Toward Autonomous Robots: Robust, Adaptive and Dynamic Motion

John Reif* *Department of Computer Science, Duke University*

Abstract

We were engaged in research in various areas of robotic movement planning. The movement planning problems investigated were static movement problems (where the obstacles do not move), minimum distance path planning, compliant motion control (which involves planning without complete knowledge of the position of the robot) and related frictional movement problems, motion planning with independently moving obstacles, and kinodynamic movement planning (where the dynamics are taken into account including driving forces and torque), motion planning within potential fields such as gravitational force potential fields and magnetic fields and pursuit movement games (which involve movement control planning with an adversary). Most of our efforts were involved in the latter problems which have been investigated less. This paper will present some of our lates results on these various problems. Also this paper will include some improved algorithms we developed for the fundamental algebraic and combinatorial problems (such as root finding) which appear frequently as subproblems in these robot movement planning problems.

1 Introduction

Robotics movement planning problem (or *robotics motion planning problem*) is to plan an obstacle-free path (or trajectory) for a robot from a specified initial configuration (position, orientation, etc.) to a specified final configuration. It is one of the most fundamental problems in robotics research. No matter what purpose a robot is designed for, the first task is how to plan the robot to move in its working space, while avoiding possible obstacles, to reach a desired configuration.

Earlier works on robotics movement planning problem were focused on the *basic robotics movement planning problem* (known as the *piano mover's problem*[25, 26, 28] when it was first proposed), which assumed a single robot moving in a static, completely known environment without any dynamic constraint. As to the complexity aspect of this problem, Reif [20] showed that the generalized mover's problem of moving n linked polyhedra through a set of 3-D obstacles is PSPACE-hard. He proved this by reducing the computation of any reversible Turing Machine on an

*Surface address: P.O. Box 90129, Duke University, Durham, NC 27705. E-mail: reif@cs.duke.edu.

input string to an instance of the generalized mover's problem. Hopcroft, Joseph and Whitesides [15] improved this result by proving that the mover's problem for 2-D linkages is PSPACE-hard. Later the generalized mover's problem was proved to be in PSPACE by Canny [5].

Most of the works on the algorithmic approach of this problem were based on the *decomposition method* [25, 26, 27, 28], which was first suggested by Reif. Using this method, the free space is partitioned into simple connected components (cells).The relationship between two components is "adjacent" if and only if there is a feasible trajectory between two configurations, one in each component. Thus, the problem of deciding whether there is a feasible path between two configurations is transformed into a relatively simpler problem of deciding whether there is a path between two points in a discrete graph. This algorithm, although giving exact solutions to the motion planning problem, takes exponential time with respect to the number of *degree of freedom* (DOF) [1].

As the research on the robotics movement planning problem was further developed, many interesting complications of this problem were studied, including (but not limited to) the following:

- Dynamic Constraints
- Uncertainty of Control
- Incomplete Information
- Moving Obstacles
- Multiple Robots
- Presence of Friction

We are particularly interested in these complications as the assumptions of the basic movement planning problem usually are not valid in real-world applications of robotics. We used theoretical and applied techniques in these areas of robotic movement planning. In particular we contributed to the theory by providing formal models for these problems where needed and giving improved algorithms (both improved sequential and then processor efficient parallel algorithms) as well as give lower bounds on the computational complexity of the problems. We contributed to the practice by developing some prototype implementations for a selected number of our algorithms.

Section 2 of this paper summarizes some of our earlier works on various robotics movement planning problems. Section 3 gives more detailed descriptions of our latest results.

2 Summary of Earlier Results

In the area of Robotic Motion Planning, Reif has given lower and upper bounds for many diverse, fundamental problems. Reif developed various exact and approximate algorithms (see [21]) for motion of robots with constraints such as velocity

[1]A robot system is said to have k degrees of freedom if its configuration can be uniquely described by k real parameters.

or acceleration bounds. (This is a classical problem in control theory and prior to these works, there were no known provable results except for simple results in 1-D.)

Kinodynamic robot motion planning, which is the problem of finding a collision-free time-optimal trajectory for a robot whose acceleration and velocity are bounded by some norm, is of major practical application to robotic engineers, who need to move robot arms in a way that not only avoids obstacles, but also the physical constraints of the real problem as well, such as acceleration constrains due to limited force or torque from motors. This problem is particularly difficult in the often occurring coupled case (for example if the acceleration and velocity are bounded in L2 norm). For this problem, Wang and Reif [22] developed a novel approximation method, using non-uniform discretization. The approximate method gives optimal solution trajectories for the robot, up to any given small approximation error, and resulted in algorithms that are much more efficient than previously known. They also showed that the same techniques gave the first known approximation algorithm for other problems, the most notable being the curvature-constrained shortest-path problem in 3 and higher dimensions. The curvature-constrained shortest-path problem is to find a collision-free shortest-path planning problem for a point robot that moves among a set of polygonal obstacles and whose path is constrained to have a bounded curvature. Again, this problem arises frequently in practice, for example in the planning of trajectories of vehicles such as cases, with a maximum radius of curvature, and also for robot joint motors which have bounds on the maximum angular deviation.

Also, Reif gave complexity results (including both algorithms and lower bounds) for motion planning with moving obstacles [RSh94]. Furthermore, Reif gave algorithms for finding the minimal path in 2D and 3D with polygonal obstacles [RS94a], [RS94b].

These and other recent research result are summarized in the following abstracts.

2.1 Robotic Motion Planning Algorithms

2.1.1 Shortest Paths in the Plane with Polygonal Obstacles

J.H. Reif and J. Storer.

JACM, 41:5, September, 1994, pp. 982-1012.

