2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

# An Optimization Approach to Planning for Mobile Manipulation

Dmitry Berenson James Kuffner Howie Choset

*The Robotics Institute*
*Carnegie Mellon University*
*5000 Forbes Ave., Pittsburgh, PA, 15213, USA*
[dberenso, kuffner, choset]@cs.cmu.edu

***Abstract*— We present an optimization-based approach to grasping and path planning for mobile manipulators. We focus on pick-and-place operations, where a given object must be moved from its start configuration to its goal configuration by the robot. Given only the start and goal configurations of the object and a model of the robot and scene, our algorithm finds a grasp and a trajectory for the robot that will bring the object to its goal configuration. The algorithm consists of two phases: optimization and planning. In the optimization phase, the optimal robot configurations and grasp are found for the object in its start and goal configurations using a co-evolutionary algorithm. In the planning phase, a path is found connecting the two robot configurations found by the optimization phase using Rapidly-Exploring Random Trees (RRTs). We benchmark our algorithm and demonstrate it on a 10 DOF mobile manipulator performing complex pick-and-place tasks in simulation.**

## I. INTRODUCTION

Motion planning for a mobile manipulator is a complex task, involving multiple levels of planning which are often divided into sub-problems to manage complexity. Consider a pick-and place operation for a mobile manipulator: the task is to move an object from some given starting configuration to some given goal configuration. The problem can be broken down into three sub-problems: 1) move the robot from its initial configuration to a configuration where it is near the object, 2) grasp the object, 3) move the robot (holding the object) to some configuration which places the object into its goal configuration. Breaking the problem into the above subproblems reduces the complexity of the overall task by allowing sub-plans to be generated in series. However, ignoring the coupling between sub-problems can turn feasible problems into infeasible ones and introduce unnecessary difficulty for the path planning algorithm. In this paper, we will show the importance of coupling the sub-problems and present an algorithm which takes this coupling into account. Our algorithm focuses on two issues central to mobile manipulation: choosing the optimal grasp and choosing the optimal locations for the mobile base.

To illustrate the importance of picking the correct grasp, consider the canonical problem of loading a dishwasher with a mobile manipulator. Suppose we wish to place a wineglass into one of the dishwasher's bins. Suppose also that we have two stable grasps for the wineglass: a power grasp of the body and a pinch at the stem. If we only consider grasp quality, we are likely to choose the power grasp because it is usually more stable than a pinch. However, as shown in Fig. 1(a), it is impossible to place the glass into the goal configuration while using the power grasp because of collisions between the fingers and the environment. Considering the feasible grasp at the goal configuration of the wineglass is essential to the completion of this task.

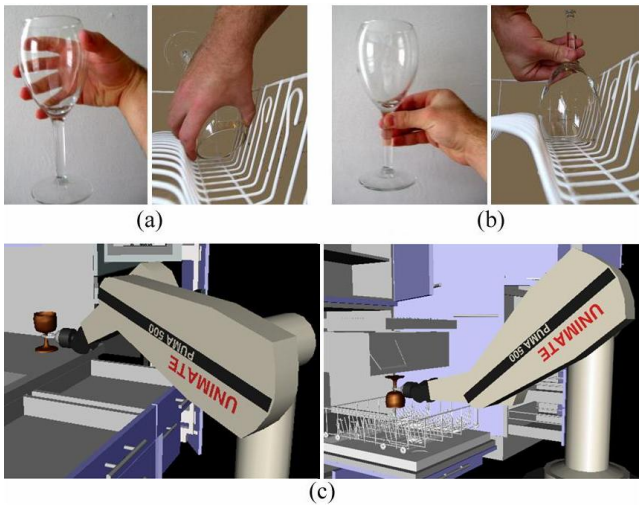


Fig. 1. An example of the importance of proper grasp selection in the wineglass problem. (a) If the glass is grasped in the initial state using a power grasp, the goal configuration of the wineglass is unachievable without collision between the hand and environment. (b) If a pinch grasp is used, the goal configuration for the wineglass can be achieved without collision. (c) The Puma arm using the correct grasp for the wineglass as determined by our algorithm.

Much work has been done on finding a suitable metric for grasp quality [2] [3] [4]. Our approach introduces a task-dependent component to the problem of grasp selection. Instead of greedily choosing the best grasp at the initial configuration of the object based solely on grasp quality, our algorithm considers the object in both its starting and goal configurations as well as the placement of the robot's base.

