit the geometric properties of the two-robot setting and develop techniques that combine sampling-based methods and geometric tools. The work in [ŠO98] introduces a sampling-based technique, which was exploited in two recent works to solve complex settings of the multi-robot problem involving many more robots than the original method could. The algorithm described in [ŠO98] produces probabilistic roadmaps for the individual robots and combines them into a *composite* roadmap embedded in the joint configuration space. Due to the size of this roadmap it can be explicitly constructed only for settings in which the number of robots is very small. In particular, the number of its vertices and the number of neighbors each vertex has are exponential in the number of robots. In [WC15] a search-based method, termed M*, is developed to traverse this enormous roadmap while representing it implicitly. M* can efficiently solve problems involving multiple robots, as long as the solution requires only mild coordination between the robots. Otherwise, it is forced to consider all the neighbors of explored vertices, which is prohibitively costly. In [SSH16a] a different traversal technique, termed *discrete-RRT* (dRRT) was introduced. It is an adaptation of RRT for the exploration of a geometrically-embedded graphs; it can cope with more tight settings of the problem as it does not require to consider all the neighbors of a given vertex in order to make progress.

In [SH14b] a centralized algorithm is described for a generalized problem termed *k-color multi-robot motion planning*—the robots are partitioned into k groups (colors) such that within each group the robots are interchangeable. Every robot is required to move to one of the target positions that are assigned to its group, such that at the end of the motion each target position is occupied by exactly one robot. This work exploits a connection between the *continuous* multi-robot problem and a discrete variant of it called *pebble motion on graphs* (see, e.g., [KMS84, GH10, KLB13, YL13]), which consists of moving a collection of pebbles from one set of vertices to another while abiding by a certain set of rules. The algorithm reduces the k-color problem into several pebble problems such that the solution to the latter can be transformed into a solution to the continuous problem.

The special case of the k-color problem with $k = 1$ is usually termed *unlabeled* multi-robot motion planning. It was first studied in [KH06], where a sampling-based algorithm was introduced. Very recently, efficient and complete algorithms for the unlabeled problem were introduced [TMMK14, ABHS15, SYZH15]. These algorithms make several simplifying assumptions concerning the separation between start and target positions, without which the problem was shown to be PSPACE-hard [SH15b]. See additional information in Chapter 50.2.3.

51.4.2 MANIPULATION PLANNING

Many robot tasks consist of achieving arrangements of physical objects. Such objects, called movable objects, cannot move autonomously; they must be grasped by a robot. Planning with movable objects is called manipulation planning. The problem is considered to be very challenging as one has to reason in a prohibitively large search space, which encompasses the various positions of the robot and the objects. In contrast with Section 51.1, here we take the viewpoint of the manipulator, rather than that of the manipulated objects.

Different instances of the problem that consist of a single robot and a single movable object with relatively simple geometry can be solved rather efficiently in a complete manner [Wil91, ALS95, BG13]. However, the problem becomes computationally intractable when multiple movable objects come into play [Wil91]. The problem of manipulation planning of multiple objects also relates to the popular *SOKOBAN* game, which was shown to be hard to solve on several occasions (see, e.g., [HD05]). Navigation Among Moveable Obstacles (NAMO) considers the path planning problem where obstacles can be moved out of the way [SK05]. Sampling-based techniques were developed to tackle more general and challenging instances of manipulation planning involving several manipulators and objects [NSO07, NSO08, BSK+08].

With the growing demand for robotic manipulators that can perform tasks in cluttered human environments an additional level of complexity was added. In *rearrangement planning*, a robotic manipulator with complex kinodynamic constraints is required to rearrange a set of objects. In [SSKA07] a technique for *monotone* instances of the problem, in which it is assumed that each object can move at most once, is described. A more recent work [KSD+14] which can cope also with non-monotone instances draws upon a relation between the problem at hand and multi-robot motion planning. The technique described in [KB15] extends previous work for monotone settings to the non-monotone case.

Finally, we mention that recent advances in the area of machine learning have led to the development of highly effective techniques for dealing with various complex manipulation tasks. See, e.g., [LWA15, LLG+15, SEBK15].

51.4.3 TASK PLANNING

Thus far the focus was on tasks with relatively simple specifications for the robot to follow. For instance, in motion planning the robot was required to "move from s to t while avoiding collision with obstacles." Real-world problems usually require to perform much more complex operations that force the robot to make decisions while performing a task. In some cases, a richer task specification can assist in performing a given task. Consider for example a robotic arm that needs to collect several books from a desk and place them on a shelf. A human operator can assist by specifying that the manipulator should first remove objects placed on top of the books before proceeding to the actual books.

Such complex problems, which are often grouped under the umbrella term *task planning*, usually require the combination of tools from motion planning, AI, control, and model checking. The predominant methods for task planning are heuristic search [HN01] and constraint satisfaction [KS92]. We mention that a recent workshop was devoted to the subject[1].

A prevalent approach for task planning in the context of motion planning is to supplement the basic input with a Boolean logical expression, usually in the form of linear temporal logic [Pnu77], which has to be satisfied in order to achieve the goal. The workspace is decomposed into regions and the expression describes the conditions on the order of visitation of regions. For instance, "the robot has to first visit A, then B, and C afterwards; it cannot pass through D, unless B has been reached." Many of the approaches in this area implement a two layer architecture: a discrete planner generates a high-level plan that the robot has to fulfill ("the robot should move from B to D"), whereas a low-level continuous planner needs to plan a path that executes the high-level plan. Earlier works integrated potential-field methods for the low-level plans (see, e.g., [KB08, FGKP09]). More recent efforts rely on sampling-based techniques for the continuous planning (see, e.g., [BKV10, BMKV11, KF12, VB13]). A recent work [LAF$^+$15] explores a setting in which every satisfying assignment is associated with a value and the goal is to find the "best" satisfying plan. Besides using logics, the combination of Satisfiability Modulo Theories (SMT) solvers and sampling-based planners for problems involving tasks that include manipulation is presented in [DKCK16].

Task planning has also earned some attention in the context of multi-robot systems [KDB11, WUB$^+$13, USD$^+$13, IS15] and manipulation [SK11, SFR$^+$14].

51.4.4 MOTION PLANNING WITH CONSTRAINTS

In robotics, there is a need to satisfy both global constraints and local constraints. One could consider the avoidance of obstacles as a global constraint. Often local constraints are modeled with differential equations and are called differential constraints. Some of the first such constraints considered in the robotics community were nonholonomic constraints, which raise very interesting geometric problems. They are discussed in this section. However planning under constraints includes planning with velocity and acceleration considerations (the problem is often called kinodynamic planning) and other variations that are also discussed below.

PLANNING FOR NONHOLONOMIC ROBOTS

The trajectories of a nonholonomic robot are constrained by $p \geq 1$ nonintegrable scalar equality constraints:

$$G(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)) = (G^1(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)), \cdots, G^p(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))) = (0, \ldots, 0),$$

where $\dot{\boldsymbol{q}}(t) \in T_{\boldsymbol{q}(t)}(\mathcal{C})$ designates the velocity vector along the trajectory $\boldsymbol{q}(t)$. At every $\boldsymbol{q}$, the function $G_{\boldsymbol{q}} = G(\boldsymbol{q}, .)$ maps the tangent space $T_{\boldsymbol{q}}(\mathcal{C})$ into $\mathbb{R}^p$. If $G_{\boldsymbol{q}}$

[1] See `http://www.kavrakilab.org/2016-rss-workshop/`

is smooth and its Jacobian has full rank (two conditions that are often satisfied), the constraint $G_{\boldsymbol{q}}(\dot{\boldsymbol{q}}) = (0, \ldots, 0)$ constrains $\dot{\boldsymbol{q}}$ to be in a linear subspace of $T_{\boldsymbol{q}}(\mathcal{C})$ of dimension $m-p$. The nonholonomic robot may also be subject to scalar inequality constraints of the form $H^j(\boldsymbol{q}, \dot{\boldsymbol{q}}) > 0$. The subset of $T_{\boldsymbol{q}}(\mathcal{C})$ that satisfies all the constraints on $\dot{\boldsymbol{q}}$ is called the set $\Omega(\boldsymbol{q})$ of controls at $\boldsymbol{q}$. A feasible path is a piecewise differentiable path whose tangent lies everywhere in the control set.