Abstract:

We present a practical algorithm for finding minimum-length paths between points in the Euclidean plane with (not necessarily convex) polygonal obstacles. Prior to this work, the best known algorithms for finding the shortest path between two points in the plane required $O(n^2 \log n)$ time and $O(n^2)$ space, where n denotes the number of obstacle edges. Assuming that a triangulation or a Voronoi diagram for the obstacle space is provided with the input (if it is not, either one can be precomputed in $O(n \log n)$ time), we present an $O(kn)$ time algorithm, where k denotes the number of "islands" (connected components) in the obstacle space. The algorithm uses only $O(n)$ space and, given a source point s, produces an $O(n)$ size data structure such that the distance between s and any other point x in the plane (x is not necessarily an obstacle vertex or a point on an obstacle edge) can be computed in $O(1)$ time. The algorithm can also be used to compute shortest paths for the movement of a disk (so that optimal movement for arbitrary objects can be computed to the accuracy of enclosing them with the smallest possible disk).

2.1.2 A Single-Exponential Upper Bound for Finding Shortest Paths in Three Dimensions

J. Reif and J. Storer.

JACM, 41:5, September 1994, pp. 1013-1019.

Abstract:

We derive a single-exponential time upper bound for finding the shortest path between two points in a 3-dimensional Euclidean space with (nonnecessarily convex) polyhedral obstacles. Prior to this work, the best known algorithm required double-exponential time. Given that the problem is known to be PSPACE-hard, the bound we present is essentially the best (in the worst-case sense) that can reasonably be expected.

2.1.3 Approximate Kinodynamic Planning Using L2 Norm Dynamic Bounds

J. Reif and S. Tate.

Computers and Mathematics with Applications, Vol.27, No.5, 1994, pp.29-44.

Abstract:

In this paper we address the issue of kinodynamic motion planning. Given a point that moves with bounded acceleration and velocity, we wish to find the time-optimal trajectory from a start state to a goal state (a state consists of both a position and a velocity). As finding exact optimal solutions to this problem seems very hard, we present a provably good approximation algorithm using the L2 norm to bound acceleration and velocity. Our results are an extension of the earlier work of Canny, Donald, Reif, and Xavier [1], who present similar results where the dynamics bounds can be examined in each dimension independently (they use the L-infinity norm to bound acceleration and velocity).

2.1.4 Non-Uniform Discretization Approximations to Motion Planning with Dynamic Constraints

J. Reif and H. Wang.

Submitted for publication, 1995

Abstract:

In the kinodynamic motion planning problem, we wish to find a collision-free, time-optimal trajectory for a robot whose motion is governed by Newtonian dynamics and whose accelerations and velocities are bounded, due to, for example, limited force or torque from motors. This is a frequently encountered problem in robotic engineering. For this problem, we developed a novel approximation method which employs a non-uniform discretization, as opposed to a uniform one used by

previous methods. The method takes advantage of the observation that the approximation can be coarser in places that are far away from any obstacles, to reduce the size of discretization. As a consequence, the running time of our approximation is significantly reduced. The approximation method computes a trajectory whose time length is optimal up to any given small errors. The non-uniform discretization technique proves to have a wide range of applications. Applying the same technique, we obtained the first known approximation algorithm for the curvature-constrained path planning problem in 3 and higher dimensions.

2.1.5 Approximation Algorithms for Curvature-Constrained Shortest Paths

H. Wang and P. Agarwal.

Proceeding of the 7th ACM-SIAM Symposium on Discrete Algorithms (SODA'96).

Abstract:

The curvature-constrained path planning problem is to compute a collision-free shortest path, such that the curvature of the path is maximumly bounded. The curvature constraint corresponds naturally to constraints imposed by a steering mechanism on a car-like robot, because the path traced out by the middle point between the two rear wheels has a maximum curvature determined by the maximum steering angle of the car. For the 2 dimensional case with polygonal obstacles, we designed an efficient approximation method that computes a path whose length is optimal up to any given small errors. Compared to the previously best known algorithm of Jacobs and Canny, the running time of our algorithm not only improves in terms of the obstacle complexity (if n is the number of obstacle vertices, our running time is $O(n^2)$ instead of $O(n^3)$), but more importantly it does not depend on L, which is the total length of obstacle edges. This factor can be large and can not be scaled down due to the curvature constraint. We also give a stronger characterization of curvature-constrained shortest paths, which, apart from being crucial for our algorithm, is interesting in its own right.

2.1.6 Social Potential Fields: A Distributed Behavioral Control for Autonomous Robots

J. Reif and H. Wang.

The Algorithmic Foundations of Robotics (A.K.Peters, ed.), Boston, MA., 1995.

Abstract:

This paper is concerned with Very Large Scale Robotic (VLSR) systems consisting of from hundreds to perhaps tens of thousands or more autonomous robots. In the near future as the costs of robots are going down and the robots are getting more compact, more capable and more flexible, we expect to see industrial applications of VLSR systems, for example, many thousands of mobile robots performing tasks such as assembling, transporting, and cleaning within the working space of factories. The traditional "global control" mechanism is not suitable for controlling VLSR systems because of its complexity, unreliability, and inflexibility. Instead we propose a distributed control paradigm. We define simple artificial force laws between pairs of groups of robots and other components of the system. Each robot's motion is controlled by the resultant artificial force from the other robots and the other components of the system. Since the force calculations can be done in a distributed manner, the control is distributed. We show by simulations that such a simple control paradigm can yield interesting and useful cooperative behaviors among robots which achieves collision control, traffic control, and behavioral control for a VLSR system. We also develop methods for quantitatively defining behaviors of VLSR systems.

2.2 Related Work in Computational Geometry and Algebraic Algorithms

2.2.1 Work Efficient Parallel Solution of Toeplitz Systems and Polynomial GCD

J. Reif.

Proc. 27th ACM Symposium on Theory of Computing (STOC 95), Las Vegas, NV, May, 1995.

Abstract:

A matrix $A = [a_{ij}]$ is Toeplitz if $a_{i,j} = a_{i+k,j+k}$ for each k where the matrix elements are defined. We define an n x n matrix to have displacement rank delta if it can be written as the sum of delta terms, where each term is either (i) the product of a lower triangular Toeplitz matrix and an upper triangular Toeplitz matrix or (ii) the product of an upper triangular Toeplitz matrix and a lower triangular Toeplitz matrix.

There are known efficient sequential algorithms [BA80,BGY80] for inverse, determinant, linear system solution, factorization, and finding the rank for the case of Toeplitz matrices and matrices of bounded displacement rank, but there are no such results for efficient parallel algorithms. We assume the input matrices have entries that are either integers with a polynomial number of bits or rational numbers expressed as a ratio of integers with a polynomial number of bits. We make no other assumption on the input. We assume the arithmetic PRAM model of parallel computation. We assume throughout this paper that the PRAM is randomized, with a sequential source of random numbers.

In this paper, we show that certain structured linear systems can be solved exactly and efficiently in parallel, dropping these processor bounds to linear, without significant slowdown. We give much improved parallel algorithms for the exact solution and factorization, determinant, inverse, and finding rank of various structured matrices: in particular Toeplitz and matrices of bounded displacement rank. We apply this result to efficient randomized parallel algorithms for the following problems in the same parallel time $O(\log^2 n)$ and $n(\log^o megan)$ processor bounds: (1) polynomial greatest common divisors (GCD) and extended GCD, (2) polynomial resultant, (3) Pade approximants of rational functions [P1892,G72], and (4) shift register synthesis and BCH decoding problems. and with a factor of $O(\log n)$ time increase and the same processor bounds, we also solve: (1) the real root problem: finding all the roots of a polynomial with only real roots, (2) the symmetric tridiagonal eigenvalue problem: finding all the eigenvalues of a symmetric tridiagonal n x n matrix.

Previously, the best parallel algorithms [PR87,P88,Pa90] for these problems required $\Omega(\log^2 n)$ time with $n^2/\log n$ processors (or $O(\log^O(1) n)$ time using at least $\Omega(n^2/\log^O(1) n)$ processors), whereas the best sequential time was $O(n \log^2 n)$.

Our results drop by a nearly linear factor the best previous processor bounds for polylog time parallel algorithms for all these problems, and our results are within a polylog factor of work compared to the best sequential work bounds of $O(n \log^2 n)$. We are the first to give parallel algorithms for these problems with polylog time with linear processors.

We first describe our parallel algorithm for structured linear systems of bounded displacement rank which costs time $O(\log^3 n)$ using $(n \log^(\omega - 1) n)$ processors, where $\omega = 2.376$, and then observe that, while our processor reduction is by far the major result of this paper, using pipelining techniques of Reif [R94], we can also further decrease our time bounds to $O(\log^2 n)$ using $(n \log^\omega n)$ processors. We view this speedup due to pipelining as less consequential compared to our processor decrease.

All our computations require bit precision $O(n(\beta + \log n))$, which is the asymptotically optimal bit precision for $\beta >= \log n$ since the determinant, exact LU factorization and matrix inverse require bit precision at least $\Omega(nbeta)$.

2.2.2 Efficient Parallel Solution of Sparse Eigenvalue and Eigenvector Problems

John H. Reif

Proc. 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS'95), Milwaukee, WI, October 1995.

Abstract:

The problem of computing all eigenvalues and eigenvectors of a n x n matrix has many engineering and scientific applications, including vibration analysis in structures, stability analysis, etc. In a large number of applications, the matrices are symmetric and sparse with small separators. As a consequence, there is an enormous amount of literature on numerical algorithms for this symmetric sparse eigenvalue problem. The problem has an efficient sequential solution, requiring $O(n^2)$ work by use of the Sparse Lanczos method. A major remaining open question is to find a polylog time parallel algorithm with matching work bounds. Unfortunately, the sparse Lanczos method can not be parallelized to faster than time $\Omega(n)$ using n processors. All previous polylog parallel algorithms for the characteristic polynomial or all the eigenvalues of a sparse matrix required at least $\Omega(P(n))$ processors, where $P(n)$ is the processor bound to multiply two $n \times n$ matrices in $O(\log n)$ parallel time.

This paper gives a new algorithm for computing the characteristic polynomial of a symmetric sparse matrix. We derive an interesting algebraic version of Nested Dissection, which constructs a sparse factorization the matrix $A - \lambda I$ where A is the input matrix. While Nested Dissection is commonly used to minimize the fill-in in the solution of sparse linear systems, our innovation is to use the separator structure to bound also the work for manipulation of rational polynomials in the recursively

factored matrices. We compute the characteristic polynomial sparse symmetric matrix in polylog time using $O(n(n+P(s(n)))) <= O(n(n+s(n)^2.376))$ processors, where the sparsity graph of the matrix has separator size $s(n)$. Our method requires only that the matrix be symmetric and nonsingular (it need not be positive definite as usual for Nested Dissection techniques); we use perturbation methods to avoid singularities. For the frequently occurring case where the matrix has small separator size our polylog parallel algorithm requires work bounds competitive with the best known sequential algorithms (i.e. sparse Lanczos methods), for example:

(1) when the sparsity graph is a planar graph, $s(n) <= \sqrt(n)$, and we require only $n^2.188$ processors, and

(2) in the case where the input matrix is b-banded, we require only $O(nP(b)) = O(n)$ processors, for constant b.

Given the characteristic polynomial of the symmetric matrix, we can apply a known efficient parallel real root algorithm for all the eigenvalues, using $n \log^{\omega} n$ processors and further time $O(\log^2 nlogB)$ (where B is the sum of output bit precision and $n \log n$ times the input bit precision).

Our other innovation is an interesting reduction from finding all eigenvectors to the problems of back-solving a sparse linear system and multi-point polynomial evaluation. In particular, we get parallel algorithms for approximating, up to the required output bit precision, all the n eigenvectors in further parallel time $O(\log^2 n)$ time using $O(n^2)$ processors.