A car-like robot is a classical example of a nonholonomic robot. It is constrained by one equality constraint (the linear velocity points along the car's main axis). Limits on the steering angle impose two inequality constraints. Other nonholonomic robots include tractor-trailers, airplanes, and satellites.

Given an arbitrary subset $U \subset \mathcal{C}$, the configuration $\boldsymbol{q}_1 \in U$ is said to be *U-accessible* from $\boldsymbol{q}_0 \in U$ if there exists a piecewise constant control $\dot{\boldsymbol{q}}(t)$ in the control set whose integral is a trajectory joining $\boldsymbol{q}_0$ to $\boldsymbol{q}_1$ that lies fully in U. Let $A_U(\boldsymbol{q}_0)$ be the set of configurations U-accessible from $\boldsymbol{q}_0$. The robot is said to be ***locally controllable*** at $\boldsymbol{q}_0$ iff for every neighborhood U of $\boldsymbol{q}_0$, $A_U(\boldsymbol{q}_0)$ is also a neighborhood of $\boldsymbol{q}_0$. It is locally controllable iff this is true for all $\boldsymbol{q}_0 \in \mathcal{C}$. Car-like robots and tractor-trailers that can go forward and backward are locally controllable [BL93].

Let X and Y be two smooth vector fields on $\mathcal{C}$. The Lie bracket of X and Y, denoted by $[X, Y]$, is the smooth vector field on $\mathcal{C}$ defined by $[X, Y] = dY \cdot X - dX \cdot Y$, where dX and dY, respectively, denote the $m \times m$ matrices of the partial derivatives of the components of X and Y w.r.t. the configuration coordinates in a chart placed on $\mathcal{C}$. To get a better intuition of the Lie bracket, imagine a trajectory starting at an arbitrary configuration $\boldsymbol{q}_s$ and obtained by concatenating four subtrajectories: the first is the integral curve of X during time δt; the second, third, and fourth are the integral curves of Y, $-X$, and $-Y$, respectively, each during the same δt. Let $\boldsymbol{q}_f$ be the final configuration reached. A Taylor expansion yields:

$$\lim_{\delta t \to 0} \frac{\boldsymbol{q}_f - \boldsymbol{q}_s}{\delta t^2} \quad = \quad [X, Y].$$

Hence, if $[X, Y]$ is not a linear combination of X and Y, the above trajectory allows the robot to move away from $\boldsymbol{q}_s$ in a direction that is not contained in the vector subspace defined by $X(\boldsymbol{q}_s)$ and $Y(\boldsymbol{q}_s)$. But the motion along this new direction is an order of magnitude slower than along any direction $\alpha X(\boldsymbol{q}_s) + \beta Y(\boldsymbol{q}_s)$.

The ***control Lie algebra*** associated with the control set Ω, denoted by $L(\Omega)$, is the space of all linear combinations of vector fields in Ω closed by the Lie bracket operation. The following result derives from the Controllability Rank Condition Theorem [BL93]:

> *A robot is locally controllable if, for every $\boldsymbol{q} \in \mathcal{C}$, $\Omega(\boldsymbol{q})$ is symmetric with respect to the origin of $T_{\boldsymbol{q}}(\mathcal{C})$ and the set $\{X(\boldsymbol{q}) \mid X \in L(\Omega(\boldsymbol{q}))\}$ has dimension m.*

The minimal length of the Lie brackets required to construct $L(\Omega)$, when these brackets are expressed with vectors in Ω, is called the ***degree of nonholonomy*** of the robot. The degree of nonholonomy of a car-like robot is 2. Except at some singular configurations, the degree of nonholonomy of a tractor towing a chain of s trailers is $2+s$ [LR96]. Intuitively, the higher the degree of nonholonomy, the more complex (and the slower) the robot's maneuvers to perform some motions.

Nonholonomic motion planning was also studied in a purely-geometric setting, where one is interested in finding a path of *bounded curvature* (see e.g., [KKP11, KC13]). The reader is referred to Chapter 50 for further information.

PLANNING FOR CONTROLLABLE NONHOLONOMIC ROBOTS

Let A be a locally controllable nonholonomic robot. A necessary and sufficient condition for the existence of a feasible free path of A between two given configurations is that they lie in the same connected component of the *open* free space. Indeed, local controllability guarantees that a possibly nonfeasible path can be decomposed into a finite number of subpaths, each short enough to be replaced by a feasible free subpath. Hence, deciding if there exists a free path for a locally controllable nonholonomic robot has the same complexity as deciding if there exists a path for the holonomic robot having the same geometry.

Transforming a nonfeasible free path τ into a feasible one can be done by recursively decomposing τ into subpaths. The recursion halts at every subpath that can be replaced by a feasible free subpath. Specific substitution rules (e.g., Reeds and Shepp curves) have been defined for car-like robots [LJTM94]. The complexity of transforming a nonfeasible free path τ into a feasible one is of the form $O(\epsilon^d)$, where ϵ is the smallest clearance between the robot and the obstacles along τ and d is the degree of nonholonomy of the robot (see [LJTM94] for the case $d = 2$).

The algorithm in [BL93] directly constructs a nonholonomic path for a car-like or a tractor-trailer robot by searching a tree obtained by concatenating short feasible paths, starting at the robot's initial configuration. The planner is ***asymptotically complete***, i.e., it is guaranteed to find a path if one exists, provided that the lengths of the short feasible paths are small enough. It can also find paths that minimize the number of cusps (changes of sign of the linear velocity).

PLANNING FOR NONCONTROLLABLE ROBOTS

Motion planning for nonholonomic robots that are not locally controllable is much less understood. Research has almost exclusively focused on car-like robots that can only move forward. Results include:

- No obstacles: A complete synthesis of the shortest, no-cusp path for a point moving with a lower-bounded turning radius [BSBL94].
- Polygonal obstacles: An algorithm to decide whether there exists such a path between two configurations; it runs in time exponential in obstacle complexity [FW91].
- Convex obstacles: The algorithm in [ART95] computes a path in polynomial time under the assumptions that all obstacles are convex and their boundaries have a curvature radius greater than the minimum turning radius of the point.
- Other polynomial algorithms (e.g., [BL93]) require some sort of discretization and are only asymptotically complete.

PLANNING UNDER DIFFERENTIAL CONSTRAINTS

If a planning problem involves constraints on at least velocity and acceleration the problem is often referred to as kinodynamic planning. Typically nonholonomic constraints are kinematic and arise from wheels in contact. But nonholonomic constraints may arise from dynamics. Trajectory planning refers to the problem of planning for both a path and a velocity function. Recently, the term planning under differential constraints is used as a catch-all term that includes nonholonomic, kinodynamic and sometimes trajectory planning. Planning under differential constraints may involve discretization of constraints and decoupled approaches where some constraints are ignored at the beginning but are then gradually introduced (see [KL16]).

Sampling-based planners were initially developed for the "geometric," i.e., holonomic, setting of the problem, but it was not long before similar techniques for planning under differential constraints started to emerge. However, even in the randomized setting of sampling-based planners, planning under differential constraints remains to be very challenging and the design of such planners usually involves much more than straightforward extensions of existing geometric planners.

Sampling-based planners for problems with differential constraints operate in the *state space* $\mathcal{X}$. Typically, every *state* represents a pairing of a configuration and a velocity vector. An inherent difficulty of the problem involves the connection of two given states in the absence of obstacles—an operation called *steering*, or the *two-point boundary-value problem* (BVP).

For robotic systems in which steering can be quickly performed (see, e.g., [ŠO97, LSL99, KF10, WB13, PPK+12, XBPA15]) holonomic sampling-based planners, or their extensions, can be applied.

Sampling-based algorithms that build trees avoid steering. They simply attempt to move in the direction of sampled states and do not require to precisely reach them (see, e.g., [HKLR02, LK01c, PKV10, SK12, LLB14] and many others). As one of many examples, the planner described in [LK04a] samples robot actions, which consist of a velocity vector and time duration, in order to avoid steering. The idea of avoiding steering has been proven very powerful in the context of sampling-based planners and has led to the solution of complex problems with differential constraints.