3 Latest Publications

3.1 The Complexity of the Two Dimensional Curvature-Constrained Shortest-Path Problem

J. Reif and H. Wang

The Proceedings of the 3rd International Workshop on Algorithmic Foundations of Robotics

The curvature-constrained shortest-path problem is to plan a path (from an initial position to a final position, where a position is defined by a location and the orientation angle) in the presence of obstacles, such that the path is the shortest among all paths with a prescribed curvature bound. This paper shows that the above problem in two dimensions is NP-hard, when the obstacles are polygons with a total of N vertices and the vertex positions are given with $N^{O(1)}$ bits. Our reduction is computed by a family of polynomial-size circuits. Previously, there is no known hardness result for the 2D curvature constrained shortest-path problem.

A *curvature-constrained path* is a path whose curvature at any point along the path is no larger than a prescribed upper bound. Curvature-constrained path planning arises in a variety of robotics problems, e.g. the motions of robot vehicles controlled by steering mechanisms. This paper studies the complexity of planning curvature-constrained paths.

To be more precise, let $P : I \to R^d$ be a d-dimensional continuous differential path parameterized by arc length $s \in I$. The *average curvature* of P in the interval

$[s_1, s_2] \subseteq I$ is defined by $\|P'(s_1) - P'(s_2)\|/|s_1 - s_2|$. A path has an upper-bounded curvature of C if its average curvature is at most C in every interval of the path.

A *position* is a tuple (p, θ), where p is a d-dimensional vector specifying the location, and θ specifies the orientation (in d-dimensions, an angle suffices). Given a curvature bound, an initial position, a final position and a set of obstacles, the *curvature-constrained shortest-path problem* is to find a path from the initial position to the final position such that the path is the shortest among all curvature-constrained paths connecting the initial position to the final position.

In the rest of the paper, without further mentioning, when we talk about paths we mean paths that satisfy the curvature constraints.

For the case of a region without obstacles, Dubins [12] characterized the shortest paths in 2D, and Sussmann [30] recently extended the characterization for shortest paths in 3D. For the case of polygonal obstacles in 2D, Fortune and Wilfong [13] gave the first algorithm which computes the exact shortest paths avoiding obstacles. However, their algorithm required super-exponential time. A polynomial-time approximation algorithm was given by Jacobs and Canny [17], whose running time is $O(N^3)$ (N is the number of obstacle vertices) and depends on the obstacle size. Their approximation algorithm was later improved by Wang and Agarwal [31] to reduce the running time to $O(N^2)$. Also the running time of their algorithm does not depend on the obstacle size. Agarwal, Raghavan, and Tamaki [2] also developed efficient approximation algorithms for the 2D shortest paths for a restricted class of obstacles (moderate obstacles). Reif and Wang [22] developed non-uniform discretization approximations for 3D kinodynamic motion planning and 3D curvature-constrained shortest paths, with considerably smaller computational complexity.

Canny and Reif [6] proved that the problem of finding a shortest 3-dimensional path, with no curvature constraints, and avoiding polygonal obstacles, is NP-hard. (Subsequently, Asano, Kirkpatrick and Yap [3] proved using similar techniques the NP-hardness of optimal motion of a rod on a plane with polygonal obstacles.) As a consequence of the above result, the curvature-constrained shortest-path problem in 3D is at least NP-hard. Although the problem in 2D is long conjectured to be a hard problem, its complexity is never stated or proved. This paper provides the first known complexity result for the curvature-constrained shortest-path problem in 2D. We show that if the number of obstacle vertices N is an input parameter and the vertex positions are given with $N^{O(1)}$ bits, this problem is NP-hard even in 2D.

Our approach to the complexity result is similar to the *path-encoding* approach developed in [6] (while the detailed techniques are quite different). This approach is to encode the boolean assignments by paths and then to reduce the 3-SAT problem to finding the shortest path. Recall that a 3-SAT formula in n variables $X_1, \ldots, X_n$ has the form

$$\bigwedge_{i=1,\ldots,m} C_i,$$

where each C_i is a clause of the form $(l_{i1} \vee l_{i2} \vee l_{i3})$. Each l_{ij} in turn is either a variable X_k or a negation of it. The 3-SAT problem is to decide whether there is a boolean assignment to $X_1, \ldots, X_n$ such that the formula is satisfied.

There are basically three steps in the path-encoding approach. First we arrange the obstacles such that there are 2^N shortest paths, where $N = 2m + n$. Each path encodes a different boolean assignment.

For a clause, we can construct the obstacles (to a "clause filter") such that the path whose encoding does not satisfy this clause is stretched a little longer than those paths with satisfying encodings. We can arrange the obstacles such that all the 2^N paths pass through all m "clause filters". The 3-SAT formula is satisfiable if and only if there exists a path whose length is not stretched more than a certain amount (the details will be in later sections).

The third and final step is to have a reverse of the obstacle arrangement so far. This will make all the 2^N paths end at the same location with the same orientation, which is the final position. From the construction, it can be shown that the total number of obstacle vertices used in the above three steps is $N^{O(1)}$.

Our techniques for generating and encoding the paths are quite different from those of [6]. In [6], 3D obstacles are used and a 3D path is represented by the obstacle edges it passes through. In our case, the paths are 2 dimensional and is represented by its orientation at certain common locations. In [6], filtering the 3 literals in a clause can be done in parallel because it is in 3D, while we have to use a more complicated encoding to emulate this in our 2D construction.

3.2 The Computational Power of Frictional Mechanical Systems

J. Reif, Z. Sun

The Proceedings of the 3rd International Workshop on Algorithmic Foundations of Robotics

We define a class of *frictional mechanical systems* consisting of rigid objects (defined by linear or quadratic surface patches) connected by frictional contact linkages between surfaces. (This class of mechanisms is similar to the Analytical Engine developed by Babbage in 1800s except that we assume frictional surfaces instead of toothed gears.) We prove that a universal Turing Machine (TM) can be simulated by a (universal) frictional mechanical system in this class. Our universal frictional mechanical system has the property that it can reach a distinguished final configuration through a sequence of legal movements if and only if the universal TM accepts the input string encoded by its initial configuration. There are two implications from this result. First, the mover's problem is undecidable when there are frictional linkages. Second, a mechanical computer can be constructed that has the computational power of any conventional electronic computer and yet has only a constant number of mechanical parts.