Finally, a number of related works exist that augment sampling-based algorithms for instances of whole-body manipulation for humanoids [KKN+02, HBL+08, YPL+10, BSK11, BHB13, Hau14]. The primary concern in these works is the generation of dynamically-stable motions for high-dimensional bipedal robots.

PLANNING UNDER OTHER TYPES OF CONSTRAINTS

Some robotic systems are subjected to constraints for particular tasks, restricting the set of valid configurations to a manifold, which has measure zero with respect to the ambient space. Naturally, sampling in the ambient space can no longer effectively produce valid configurations. Some approaches target specific kinds of constraint problems, such as open- or closed-loop kinematic linkage constraints [TTCA10, CS05], end-effector pose constraints [BS10], or n-point contact constraints [Hau14]. Some approaches project invalid samples through gradient descent [YLK01] or by more robust means [KBV+12]. Recent successful algorithms for planning under abstract constraints are CBiRRT2 [BSK11]—which like RRT steps toward sampled points, but after every step it projects the current point back to the manifold—as well as AtlasRRT [JP13] and [KTSP16]. AtlasRRT constructs a piecewise-linear approximation of the manifold by computing tangent spaces. This approximation allows the planner to reason about a difficult, implicitly-defined space by projecting regions onto simpler spaces.

51.5 SOFTWARE FOR SAMPLING-BASED PLANNERS

Robot motion planning requires the integration of numerous software components, and several existing packages address this goal. The Open Motion Planning Library (OMPL) [SMK12] includes abstract implementations of many planning algorithms discussed in this chapter. The robot modeling—loading 3D meshes, kinematics, visualization, etc.—necessary to apply OMPL is handled by external packages such as MoveIt! [SC15], which integrates with the ROS framework [QGC$^+$09], and Amino [Dan16] which provides capabilities to support task planning and real-time control.

Software packages performing nearest-neighbor search include E2LSH [AI06], ANN [MA10], FLANN [ML10] and RTG [KSH15]. Collision checking is typically handled by specialized packages such as the Flexible Collision Library [PCM12] and libccd [Fis16]. The Orocos Kinematics and Dynamics Library (KDL) [Smi] is widely used for basic kinematics and dynamics computations. Several dedicated packages provide more advanced capabilities for dynamic simulation [S$^+$05, LGS$^+$16, C$^+$16, SSD11]. Other all-in-one robotics packages include OpenRAVE [Dia10] and Klamp't [Hau16].

51.6 APPLICATIONS BEYOND ROBOTICS

Although the focus of this chapter was restricted to applications of motion planning in robotics, we mention that the problem naturally arises in many other settings [Lat99]. For instance, in computer games[2] motion planning is used for determining the motion of single or multiple agents. In structural bioinformatics motion-planning tools are employed in order to study the motion of molecules such a proteins, which can be modeled as highly-complex robots. This is one application domain where sampling-based tools enable motion planning with hundreds of degrees of freedom [LK01a, RESH09, GMK13, ETA15, ADS03, ABG$^+$03]. In Computer-Aided-Design (CAD) motion planning has been used to find intricate separation paths to take out parts for maintenance out of a complex and cluttered environment [CL95, FL04].

51.7 SOURCES AND RELATED MATERIAL

Craig's book [Cra05] provides an introduction to robot arm kinematics, dynamics, and control. The mechanics of robotic manipulation is covered in Mason's book [Mas01]. The book by Siciliano and Khatib [SK16] provides a comprehensive overview of the field of robotics.

Robot motion planning and its variants are discussed in Latombe's book [Lat91]. This book takes an algorithmic approach to a variety of advanced issues in robotics. The books by LaValle [LaV06] and Choset et al. [CLS$^+$05] provide a more current overview of motion planning, and also include detailed description

[2] See the *Motion in Games conference*: `http://www.motioningames.org/`.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

of several sampling-based approaches.

Algorithmic issues in robotics are covered in several conferences including the Workshop on Algorithmic Foundations of Robotics (WAFR), Robotics: Science and Systems (RSS), and the broader IEEE International Conference on Robotics and Automation (ICRA) and the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Several computational-geometry books contain sections on robotics or motion planning [O'R98, SA95, BKOC08]. The book by Fogel et al. [FHW12] includes implementation details for basic geometric constructions useful in motion planning, such as Minkowski sums.

RELATED CHAPTERS

Chapter 28: Arrangements
Chapter 31: Shortest paths and networks
Chapter 33: Visibility
Chapter 34: Geometric reconstruction problems
Chapter 37: Computational and quantitative real algebraic geometry
Chapter 39: Collision and proximity queries
Chapter 50: Algorithmic motion planning
Chapter 53: Modeling motion
Chapter 60: Geometric applications of the Grassmann-Cayley algebra
Chapter 68: Two computational geometry libraries: LEDA and CGAL

REFERENCES

[ABG+03] M.S. Apaydin, D.L. Brutlag, C. Guestrin, D. Hsu, J.-C. Latombe, and C. Varma. Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion. *J. Comput. Biol.*, 10:257–281, 2003.

[ABHS15] A. Adler, M. de Berg, D. Halperin, and K. Solovey. Efficient multi-robot motion planning for unlabeled discs in simple polygons. *IEEE Trans. Autom. Sci. Eng.*, 12:1309–1317, 2015.

[ABR12] T.F. Allen, J.W. Burdick, and E. Rimon. Two-fingered caging of polygons via contact-space graph search. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 4183–4189, 2012.

[ACH01] P.K. Agarwal, A.D. Collins, and J.L. Harer. Minimal trap design. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2243–2248, 2001.

[ADS03] N.M. Amato, K.A. Dill, and G. Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. *J. Comput. Biol.*, 10:239–255, 2003.

[AHLM00] S. Akella, W.H. Huang, K.M. Lynch, and M.T. Mason. Parts feeding on a conveyor with a one joint robot. *Algorithmica*, 26:313–344, 2000.

[AI06] A. Andoni and P. Indyk. LSH algorithm and implementation. `http://web.mit.edu/andoni/www/LSH/`, 2006.

[ALS95] R. Alami, J.-P. Laumond, and T. Siméon. Two manipulation planning algorithms. In *Algorithmic Foundations of Robotics*, pages 109–125, A.K. Peters, Natick, 1995.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

[ARB14] T.F. Allen, E. Rimon, and J.W. Burdick. Two-finger caging of 3D polyhedra using contact space search. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2005–2012, 2014.

[ARB15] T.F. Allen, E. Rimon, and J.W. Burdick. Robust three-finger three-parameter caging of convex polygons. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 4318–4325, 2015.

[ART95] P.K. Agarwal, P. Raghavan, and H. Tamaki. Motion planning for a steering-constrained robot through moderate obstacles. In *27th ACM Sympos. Theory Comput.*, pages 343–352, 1995.

[ASN+16] S. Aine, S. Swaminathan, V. Narayanan, V. Hwang, and M. Likhachev. Multi-heuristic A*. *I. J. Robotics Res.*, 35:224–243, 2016.

[AT13] O. Arslan and P. Tsiotras. Use of relaxation methods in sampling-based algorithms for optimal motion planning. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2421–2428, 2013.

[BB15] D. Bareiss and J.P. van den Berg. Generalized reciprocal collision avoidance. *I. J. Robotics Res.*, 34:1501–1514, 2015.

[BBDG00] K.-F. Böhringer, V. Bhatt, B.R. Donald, and K.Y. Goldberg. Algorithms for sensorless manipulation using a vibrating surface. *Algorithmica*, 26:389–429, 2000.

[BBT02] M. Bennewitz, W. Burgard, and S. Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics Autonomous Systems*, 41:89–99, 2002.

[BDH99] K.-F. Böhringer, B.R. Donald, and D. Halperin. On the area bisectors of a polygon. *Discrete Comput. Geom.*, 22:269–285, 1999.

[BDKL00] K.-F. Böhringer, B.R. Donald, L.E. Kavraki, and F. Lamiraux. Part orientation with one or two stable equilibria using programmable force fields. *IEEE Trans. Robot. Autom.*, 16:157–170, 2000.

[BDM99] K.F. Böhringer, B.R. Donald, and N.C. MacDonald. Programmable force fields for distributed manipulation, with applications to MEMS actuator arrays and vibratory parts feeders. *I. J. Robotics Res.*, 18:168–200, 1999.

[BG96] R.C. Brost and K.Y. Goldberg. A complete algorithm for designing planar fixtures using modular components. *IEEE Trans. Robot. Autom.*, 12:31–46, 1996.

[BG13] M. de Berg and D.H.P. Gerrits. Computing push plans for disk-shaped robots. *Internat. J. Comput. Geom. Appl.*, 23:29–48, 2013.

[BGHN01] J. Basch, L.J. Guibas, D. Hsu, and A.T. Nguyen. Disconnection proofs for motion planning. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1765–1772, 2001.

[BGMK12] K.E. Bekris, D.K. Grady, M. Moll, and L.E. Kavraki. Safe distributed motion coordination for second-order systems with different planning cycles. *I. J. Robotics Res.*, 31:129–150, 2012.

[BGOS98] R.-P. Berretty, K.Y. Goldberg, M.H. Overmars, and A.F. van der Stappen. Computing fence designs for orienting parts. *Comput. Geom.*, 10:249–262, 1998.

[BGOS01] R.-P. Berretty, K.Y. Goldberg, M.H. Overmars, and A.F. van der Stappen. Trap design for vibratory bowl feeders. *I. J. Robotics Res.*, 20:891–908, 2001.

[BH95] M. Barbehenn and S. Hutchinson. Efficient search and hierarchical motion planning by dynamically maintaining single-source shortest paths trees. *IEEE Trans. Robot. Autom.*, 11:198–214, 1995.

[BHB13] F. Burget, A. Hornung, and M. Bennewitz. Whole-body motion planning for manipulation of articulated objects. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1656–1662, 2013.

[BK00] R. Bohlin and L.E. Kavraki. Path planning using lazy PRM. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 521–528, 2000.

[BKOC08] M. de Berg, M. van Kreveld, M. Overmars, and O. Cheong. *Computational Geometry: Algorithms and Applications*, 3rd edition. Springer-Verlag, Berlin, 2008.

[BKV10] A. Bhatia, L.E. Kavraki, and M.Y. Vardi. Motion planning with hybrid dynamics and temporal goals. In *Proc. IEEE Conf. Decision and Control*, pages 1108–1115, 2010.

[BL93] J. Barraquand and J.-C. Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. *Algorithmica*, 10:121–155, 1993.

[BMKV11] A. Bhatia, M.R. Maly, L.E. Kavraki, and M.Y. Vardi. Motion planning with complex goals. *IEEE Robot. Autom. Mag.*, 18:55–64, 2011.

[BO05] J.P. van den Berg and M.H. Overmars. Prioritized motion planning for multiple robots. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 430–435, 2005.

[BOS99] V. Boor, M.H. Overmars, and A.F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1018–1023, 1999.

[BS10] D. Berenson and S. Srinivasa. Probabilistically complete planning with end-effector pose constraints. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2724–2730, 2010.

[BSBL94] X.N. Bui, P. Souères, J.-D. Boissonnat, and J.-P. Laumond. Shortest path synthesis for dubins non-holonomic robot. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2–7, 1994.

[BSK$^+$08] J. van den Berg, M. Stilman, J. Kuffner, M.C. Lin, and D. Manocha. Path planning among movable obstacles: A probabilistically complete approach. In *Algorithmic Foundations of Robotics VIII*, pages 599–614, Springer, Berlin, 2008.

[BSK11] D. Berenson, S. Srinivasa, and J.J. Kuffner. Task space regions: A framework for pose-constrained manipulation planning. *I. J. Robotics Res.*, 30:1435–1460, 2011.

[BSLM09] J.P. van den Berg, J. Snoeyink, M.C. Lin, and D. Manocha. Centralized path planning for multiple robots: Optimal decoupling into sequential plans. In *Robotics: Science and Systems*, 2009.

[C$^+$16] E. Coumans et al. Bullet physics library. `http://bulletphysics.org/`, 2016.

[CI95] Y.B. Chen and D.J. Ierardi. The complexity of oblivious plans for orienting and distinguishing polygonal parts. *Algorithmica*, 14:367–397, 1995.

[CJS08] J. Cortés, L. Jaillet, and T. Siméon. Disassembly path planning for complex articulated objects. *IEEE Trans. Robot.*, 24:475–481, 2008.

[CL95] H. Chang and T.-Y. Li. Assembly maintainability study with motion planning. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1012–1019, 1995.

[CLS$^+$05] H. Choset, K.M. Lynch, S.Hutchinson, G.A. Kantor, W. Burgard, L.E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations.* MIT Press, Cambridge, 2005.

[Cra05] J.J. Craig. *Introduction to Robotics: Mechanics and Control*, 3rd edition. Pearson Prentice Hall, Upper Saddle River, 2005.

[CS05] J. Cortés and T. Siméon. Sampling-based motion planning under kinematic loop-closure constraints. In *Algorithmic Foundations of Robotics VI*, pages 75–90, Springer, Berlin, 2005.

[CSG+07] J.-S. Cheong, A.F. van der Stappen, K.Y. Goldberg, M.H. Overmars, and E. Rimon. Immobilizing hinged polygons. *Internat. J. Comput. Geom. Appl.*, 17:45–70, 2007.

[Dan16] N.T. Dantam. Amino. `http://amino.golems.org`, 2016.

[DB98] C. Davidson and A. Blake. Caging planar objects with a three-finger one-parameter gripper. In *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, pages 2722–2727, 1998.

[DB14] A. Dobson and K.E. Bekris. Sparse roadmap spanners for asymptotically near-optimal motion planning. *I. J. Robotics Res.*, 33:18–47, 2014.

[Dia10] R. Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute, August 2010.

[DKCK16] N.T. Dantam, Z.K. Kingston, S. Chaudhuri, and L.E. Kavraki. Incremental task and motion planning: A constraint-based approach. In *Robotics: Science and Systems*, 2016.

[DMB15] A. Dobson, G.V. Moustakides, and K.E. Bekris. Geometric probability results for bounding path quality in sampling-based roadmaps after finite computation. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 4180–4186, 2015.

[Don87] B.R. Donald. A search algorithm for motion planning with six degrees of freedom. *Artif. Intell.*, 31(3):295–353, 1987.

[DRP+14] N.C. Dafle, A. Rodriguez, R. Paolini, B. Tang, S.S. Srinivasa, M. Erdmann, M.T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge. Extrinsic dexterity: In-hand manipulation with external forces. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1578–1585, 2014.

[EM88] M.A. Erdmann and M.T. Mason. An exploration of sensorless manipulation. *IEEE J. Robot. Autom.*, 4:369–379, 1988.

[EMJ93] M.A. Erdmann, M.T. Mason, and G. Vaněček Jr. Mechanical parts orienting: The case of a polyhedron on a table. *Algorithmica*, 10:226–247, 1993.

[Epp90] D. Eppstein. Reset sequences for monotonic automata. *SIAM J. Comput.*, 19:500–510, 1990.

[ETA15] C. Ekenna, S. Thomas, and N.M. Amato. Adaptive local learning in sampling based motion planning for protein folding. In *Proc. IEEE Intl. Conf. Bioinf. Biomed.*, pages 61–68, 2015.

[ETRP07] J. Erickson, S. Thite, F. Rothganger, and J. Ponce. Capturing a convex object with three discs. *IEEE Trans. Robot.*, 23:1133–1140, 2007.

[FC92] C. Ferrari and J. Canny. Planning optimal grasps. In *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, pages 2290–2295, 1992.

[FGKP09] G.E. Fainekos, A. Girard, H. Kress-Gazit, and G.J. Pappas. Temporal logic motion planning for dynamic robots. *Automatica*, 45:343–352, 2009.