Previous constructions for mechanical computing devices (such as Babbage's Analytical Engine) either provided no general construction for finite state control or the control was provided by electronic devices (as was common in electro-mechanical computers such as Mark I subsequent to Turing's result). Our result seems to be the first to provide a general proof of the simulation of a universal TM via a purely mechanical mechanism.

In addition, we discuss the universal frictional mechanical system in the context of an error model that allows error up to ϵ in each mechanical operation. First, we show that for a universal TM M, a frictional mechanical system of this ϵ-error model can be constructed such that, given any space bound S, the system can simulate the computation of M on any input string ω if M decides ω in space bound S, provided that $\epsilon < 2^{-cS}$ for some constant c. Second, we show that, for any universal TM M and space bound S, if we let $\epsilon = O(S^{-1})$, there exists a frictional mechanical system in the ϵ-error model that can simulate M on any input ϵ which M decides in space bound S.

All the models of works in robotics movement planning problems mentioned earlier assume that there is no friction between objects, and most of them only allow collision-free movements so that different objects cannot even contact with each other. The only work that addressed on motion planning in the presence of friction is by Sellen[29]. He proved that dynamic motion planning problem with forbidden movements (in particular, sliding) is undecidable by showing that the actions of a TM can be realized by logical and arithmetic operations, which can be implemented by mechanical means. However, in his model, the motions of the objects corresponding to the computations of the TM can not be generated deterministically. Therefore, this model can not be used for constructing a mechanical computer.

We define the *frictional mover's problem* to be the problem of determining whether the objects in a frictional mechanical system can be moved legally from a specified initial configuration to a specified final configuration. We prove that the frictional mover's problem is undecidable by reducing the acceptance problem for Turing Machine[2] (TM), A_{TM}[3], to the frictional mover's problem. Given a universal TM [4] M, we construct a frictional mechanical system to simulate this machine. This frictional mechanical system will have the property that the objects in this system can be moved from an initial configuration, which encodes an input string ω of M, to a configuration corresponding to the accepting state of M if and only if M accepts ω. Therefore, as the acceptance problem for Turing Machine is undecidable, so is the frictional mover's problem. This implies that there is no realistic machine that can solve this problem.

An interesting property of this frictional mechanical system is that, if M accepts ω, there will be a unique simple path (i.e., one that does not repeat the same configuration) from the initial configuration to the final configuration.

Our proof will actually construct, for any given TM M, a frictional mechanical system that simulates M. Every movable object in the system is engaged or linked directly or indirectly with a "power disc" so that, when the power disc rotates, it will make those objects move accordingly. For any input string ω of M, this frictional mechanical system can be set to an initial configuration encoding ω so that, after

[2]A Turing Machine is an abstract machine with a finite state control and a tape that can store an infinite string of symbols. There is a read-write head on the tape that allows the machine to read or write the symbol at the current position of the head. The machine can write to the current position and move the head left or right according to the current state and the current symbol in a specified way.

[3]The acceptance problem A_{TM} for Turing Machine is to determine, given the description of a TM M and its input ω, whether M accepts ω.

[4]A universal TM M will take the description of any TM M' and any input string ω of M' as an input and simulate the behavior of M' presented with input ω.

the "power disc" has rotated a sufficient number of cycles, this system will result in a configuration encoding the accepting state (rejecting state) of M, if and only if M accepts (rejects) ω.

This problem can be regarded as a generalization of the *movable object problem*. The movable object problem is to ask whether a robot can move the objects amidst obstacles in a space to reach a target configuration. That is, there are two types of objects in the space: obstacles that can not be moved (or even touched) by the robot; and objects that are movable by the robot. And the goal of the robot is to either rearrange the movable objects in the space to a desired configuration or reach a target configuration of itself (or both). The first result on movable object problem was given by Wilfong [33] who studied this problem for the case of a polygonal robot moving in translation amidst polygonal movable objects in a bounded polygonal space. He proved that if the final configurations of the objects are not specified as part of the goal of the motion planning, this problem is NP-hard; otherwise, it is PSPACE-hard. He also gave one algorithm for each of the two cases where only one movable object is present.

In Wilfong's model, a robot can only grasp an object from a finite number of positions. This problem is considered to be more difficult when this number is infinite. Chen and Hwang [7] gave a heuristic algorithm to solve one model of this problem where the total weight of objects moved by a robot is to be minimized. Dacre-Wright, Laumond and Alami [9] extended Wilfong's work by providing an $O(n^3 \log n)$ algorithm for the infinite grasping position case where the final configurations of the objects are specified as part of the goal.

Compared to these previous works, our model is in 3-D and the surfaces of the objects in the space can be non-linear. Further, in addition to moving an object by "grasping" or "pushing" (directly or indirectly), a robot in our model can move objects by using the friction between it and surrounding objects. More specifically, the power disc in the frictional mechanical system can be deemed as a rather "dumb" robot, who is restricted to rotate in a specified direction without translation. And the problem is to ask whether this robot can rearrange the objects in the system to a target configuration by its rotation. In our following discussion, we will refer our problem as the "frictional mover's problem" although all the discussion applies to this generalization of the movable object problem.

We prove that a frictional mechanical system can be constructed to simulate a universal TM. Therefore, this system can be used for arbitrary finite computation, just like any conventional computer. However, the underlying assumption is that this frictional mechanical system can be constructed exactly as it is specified. We are also interested in the computational power of such a frictional mechanical system in the case inaccuracy is allowed in the construction of the mechanical devices in the system.

There are many factors that might induce error in the computations by a frictional mechanical system including the precision of manufacture of the parts. For example, the circumference of a disc may not exactly be manufactured to be a circle. The radius of a disc may not be manufactured to be exactly as it is specified. When two discs are very close but still not in contact with each other theoretically, they may already have surface contact so that the rotation of one disc will move the other one, even though they are not supposed to do so.

Since there are constant number of mechanical devices in our mechanical system, we can let ϵ to be the upper bound for the errors that occur in a single operation. This is our ϵ-error model. We prove that, given a space bound S, our frictional mechanic system in this ϵ-model can simulate the universal TM M on any input string ω that can be decided by M in space bound S, provided that $\epsilon = O(2^{-cS})$ for some constant c.

We also prove that, given a universal TM M with space bound S, there exists an $\epsilon = O(1)$ such that a frictional mechanical system in ϵ-error model can simulate the computation of M presented with any input ω if M decides ω in space bound S. This result provides decreased required precision of parts at the expense of increased number of parts, which increases with S.

The first mechanical computing machine, the Difference Engine, was designed and constructed by Charles Babbage in 1822. As electronic devices were not available in 1800's, Babbage had to exploit a purely mechanical system to build a computer. Subsequent electro-mechanical computers could exploit electronics for control, and of course so do the modern computers. Today, Babbage's concept of a purely mechanical computer would at first seem to be out of date, as computers built by much faster electronic technology prevail in every corner of the world. However, the emergence of nanotechnology provides new motivation on studies of mechanical computers.

The ultimate goal of nanotechnology [19] (see also [10], [11] and [8]) is to build microscopic computer systems at molecular or atomic scale. Such a computer might be constructed from fullerenes or diamonoids.

The presumption of nanotechnology is the capability of manipulating individual molecules and atoms. This has been the major difficulty in nanotechnology and has produced substantial skepticism over the practicability of nanotechnology. However, recent progresses in theoretical and laboratory chemistry provided considerable evidence for the possibility of manipulation of individual molecules and atoms needed for building systems at a microscopic scale. For example, Han, Globus, Jaffe and Deardorff [14] proposed molecular gears fashioned from carbon nanotubes with teeth added via a benzyne reaction. Using such technology it is conceivable to construct nano-structures that can simulate frictional surfaces.

As of today it is not known yet how to design electronic components at molecular scale. Therefore, there is a motivation to consider a nano-computer constructed by purely mechanical nano-devices, such as the fullerene gears mentioned above. There has been work done in the direction of building a nano-computer following the model of electronic computers, (i.e., building nano-components capable of data storage, switching, random access memory, etc.,) as described by Drexler [10]. It can also be envisioned that, in the far future, a programmable molecular machine could be built following the model of the frictional mechanical system described in our paper if the technology of producing nano-devices further matures.

3.3 Micro Flow Bio-Molecular Computation

Recently efforts have been made to harness the tools developed by molecular biologists in the field of Recombinant DNA (RDNA) technology to perform computation. By this means, a new type of computation has been created which has been termed bio-molecular computation (BMC). By focusing on a particular subset of operations available in the biochemistry laboratory (or operations using alternative technologies) and suitably abstracting them, one can obtain formal models for BMC.

MEMS is the technology of miniature actuators, valves, pumps, sensors and other such mechanisms. In particular, when controlling fluids it is known as micro-flow device technology. In this paper we propose bio-molecular computation (BMC) that includes MEMS technology, as this provides multiple benefits over the current techniques for BMC.

Some of the current limits of BMC stem from the labor intensive nature of the laboratory work, the large volumes needed for the desired bio-molecular reactions to occur, and the error rates. Automating the steps using available MEMS technology allows a drastic reduction in the first aspect. Controlling the location of the bio-molecular complexes allows a significant reduction of the total volume needed. Viable parallel execution of each step of an algorithm makes achieving higher reliability more practical.

In the paper we provide a model for micro-flow based bio-molecular computation (MF-BMC). It provides an abstraction for the design of algorithms which account for the constraints of the model. Our MF-BMC model uses abstractions of both the recombinant DNA (RDNA) technology as well as of the micro-flow technology and takes into account both of their limitations. For example, when considering the efficiency of the recombinant DNA operation of annealing, we take into account the limitation imposed by the concentration of the reactants. The fabrication technology used to construct MEMS is limited to constructing relatively thin 3D structures. We abstract this by limiting the model to a small constant number of layers (as is done with VLSI models).

Besides our contribution of this formal MF-BMC model, the paper contains two other classes of results. The main result is the volume and time efficient algorithm for message routing in the MF-BMC model, specifically useful for PA-Match [23]. We will show that routing of strands between chambers will occur in time $\mathcal{O}(\frac{N \cdot D}{m \cdot n})$, where N is the number of strands in the MF-BMC, n is the number of chambers where RDNA operations are occurring, D is the diameter of the topology of the layout of the chambers, and m is proportional to the channel width. Operations that need annealing, such as PA-Match, are shown feasible in $\mathcal{O}(\frac{N^2}{n^{\frac{3}{2}}})$ volume instead of the previous use of $\mathcal{O}(N^2)$ volume, with reasonable time constraints. Applications of the volume efficient algorithm include the use of the Join operation for databases, logarithmic depth solutions to SAT (Boolean formula satisfiability) problems and parallel algorithms that execute on a PRAM. Existent algorithms can be mapped to ones that work efficiently in the MF-BMC model, whereas previous methods for applications such as PRAM simulation in BMC were not both time and volume efficient. Our other class of results are theoretical lower bounds on the quantities of DNA and the time needed to solve a problem in the MF-BMC model, analogous to lower bounds in VLSI. We bound the product BT from below, and further show that BT^2 has a stronger lower bound of I^2 . Here B is the maximum amount of information encoded in the MF-BMC system at a time, T is the time for an algorithm to complete, and I is the information content of a problem.