[FH13] E. Fogel and D. Halperin. Polyhedral assembly partitioning with infinite translations or the importance of being exact. *IEEE Trans. Autom. Sci. Eng.*, 10:227–241, 2013.

[FHK08] J. Fink, M.A. Hsieh, and V. Kumar. Multi-robot manipulation via caging in environments with obstacles. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1471–1476, 2008.

[FHW12] E. Fogel, D. Halperin, and R. Wein. *CGAL Arrangements and Their Applications: A Step-by-Step Guide*. Vol 7 of *Geometry and Computing*, Springer, Berlin, 2012.

[Fis16] D. Fiser. `libccd`. `http://libccd.danfis.cz/`, 2016.

[FL04] E. Ferre and J.-P. Laumond. An iterative diffusion algorithm for part disassembly. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 3149–3154, 2004.

[FW91] S. Fortune and G. Wilfong. Planning constrained motion. *Ann. Math. Artif. Intell.*, 3:21–82, 1991.

[GG05] K. Gopalakrishnan and K.Y. Goldberg. D-space and deform closure grasps of deformable parts. *I. J. Robotics Res.*, 24:899–910, 2005.

[GGB+04] K. Gopalakrishnan, K.Y. Goldberg, G.M. Bone, M. Zaluzec, R. Koganti, R. Pearson, and P. Deneszczuk. Unilateral fixtures for sheet-metal parts with holes. *IEEE Trans. Autom. Sci. Eng.*, 1:110–120, 2004.

[GGGDC12] J. Guzzi, A. Giusti, L.M. Gambardella, and G.A. Di Caro. Bioinspired obstacle avoidance algorithms for robot swarms. In *Bio-Inspired Models of Network, Information, and Computing Systems*, pages 120–134, Springer, Berlin, 2012.

[GGS06] O.C. Goemans, K.Y. Goldberg, and A.F. van der Stappen. Blades: A new class of geometric primitives for feeding 3D parts on vibratory tracks. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1730–1736, 2006.

[GH10] G. Goraly and R. Hassin. Multi-color pebble motion on graphs. *Algorithmica*, 58:610–636, 2010.

[GHH+98] L.J. Guibas, D. Halperin, H. Hirukawa, J.-C. Latombe, and R.H. Wilson. Polyhedral assembly partitioning using maximally covered cells in arrangements of convex polytopes. *Internat. J. Comput. Geom. Appl.*, 8:179–200, 1998.

[GMK13] B. Gipson, M. Moll, and L.E. Kavraki. SIMS: A hybrid method for rapid conformational analysis. *PLoS ONE*, 8:e68826, 2013.

[GO07] R. Geraerts and M.H. Overmars. Creating high-quality paths for motion planning. *I. J. Robotics Res.*, 26:845–863, 2007.

[Gol93] K.Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10:210–225, 1993.

[Hau14] K. Hauser. Fast interpolation and time-optimization with contact. *I. J. Robotics Res.*, 33:1231–1250, 2014.

[Hau15] K. Hauser. Lazy collision checking in asymptotically-optimal motion planning. In *proc. IEEE Int. Conf. Robot. Autom.*, pages 2951–2957, 2015.

[Hau16] K. Hauser. Robust contact generation for robot simulation with unstructured meshes. In *Robotics Research*, pages 357–373, Springer, Berlin, 2016. `http://motion.pratt.duke.edu/klampt/`.

[HBL+08] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada, and B. Wilcox. Motion planning for legged robots on varied terrain. *I. J. Robotics Res.*, 27:1325–1349, 2008.

[HD05] R.A. Hearn and E.D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theor. Comput. Sci.*, 343:72–96, 2005.

[HdMS91] L.S. Homem de Mello and A.C. Sanderson. A correct and complete algorithm for the generation of mechanical assembly sequences. *IEEE Trans. Robot. Autom.*, 7:228–240, 1991.

[HGL+97] L. Han, Y.S. Guan, Z.X. Li, Q. Shi, and J.C. Trinkle. Dextrous manipulation with rolling contacts. In *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, pages 992–997, 1997.

[HH02] S. Hirsch and D. Halperin. Hybrid motion planning: Coordinating two discs moving among polygonal obstacles in the plane. In *Algorithmic Foundations of Robotics V*, pages 239–256, Springer, Berlin, 2002.

[HKLR02] D. Hsu, R. Kindel, J.-C. Latombe, and S.M. Rock. Randomized kinodynamic motion planning with moving obstacles. *I. J. Robotics Res.*, 21:233–256, 2002.

[HLM99] D. Hsu, J.-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *Internat. J. Comput. Geom. Appl.*, 9:495–512, 1999.

[HLW00] D. Halperin, J.-C. Latombe, and R.H. Wilson. A general framework for assembly planning: The motion space approach. *Algorithmica*, 26:577–601, 2000.

[HN01] J. Hoffmann and B. Nebel. The FF planning system: Fast plan generation through heuristic search. *J. Artif. Int. Res.*, 14:253–302, 2001.

[HW96] D. Halperin and R.H. Wilson. Assembly partitioning along simple paths: The case of multiple translations. *Advanced Robotics*, 11:127–145, 1996.

[IS15] F. Imeson and S.L. Smith. Multi-robot task planning and sequencing using the SAT-TSP language. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 5397–5402, 2015.

[Jim13] P. Jiménez. Survey on assembly sequencing: A combinatorial and geometrical perspective. *J. Intell. Manufacturing*, 24:235–250, 2013.

[JP13] L. Jaillet and J.M. Porta. Path planning under kinematic constraints by rapidly exploring manifolds. *IEEE Trans. Robot.*, 29:105–117, 2013.

[JSCP15] L. Janson, E. Schmerling, A.A. Clark, and M. Pavone. Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *I. J. Robotics Res.*, 34:883–921, 2015.

[Kav97] L.E. Kavraki. Part orientation with programmable vector fields: Two stable equilibria for most parts. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2446–2451, 1997.

[KB08] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Trans. Autom. Control*, 53:287–297, 2008.

[KB15] A. Krontiris and K.E. Bekris. Dealing with difficult instances of object rearrangement. In *Proc. Robotics: Science and Systems*, 2015.

[KBV+12] P. Kaiser, D. Berenson, N. Vahrenkamp, T. Asfour, R. Dillmann, and S. Srinivasa. Constellation – An algorithm for finding robot configurations that satisfy multiple constraints. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 436–443, 2012.

[KC13] H.-S. Kim and O. Cheong. The cost of bounded curvature. *Comput. Geom.*, 46:648–672, 2013.

[KDB11] M. Kloetzer, X.C. Ding, and C. Belta. Multi-robot deployment from LTL specifications with reduced communication. In *Proc. IEEE Conf. Decision and Control*, pages 4867–4872, 2011.

[KEK10] G.A. Kaminka, D. Erusalimchik, and S. Kraus. Adaptive multi-robot coordination: A game-theoretic perspective. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 328–334, 2010.

[KF10] S. Karaman and E. Frazzoli. Optimal kinodynamic motion planning using incremental sampling-based methods. In *Proc. IEEE Conf. Decision and Control*, pages 7681–7687, 2010.

[KF11] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *I. J. Robotics Res.*, 30:846–894, 2011.

[KF12] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning with deterministic μ-calculus specifications. In *Proc. American Control Conference*, pages 735–742, 2012.

[KH06] S. Kloder and S. Hutchinson. Path planning for permutation-invariant multirobot formations. *IEEE Trans. Robot.*, 22:650–665, 2006.

[Kha86] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *I. J. Robotics Res.*, 5:90–98, 1986.

[KK95] L.E. Kavraki and M.N. Kolountzakis. Partitioning a planar assembly into two connected parts is NP-complete. *Inform. Process. Lett.*, 55:159–165, 1995.

[KKL98] L.E. Kavraki, M.N. Kolountzakis, and J.-C. Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Trans. Robot. Autom.*, 14:166–171, 1998.

[KKN$^+$02] J.J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue. Dynamically-stable motion planning for humanoid robots. *Autonomous Robots*, 12:105–118, 2002.