Bio-molecular computation (BMC) is the use of biological mechanisms and structures to perform computation. Two reasons make it ripe with potential. The first is the large amount of information that could conceivably be encoded in a limited volume. The other important aspect is the available recombinant DNA (RDNA) technology that makes it possible to harness the massive parallelism of bio-chemical reactions. Performing an operation in parallel on a large quantity of information, if the implementation is perfect, has obvious benefits, such as classifying all the data into categories rapidly. However, most techniques developed and used in the field of BMC thus far have an implementation that closely matches but is not the perfect analogue of the abstraction that BMC algorithm designers use [1, 23].

The gap between reality and theory has many aspects to it and has been attacked in many different ways, such as the development of systematic techniques to reduce errors [1, 4, 18]. An aspect which this paper seeks to address is the desired and undesired interaction of reactants. In certain contexts we wish reactants to interact, and in other contexts we wish them not to do so.

A significant aspect of chemical reactions is the randomness with which they occur at the molecular level even though there is a discernible pattern at a higher level as described in kinetic models [34, 32, 24]. One of the results is that even though we can predict that a reaction between two molecules will eventually occur, the larger the volume in which the reactants exist, the less likely it is and so the longer it will take. This is a situation where we wish particular reactants to interact with each other but not with other incidental complexes as that merely increases the overall time before the desired reaction is completed. In such a situation if we were able to place molecules between which an interaction was desired, into a smaller volume then they would interact much more quickly with high likelihood.

Conversely, there are certain reactants that we would prefer to isolate. While the ability to perform operations in parallel on the entire data set has advantages, it brings with it the drawback that reagents added to execute an operation necessarily interact with all the bio-molecular material present, even that which it does not need to. In addition, the presence of extraneous material increases the chance of partial matching and other such incomplete interactions occurring, causing avoidable errors. In such cases, we would like to be able to separate the data set where possible.

Living systems have evolved a simple method to effect some control in such situations. We refer here to the division of an organism into its many constituent cells within which independent bio-chemical reactions occur. Through the use of a circulatory system, materials are routed to particular cells which perform different operations in parallel on the material. In addition the material can be processed and then sent along to another destination through the infrastructure of the circulatory system. By having limited volumes in each cell, the reactants that need to interact can do so efficiently. Certain highly reactive chemicals which have positive uses but can have a detrimental effect on other reactants and reactions can be isolated within particular cells, where they can be productively used without affecting other material.

Analogously, we propose a framework in which BMC can occur in a large number of isolated regions of limited volume. As with a living system, there will be an associated infrastructure that will allow reactants, reagents and associated material to be transported from one region to another. Since our method will make use of micro-flow device technology, we refer to the framework as micro-flow bio-molecular computation (MF-BMC). A 3D MEMS fluid system has been proposed in a different context [16] but does not examine the BMC potential in any detail.

4 Publications

[ARW95] Approximation Algorithms for Shortest Paths with Bounded Curvatures in 2 and Higher Dimensions, P. Agarwal, J. Reif and H. Wang. Submitted for publication, 1995.

Journal Articles: ——————

[RS94a] Shortest Paths in the plane with polygonal obstacles, J.H. Reif and J.A. Storer. JACM 41:5, September, 1994, pp.982-1012.

[RS94b] A Single-Exponential Upper Bound for Finding Shortest Paths in Three Dimensions, J.H. Reif and J. Storer. JACM 41:5, September 1994, pp.1013-1019.

[RT94] Approximate Kinodynamic Planning Using L2-norm Dynamic Bounds, J. Reif and S. Tate. Computers and Mathematics with Applications, Vol.27, No.5, 1994, pp.29-44.

[RW95b] Social Potential Fields: A Distributed Behavioral Control for Autonomous Robots, J. Reif and H. Wang. Proc. Workshop on Algorithmic Foundations of Robotics (WAFR'94), San Francisco, California, February, 1994. Published in "The Algorithmic Foundations of Robotics", A.K.Peters, Boston, MA,1995.

Conference Articles: ———————

[RW95a] Non-Uniform Discretization Approximations to Motion Planning with Dynamic Constraints, J. Reif and H. Wang. Proceedings of the 2nd Workshop of the Algorithmic Foundations of Robotics, 1996

[WA95] Approximation Algorithms for Curvature-Constrained Shortest Paths, H. Wang and P. Agarwal. To appear in the Proceeding of the 7th ACM-SIAM Symposium on Discrete Algorithms (SODA'96).

[Rei95] Work Efficient Parallel Solution of Toeplitz Systems and Polynomial GCD, J. Reif. Proc. of the 27th ACM Symposium on Theory of Computing (STOC 95), Las Vegas, NV, May 29-June 1, 1995.

[Rei95] Efficient Parallel Solution of Sparse Eigenvalue and Eigenvector Problems, J. Reif. Proc. of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS'95) Milwaukee, WI, October 23-25, 1995.

[Rei95] Parallel Molecular Computation: Models and Simulations, J. Reif. Proc. 7th Annual

ACM Symposium on Parallel Algorithms and Architectures (SPAA'95) Santa Barbara, CA, July 1995.

5 Acknowlegdments

This work was upported by grants NSF CCR-9725021, NSF IRI-9619647, NSF/DARPA CCR-92725021, and ARO contract DAAH-04-96-1-044.

References

[1] Leonard Adleman. On constructing a molecular computer. Available via anonymous FTP from the University of Southern California Department of Computer Science, 1995.

[2] P. K. Agarwal, P. Raghavan, and H. Tamaki. Motion planning for a steering-constrained robot through moderate obstacles. In *Proceedings of the 27th Annual ACM Symposium on the Theory of Computing (STOC '95)*, pages 343–352, New York, May 1995. ACM.

[3] T. Asano, D. Kirkpatrick, and C. K. Yap. d_1-optimal motion for a rod. In *Proceedings of the Twelfth Annual Symposium On Computational Geometry (ISG '96)*, pages 252–263, New York, May 1996. ACM Press.