[KKP11] D. Kirkpatrick, I. Kostitsyna, and V. Polishchuk. Hardness results for two-dimensional curvature-constrained motion planning. In *Proc. 23rd Canad. Conf. Comput. Geom.*, Toronto, 2011.

[KL16] L.E. Kavraki and S.M. LaValle. Motion planning. Chap. 7 in B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, 2nd edition, pages 139–158, Springer, Berlin, 2016.

[KLB13] A. Krontiris, R. Luna, and K.E. Bekris. From feasibility tests to path planners for multi-agent pathfinding. In *Proc. 6th AAAI Int. Sympos. Combin. Search*, 2013.

[KLMR98] L.E. Kavraki, J.-C. Latombe, R. Motwani, and P. Raghavan. Randomized query processing in robot path planning. *J. Comput. Syst. Sci.*, 57:50–66, 1998.

[KMS84] D. Kornhauser, G. Miller, and P. Spirakis. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications. In *Proc. 54th IEEE Sympos. Found. Comp. Sci.*, pages 241–250, 1984.

[KMY92] D. Kirkpatrick, B. Mishra, and C.K. Yap. Quantitative Steinitz's theorems applications to multifingered grasping. *Discrete Comput. Geom.*, 7:295–318, 1992.

[Kob12] M. Kobilarov. Cross-entropy motion planning. *I. J. Robotics Res.*, 31:855–871, 2012.

[KR12] R.A. Knepper and D. Rus. Pedestrian-inspired sampling-based multi-robot collision avoidance. In *Proc. IEEE Int. Sympos. Robot Human Interactive Comm.*, pages 94–100, 2012.

[KS92] H.A. Kautz and B. Selman. Planning as satisfiability. *proc. 10th European Conf. Artif. Intell.*, pages 359–363, Wiley, New York, 1992.

[KSD$^+$14] A. Krontiris, R. Shome, A. Dobson, A. Kimmel, and K.E. Bekris. Rearranging similar objects with a manipulator using pebble graphs. In *Proc. IEEE/RAS Int. Conf. Humanoid Robots*, pages 1081–1087, 2014.

[KSH15] M. Kleinbort, O. Salzman, and D. Halperin. Efficient high-quality motion planning by fast all-pairs r-nearest-neighbors. In *IEEE Int. Conf. Robot. Autom.*, pages 2985–2990, 2015.

[KŠLO96] L.E. Kavraki, P. Švestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.*, 12:566–580, 1996.

[KTSP16] B. Kim, T.U. Taewoong, C. Suh, and F.C. Park. Tangent bundle RRT: A randomized algorithm for constrained motion planning. *Robotica*, 34:202–225, 2016.

[LAF+15] M. Lahijanian, S. Almagor, D. Fried, L.E. Kavraki, and M.Y. Vardi. This time the robot settles for a cost: A quantitative approach to temporal logic planning with partial satisfaction. In *AAAI Conf. Artif. Intell.*, pages 3664–3671, 2015.

[Lat91] J.-C. Latombe. *Robot Motion Planning*. Volume 124 of *Int. Ser. Engin. Comp. Sci.*, Kluwer, Boston, 1991.

[Lat99] J.-C. Latombe. Motion planning: A journey of robots, molecules, digital actors, and other artifacts. *I. J. Robotics Res.*, 18:1119–1128, 1999.

[LaV06] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[LCS09] D.T. Le, J. Cortés, and T. Siméon. A path planning approach to (dis)assembly sequencing. In *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, pages 286–291, 2009.

[LF09] M. Likhachev and D. Ferguson. Planning long dynamically feasible maneuvers for autonomous vehicles. *I. J. Robotics Res.*, 28:933–945, 2009.

[LGS+16] J. Lee, M.X. Grey, M. Stilman, K. Liu, et al. Dart simulator. `http://dartsim.github.io/`, 2016.

[LJTM94] J.-P. Laumond, P.E. Jacobs, M. Taïx, and R.M. Murray. A motion planner for nonholonomic mobile robots. *IEEE Trans. Robot. Autom.*, 10:577–593, 1994.

[LK01a] F. Lamiraux and L.E. Kavraki. Planning paths for elastic objects under manipulation constraints. *I. J. Robotics Res.*, 20:188–208, 2001.

[LK01b] F. Lamiraux and L.E. Kavraki. Positioning of symmetric and nonsymmetric parts using radial and constant fields: Computation of all equilibrium configurations. *I. J. Robotics Res.*, 20:635–659, 2001.

[LK01c] S.M. LaValle and J.J. Kuffner. Randomized kinodynamic planning. *I. J. Robotics Res.*, 20:378–400, 2001.

[LK04a] A.M. Ladd and L.E. Kavraki. Fast tree-based exploration of state space for robots with dynamics. In *Algorithmic Foundations of Robotics VI*, pages 297–312, Springer, Berlin, 2004.

[LK04b] A.M. Ladd and L.E. Kavraki. Measure theoretic analysis of probabilistic path planning. *IEEE Trans. Robot. Autom.*, 20:229–242, 2004.

[LLB14] Y. Li, Z. Littlefield, and K.E. Bekris. Sparse methods for efficient asymptotically optimal kinodynamic planning. In *Algorithmic Foundations of Robotics XI*, pages 263–282, Springer, Berlin, 2014.

[LLG+15] A.X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel. Learning force-based manipulation of deformable objects from multiple demonstrations. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 177–184, 2015.

[LLS99] S. Leroy, J.-P. Laumond, and T. Siméon. Multiple path coordination for mobile robots: A geometric algorithm. In *Proc. 16th Int. Joint Conf. Artif. Intell.*, pages 1118–1123, Morgan Kaufmann, San Francisco, 1999.

[LMC01] J.E. Luntz, W.C. Messner, and H. Choset. Distributed manipulation using discrete actuator arrays. *I. J. Robotics Res.*, 20:553–583, 2001.

[LMS14] B. León, A. Morales, and J. Sancho-Bru. *From Robot to Human Grasping Simulation*. Vol. 19 of *Cognitive Systems Monographs*, Springer, Berlin, 2014.

[LP83] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Trans. Comput.*, C-32:108–120, 1983.

[LR96] J.-P. Laumond and J.-J. Risler. Nonholonomic systems: Controllability and complexity. *Theoret. Comput. Sci.*, 157:101–114, 1996.

[LSL99] F. Lamiraux, S. Sekhavat, and J.-P. Laumond. Motion planning and control for hilare pulling a trailer. *IEEE Trans. Robot. Autom.*, 15:640–652, 1999.

[LTA03] J.-M. Lien, S.L. Thomas, and N.M. Amato. A general framework for sampling on the medial axis of the free space. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 4439–4444, 2003.

[LWA15] S. Levine, N. Wagener, and P. Abbeel. Learning contact-rich manipulation skills with guided policy search. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 156–163, 2015.

[MA10] D.M. Mount and S. Arya. ANN: A library for approximate nearest neighbor searching. `https://www.cs.umd.edu/~mount/ANN/`, 2010.

[Mas82] M.T. Mason. *Manipulation by Grasping and Pushing Operations.* PhD. thesis, MIT, Artificial Intelligence Lab., Cambridge, 1982.

[Mas01] M.T. Mason. *Mechanics of Robotic Manipulation.* MIT Press, Cambridge, 2001.

[Mat95] R.S. Mattikalli. *Mechanics Based Assembly Planning.* PhD. thesis, Carnegie Mellon Univ., Pittsburgh, 1995.

[MBH12] Z. McCarthy, T. Bretl, and S. Hutchinson. Proving path non-existence using sampling and alpha shapes. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2563–2569, 2012.

[ME02] M. Moll and M.A. Erdmann. Manipulation of pose distributions. *I. J. Robotics Res.*, 21:277–292, 2002.

[MGEF02] M. Moll, K. Goldberg, M.A. Erdmann, and R. Fearing. Aligning parts for micro assemblies. *Assembly Automation*, 22:46–54, 2002.

[Mis91] B. Mishra. Workholding: analysis and planning. In *Proc. IEEE/RSJ Int. Workshop Intell. Robots and Systems*, pages 53–57, 1991.

[Mis95] B. Mishra. Grasp metrics: Optimality and complexity. In *Proc. Algorithmic Foundations of Robotics I*, pages 137–165, A.K. Peters, Natick, 1995.

[ML10] M. Muja and D.G. Lowe. FLANN - Fast library for approximate nearest neighbors. `http://www.cs.ubc.ca/research/flann/`, 2010.

[MLP90] X. Markenscoff, N. Luqun, and C.H. Papadimitriou. The geometry of grasping. *I. J. Robotics Res.*, 9:61–74, 1990.

[MSS87] B. Mishra, J.T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2:541–558, 1987.

[MZG$^+$96] B. Mirtich, Y. Zhuang, K.Y. Goldberg, J. Craig, R. Zanutta, B. Carlisle, and J.F. Canny. Estimating pose statistics for robotic part feeders. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1140–1146, 1996.

[Nat88] B.K. Natarajan. On planning assemblies. In *Proc. 4th Sympos. Comput. Geom.*, pages 299–308, ACM Press, 1988.

[Nat89] B.K. Natarajan. Some paradigms for the automated design of parts feeders. *I. J. Robotics Res.*, 8:98–109, 1989.

[Ngu88] V.-D. Nguyen. Constructing force-closure grasps. *I. J. Robotics Res.*, 7:3–16, 1988.

[NO04] D. Nieuwenhuisen and M.H. Overmars. Useful cycles in probabilistic roadmap graphs. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 446–452, 2004.

[NRH10] O. Nechushtan, B. Raveh, and D. Halperin. Sampling-diagram automata: A tool for analyzing path quality in tree planners. In *Algorithmic Foundations of Robotics IX*, pages 285–301, Springer, Berlin, 2010.

[NSO08] D. Nieuwenhuisen, A.F. van der Stappen, and M.H. Overmars. An effective framework for path planning amidst movable obstacles. In *Algorithmic Foundations of Robotics VIII*, pages 87–102, Springer, Berlin, 2008.

[NSO07] D. Nieuwenhuisen, A.F. van der Stappen, and M.H. Overmars. Pushing a disk using compliance. *IEEE Trans. Robot.*, 23:431–442, 2007.

[O'R98] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1998.

[ORSW95] M.H. Overmars, A. Rao, O. Schwarzkopf, and C. Wentink. Immobilizing polygons against a wall. In *Proc. 11th Sympos. Comput. Geom.*, pages 29–38, ACM Press, 1995.

[PCM12] J. Pan, S. Chitta, and D. Manocha. FCL: A general purpose library for collision and proximity queries. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 3859–3866, 2012.

[PKV10] E. Plaku, L.E. Kavraki, and M.Y. Vardi. Motion planning with dynamics by a synergistic combination of layers of planning. *IEEE Trans. Robot.*, 26:469–482, jun. 2010.

[Pnu77] A. Pnueli. The temporal logic of programs. In *Proc. 18th IEEE Sympos. Found. Comp. Sci.*, pages 46–57, 1977.

[PPK+12] A. Perez, R. Platt, G. Konidaris, L.P. Kaelbling, and T Lozano-Pérez. LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2537–2542, 2012.

[PS06] P. Pipattanasomporn and A. Sudsang. Two-finger caging of concave polygon. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2137–2142, 2006.

[PS11] P. Pipattanasomporn and A. Sudsang. Two-finger caging of nonconvex polytopes. *IEEE Trans. Robot.*, 27:324–333, 2011.

[PS15] F. Panahi and A.F. van der Stappen. Reprint of: Bounding the locus of the center of mass for a part with shape variation. *Comput. Geom.*, 48:398–406, 2015.

[PSS+97] J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *I. J. Robotics Res.*, 16:11–35, 1997.

[PVS08] P. Pipattanasomporn, P. Vongmasa, and A. Sudsang. Caging rigid polytopes via finger dispersion control. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1181–1186, 2008.

[QGC+09] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: An open-source robot operating system. In *IEEE ICRA Workshop on Open Source Robotics*, 2009.

[RB98] E. Rimon and J.W. Burdick. Mobility of bodies in contact. i. A 2nd-order mobility index for multiple-finger grasps. *IEEE Trans. Robot. Autom.*, 14:696–708, 1998.

[RB99] E. Rimon and A. Blake. Caging planar bodies by one-parameter two-fingered gripping systems. *I. J. Robotics Res.*, 18:299–318, 1999.

[REH11] B. Raveh, A. Enosh, and D. Halperin. A little more, a lot better: Improving path quality by a path-merging algorithm. *IEEE Trans. Robot.*, 27:365–371, 2011.

[RESH09] B. Raveh, A. Enosh, O. Schueler-Furman, and D. Halperin. Rapid sampling of molecular motions with prior information constraints. *PLoS Comput. Biol.*, 5:e1000295, 2009.

[RG95] A.S. Rao and K.Y. Goldberg. Manipulating algebraic parts in the plane. *IEEE Trans. Robot. Autom.*, 11:598–602, 1995.

[RMC00] D. Reznik, E. Moshkovich, and J.F. Canny. Building a universal planar manipulator. In K.F. Böhringer and H. Choset, editors, *Distributed Manipulation*, pages 147–171, Springer, New York, 2000.

[RMF12] A. Rodriguez, M.T. Mason, and S. Ferry. From caging to grasping. *I. J. Robotics Res.*, 31(7):886–900, 2012.

[Rus99] D. Rus. In-hand dexterous manipulation of piecewise-smooth 3-d objects. *I. J. Robotics Res.*, 18:355–381, 1999.

[RS12] E. Rimon and A.F. van der Stappen. Immobilizing 2-d serial chains in form-closure grasps. *IEEE Trans. Robot*, 28:32–43, 2012.

[S+05] R. Smith et al. Open dynamics engine. `http://www.ode.org/`, 2005.

[SA95] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, 1995.

[SBGM11] J. Snape, J.P. van den Berg, S.J. Guy, and D. Manocha. The hybrid reciprocal velocity obstacle. *IEEE Trans. Robot.*, 27:696–706, 2011.

[SC15] I. Sucan and S. Chitta. MoveIt! `http://moveit.ros.org`, 2015.

[SDH+14] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel. Motion planning with sequential convex optimization and convex collision checking. *I. J. Robotics Res.*, 33:1251–1270, 2014.

[SEBK15] J.A. Stork, C.H. Ek, Y. Bekiroglu, and D. Kragic. Learning predictive state representation for in-hand manipulation. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 3207–3214, 2015.

[SFR+14] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S.J. Russell, and P. Abbeel. Combined task and motion planning through an extensible planner-independent interface layer. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 639–646, 2014.

[SG00] A.F. van der Stappen and K.Y. Goldberg. Geometric eccentricity and the complexity of manipulation plans. *Algorithmica*, 26:494–514, 2000.

[SH14a] O. Salzman and D. Halperin. Asymptotically near-optimal RRT for fast, high-quality, motion planning. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 4680–4685, 2014.

[SH14b] K. Solovey and D. Halperin. k-Color multi-robot motion planning. *I. J. Robotics Res.*, 33:82–97, 2014.

[SH15a] O. Salzman and D. Halperin. Asymptotically-optimal motion planning using lower bounds on cost. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 4167–4172, 2015.

[SH15b] K. Solovey and D. Halperin. On the hardness of unlabeled multi-robot motion planning. In *Proc. Robotics: Science and Systems*, 2015.

[SHH15] O. Salzman, M. Hemmer, and D. Halperin. On the power of manifold samples in exploring configuration spaces and the dimensionality of narrow passages. *IEEE Trans. Autom. Sci. Eng.*, 12:529–538, 2015.

[SK01] A. Sudsang and L.E. Kavraki. A geometric approach to designing a programmable force field with a unique stable equilibrium for parts in the plane. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1079–1085, 2001.

[SK05] M. Stilman and J.J Kuffner. Navigation among movable obstacles: Real-time reasoning in complex environments. *I. J. Humanoid Robot.*, 2:479–503, 2005.

[SK11] I.A. Sucan and L.E. Kavraki. Mobile manipulation: Encoding motion planning options using task motion multigraphs. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 5492–5498, 2011.