[4] Dan Boneh and Richard Lipton. Making dna computers error resistant. In *Proceedings of the Second Annual DIMACS Meeting on DNA Based Computers*, 1996.

[5] J. Canny. Some algebraic and geometric computations in PSPACE. In Richard Cole, editor, *Proceedings of the 20th Annual ACM Symposium on the Theory of Computing*, pages 460–467, Chicago, IL, May 1988. ACM Press.

[6] J. Canny and J. Reif. New lower bound techniques for robot motion planning problems. In Ashok K. Chandra, editor, *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, pages 49–60, Los Angeles, CA, October 1987. IEEE Computer Society Press.

[7] PC Chen and YK Hwang. Motion planning for a robot and a movable object amidst polygonal obstacles. In *Proceedings: 1991 IEEE International Conference on Robotics And Automation*, volume 1, pages 444–449, Sacramento, CA, USA, 7–12 April 1991. IEEE Computer Society Press.

[8] BC Crandall. *Nanotechnology: molecular speculations on global abundance.* MIT Press, 1996.

[9] B. Dacre-Wright, J.-P. Laumond, and R. Alami. Motion planning for a robot and a movable object amidst polygonal obstacles. In *Proceedings: 1992 IEEE International Conference on Robotics And Automation*, volume 3, pages 2474–2480, Nice, France, 12–14 May 1992. IEEE Computer Society Press.

[10] K. E. Drexler. *Nanosystems: molecular machinery, manufacturing, and computation.* Wiley Interscience, 1992.

[11] K. E. Drexler, C. Peterson, and G. Pergamit. *Unbounding the Future: the Nanotechnology Revolution.* William Morrow and Company, Inc., 1991.

[12] L. E. Dubins. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *Amer. J. Math.*, 79:497–516, 1957.

[13] S. J. Fortune and G. Wilfong. Planning constrained motion. *Annals of Mathematics and Artificial Intelligence*, 3:21–82, 1991.

[14] J. Han, A Globus, R. Jaffe, and G. Deardorff. Molecular dynamics simulation of carbon nanotube based gears. *Nanotechnology*, 8(3):95–102, September 1997.

[15] J. E. Hopcroft, D. A. Joseph, and S. H. Whitesides. Movement problems for 2-dimensional linkages. *SIAM Journal on Computing*, 13(3):610–629, 1984.

[16] Koji Ikuta. 3d micro integrated fluid system toward living lsi. In *International Workshop on Artificial Life*, 1996.

[17] P. Jacobs and J. Canny. Planning smooth paths for mobile robots. In Z. Li and J. Canny, editors, *Nonholonomic Motion Planning*, pages 271–342. Kluwer Academic Publishers, 1992.

[18] Richard Karp, Claire Kenyon, and Orli Waarts. Error-resilient dna computation. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, 1996.

[19] M. Krummenacker and J. Lewis. *Prospects in Nanotechnology: Toward Molecular Manufacturing.* Wiley, 1995.

[20] J. Reif. Complexity of the mover's problem and generalizations. In *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science*, pages 421–427, San Juan, Puerto Rico, October 29–31 1979. IEEE, IEEE Computer Society Press.

[21] J. Reif and S. Tate. Approximate kinodynamic planning using l2-norm dynamic bounds. *Computers and Mathematics with Applications*, 27(5):29–44, 1995.

[22] J. Reif and H. Wang. Non-uniform discretization approximations for kinodynamic motion planning and its applications. In *Proceedings of the 2nd Workshop of the Algorithmic Foundations of Robotics*, 1996.

[23] John Reif. Parallel molecular computation: Models and simulations. *Algorithmica, Special Issue on Computational Biology*, 1998.

[24] John SantaLucia, Jr., Hatim Allawi, and P. Ananda Seneviratne. Improved nearest-neighbour parameters for predicting dna duplex stability. *Biochemistry*, 1996.

[25] J. T. Schwartz and M. Sharir. On the piano movers problem: I. the case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Comm. Pure Appl. Math.*, 36:345–398, 1983.

[26] J. T. Schwartz and M. Sharir. On the piano movers problem: II. general techniques for computing topological properties of real algebraic manifolds. *Advances in applied mathematics*, 4:298–351, 1983.

[27] J. T. Schwartz and M. Sharir. On the piano movers problem: III. coordinating the motion of several independent bodies: The special case of circular bodies moving amist polygonal barriers. *The International Journal of Robotics Research*, 2(3):46–75, 1983.

[28] J. T. Schwartz and M. Sharir. On the piano movers problem: V. the case of a rod moving in three dimensional space amidst polyhedral obstacles. Technical Report 083 R13, New York University, Courant Institute, Department of Computer Sciences, July 1983.

[29] J. Sellen. Lower bounds for geometrical and physical problems. *SIAM Journal on Computing*, 25(6):1231–1253, December 1996.

[30] H. J. Sussmann. Shortest 3-dimensional paths with a prescribed curvature bound. In *Proceedings of the 34th Conference on Decision and Control*, pages 3306–3312, 1995.

[31] H. Wang and P. K. Agarwal. Approximation algorithms for curvature-constrained shortest paths. In *Proceedings of the 7th ACM-SIAM Symposium on Discrete Algorithms*, pages 409–418, 1996.

[32] James Wetmur. Physical chemistry of nucleic acid hybridization. In *Proceedings of the Third Annual DIMACS Meeting on DNA Based Computers*, 1997.

[33] G. Wilfong. Motion planning in the presence of movable obstacles. In *Proceedings of the Fourth Annual Symposium on Computational Geometry (Urbana-Champaign, IL, June 6–8, 1988)*, pages 279–288, New York, 1988. ACM, ACM Press.

[34] Erik Winfree. Simulations of computing by self-assembly. In *Proceedings of the Third DIMACS Meeting on DNA Based Computing*, 1998.