[SK12] I. Sucan and L.E. Kavraki. A sampling-based tree planner for systems with complex dynamics. *IEEE Trans. Robot.*, 28:116–131, 2012.

[SK16] B. Siciliano and O. Khatib, editors. *Springer Handbook of Robotics*. Springer, Berlin, 2016.

[SL03] G. Sánchez and J.-C. Latombe. A single-query bi-directional probabilistic roadmap planner with lazy collision checking. In *Robotics Research*, vol. 6 of *Tracts in Advanced Robotics*, pages 403–417, Springer, Berlin, 2003.

[SL02] G. Sánchez and J.-C. Latombe. On delaying collision checking in PRM planning. *I. J. Robotics Res.*, 21:5–26, 2002.

[SLN00] T. Siméon, J.-P. Laumond, and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. *Advanced Robotics*, 14:477–493, 2000.

[Smi] R. Smits. KDL: Kinematics and Dynamics Library. `http://www.orocos.org/kdl`.

[SMK12] I. Sucan, M. Moll, and L.E. Kavraki. The open motion planning library. *Robotics & Automation Magazine*, 19(4):72–82, 2012.

[ŠO97] P. Švestka and M.H. Overmars. Motion planning for carlike robots using a probabilistic learning approach. *I. J. Robotics Res.*, 16:119–143, 1997.

[ŠO98] P. Švestka and M.H. Overmars. Coordinated pat planning for multiple robots. *Robotics Autonom. Syst.*, 23:125–152, 1998.

[SRP02] A. Sudsang, F. Rothganger, and J. Ponce. Motion planning for disc-shaped robots pushing a polygonal object in the plane. *IEEE Trans. Robot. Autom.*, 18:550–562, 2002.

[SS94] J. Snoeyink and J. Stolfi. Objects that cannot be taken apart with two hands. *Discrete Comput. Geom.*, 12:367–384, 1994.

[SSAH14] O. Salzman, D. Shaharabani, P.K. Agarwal, and D. Halperin. Sparsification of motion-planning roadmaps by edge contraction. *I. J. Robotics Res.*, 33:1711–1725, 2014.

[SSD11] M.A. Sherman, A. Seth, and S.L. Delp. Simbody: multibody dynamics for biomedical research. *Procedia IUTAM*, 2:241–261, 2011.

[SSH16a] K. Solovey, O. Salzman, and D. Halperin. Finding a needle in an exponential haystack: Discrete RRT for exploration of implicit roadmaps in multi-robot motion planning. *I. J. Robotics Res.*, 35:501–513, 2016.

[SSH16b] K. Solovey, O. Salzman, and D. Halperin. New perspective on sampling-based motion planning via random geometric graph. In *Proc. Robotics: Science and Systems*, 2016.

[SSKA07] M. Stilman, J.U. Schamburek, J.J. Kuffner, and T. Asfour. Manipulation planning among movable obstacles. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 3327–3332, 2007.

[SWO00] A.F. van der Stappen, C. Wentink, and M.H. Overmars. Computing immobilizing grasps of polygonal parts. *I. J. Robotics Res.*, 19:467–479, 2000.

[SYZH15] K. Solovey, J. Yu, O. Zamir, and D. Halperin. Motion planning for unlabeled discs with optimality guarantees. In *Proc. Robotics: Science and Systems*, 2015.

[TMMK14] M. Turpin, K. Mohta, N. Michael, and V. Kumar. Goal assignment and trajectory planning for large teams of interchangeable robots. *Auton. Robots*, 37:401–415, 2014.

[TTCA10] X. Tang, S. Thomas, P. Coleman, and N.M. Amato. Reachable distance space: Efficient sampling-based planning for spatially constrained systems. *I. J. Robotics Res.*, 29:916–934, 2010.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

[US03] C. Urmson and R.G. Simmons. Approaches for heuristically biasing RRT growth. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems*, pages 1178–1183, 2003.

[USD+13] A. Ulusoy, S.L. Smith, X.C. Ding, C. Belta, and D. Rus. Optimality and robustness in multi-robot path planning with temporal logic constraints. *I. J. Robotics Res.*, 32:889–911, 2013.

[Vár14] P.L. Várkonyi. Estimating part pose statistics with application to industrial parts feeding and shape design: New metrics, algorithms, simulation experiments and datasets. *IEEE Trans. Autom. Sci. Eng*, 11:658–667, 2014.

[VB13] C.I. Vasile and C. Belta. Sampling-based temporal logic path planning. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems*, pages 4817–4822, 2013.

[VS08] M. Vahedi and A.F. van der Stappen. Caging polygons with two and three fingers. *I. J. Robotics Res.*, 27:1308–1324, 2008.

[WB13] D.J. Webb and J.P. van den Berg. Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 5054–5061, 2013.

[WB08] N.A. Wedge and M.S. Branicky. On heavy-tailed runtimes and restarts in rapidly-exploring random trees. In *Proc. AAAI Conf. Artif. Intell.*, pages 127–133, 2008.

[WBC15] W. Wang, D. Balkcom, and A. Chakrabarti. A fast online spanner for roadmap construction. *I. J. Robotics Res.*, 34:1418–1432, 2015.

[WC15] G. Wagner and H. Choset. Subdimensional expansion for multirobot path planning. *Artif. Intell.*, 219:1–24, 2015.

[WCY15] C. Wang, Y.-J. Chiang, and C. Yap. On soft predicates in subdivision motion planning. *Comput. Geom.*, 48:589–605, 2015.

[WGPB96] J. Wiegley, K.Y. Goldberg, M.A. Peshkin, and M. Brokowski. A complete algorithm for designing passive fences to orient parts. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1133–1139, 1996.

[Wil91] G.T. Wilfong. Motion planning in the presence of movable obstacles. *Ann. Math. Artif. Intell.*, 3:131–150, 1991.

[WKLL95] R.H. Wilson, L.E. Kavraki, T. Lozano-Pérez, and J.-C. Latombe. Two-handed assembly sequencing. *I. J. Robotics Res.*, 14:335–350, 1995.

[WL94] R.H. Wilson and J.-C. Latombe. Geometric reasoning about mechanical assembly. *Artif. Intell.*, 71:371–396, 1994.

[WUB+13] T. Wongpiromsarn, A. Ulusoy, C. Belta, E. Frazzoli, and D. Rus. Incremental synthesis of control policies for heterogeneous multi-agent systems with linear temporal logic specifications. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 5011–5018, 2013.

[XBPA15] C. Xie, J.P. van den Berg, S. Patil, and P. Abbeel. Toward asymptotically optimal motion planning for kinodynamic systems using a two-point boundary value problem solver. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 4187–4194, 2015.

[YL13] J. Yu and S.M. LaValle. Structure and intractability of optimal multi-robot path planning on graphs. In *Proc. AAAI Conf. Artif. Intell.*, 2013.

[YLK01] J.H. Yakey, S.M. LaValle, and L.E. Kavraki. Randomized path planning for linkages with closed kinematic chains. *IEEE Trans. Robot. Autom.*, 17:951–958, 2001.

[YPL+10] E. Yoshida, M. Poirier, J.-P. Laumond, O. Kanoun, F. Lamiraux, R. Alami, and K. Yokoi. Pivoting based manipulation by a humanoid robot. *Auton. Robots*, 28:77–88, 2010.

[YRB00] M. Yim, J. Reich, and A.A. Berlin. Two approaches to distributed manipulation. In K.F. Böhringer and H. Choset, editors, *Distributed Manipulation*, pages 237–261, Springer, New York, 2000.

[YTEA12] H.-Y. Yeh, S.L. Thomas, D. Eppstein, and N.M. Amato. UOBPRM: A uniformly distributed obstacle-based PRM. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems*, pages 2655–2662, 2012.

[ZG96] Y. Zhuang and K.Y. Goldberg. On the existence of solutions in modular fixturing. *I. J. Robotics Res.*, 15:646–656, 1996.

[ZKM08] L. Zhang, Y.J. Kim, and D. Manocha. Efficient cell labelling and path non-existence computation using C-obstacle query. *I. J. Robotics Res.*, 27:1246–1257, 2008.