The placement of the mobile base of the manipulator is crucial to the success of the path planning algorithm. An incorrect placement of the base could make the robot unable to reach the object. If the robot is able to reach the object, it may do so in an undesirable way, such as placing itself near a singular configuration where controlling the arm is difficult. The placement of the base in the initial and final configurations also determines the difficulty of the path

planning subproblem. If we choose to place the base in a very cluttered area, it may be more difficult or impossible to escape, whereas if we place the base in an open area, we will likely have an easier time planning a path through the environment. Base placement also influences the configuration of the the rest of the mobile manipulator through inverse kinematics. Finally, an improper base placement could place the arm into collision when grasping the object.

In this paper, we address the issues of grasp selection and base placement from the perspective of optimization. In the following sections we discuss related work in mobile manipulation, formally define the pick-and-palce problem, and present three metrics for evaluating the quality of a robot configuration: grasp quality, configuration desirability, and configuration clutter. These metrics are evaluated at each candidate configuration where the robot is grasping the object—thus giving the configuration an overall score. We show how to use a co-evolutionary algorithm to search a constrained space of base placements and grasps for the optimal (as defined by the above score) robot configurations for the start and goal configurations of the object we wish to move. Then we show how to use Bidirectional RRTs [1] to find a path linking the the start and goal robot configurations found by optimization. Finally, we benchmark our optimization method against random sampling and demonstrate our algorithm in simulation on a 10 DOF mobile manipulator.

## II. RELATED WORK

In previous work, mobile manipulation planning has been divided into four main areas: path planning for robot arm(s), navigation planning for the base of the robot, grasping, and frameworks for general manipulation planning. Our contribution is the integration of path planning for arms, navigation, and grasping into a single optimization-based algorithm which can be used as a module for general manipulation planning frameworks.

Lozano-Perez et al. [10] were one of the first to examine integrated grasping and planning for a fixed-base arm. They present a framework that recognizes objects, chooses grasps, and plans arm motions. We revist the same problem in this paper with more modern algorithms that consider a mobile base and optimality concerns when choosing grasps. Koga et al. [11] examine the problem of manipulation planning for one or more fixed-base arms by planning a path for the object and then using inverse kinematics to find the necessary grasps and arm configurations at each point in the object's trajectory. Unfortunately, inverse kinematics cannot determine the necessary base positions for a mobile manipulator, much less determine base positions that will be collision-free. Hsu et al. [20] study the problem of where to place a fixed base manipulator in a factory environment for optimal task execution.

Zhao et al. [5] use a genetic algorithm to plan a path for the mobile base between a discrete set of feasible base-placements. The set of placements is determined by exhaustive search but obstacles and grasping are not considered. Chen and Zalzala [6] use a genetic algorithm to plan a path through a gridded environment using distance to obstacles as one of the criteria for optimization. Vannoy and Xiao [7] use a genetic algorithm to create a population of trajectories for a mobile manipulator that allow it to avoid dynamic obstacles while maintaining good manipulability. Similarly, our approach considers both distance to obstacles and manipulability when determining the fitness of a robot configuration.

Li and Sastry [4] define a grasp-quality metric that is task-dependent, based on the types of motions one wishes to perform with the object, but do not consider the geometric environment inhabited by the robot and the coupling between a grasp and robot arm configuration. However, this metric can be used as one of the configuration quality metrics in our configuration scoring function (see Methods).

Stilman et al. [15] and Cambon et al. [14] both outline manipulation planning frameworks. Stilman et al. examine the problem of manipulation planning among movable obstacles. Here the object is initially inaccessible and movable obstacles must be displaced in order to pick up the object and place it in its goal configuration. Cambon et al. have developed a motion planning framework where high-level action plans and low-level path plans are searched in parallel. The high level plans can involve regrasping and moving obstructing obstacles out of the way. Our algorithm can be integrated with these schemes to choose the optimal base-placements and grasps for all low-level path plans. Likewise if the object must be slid through a narrow space as in [13] and [12].

Other issues in mobile manipulation have also been studied: Seraji [18] and Yamamoto and Yun [17] explore issues in control for mobile manipulators, and Nagatani et al. [16] show a control approach to the task of door opening. Petrovskaya and Ng [19] show how to integrate localization and object detection for navigation and manipulation. Brock and Khatib [21] explore the problem of following an end-effector trajectory while avoiding obstacles in a dynamic environment.

## III. PROBLEM DEFINITION

We formally define the problem of pick-and-place mobile manipulation as follows:

Define the following configurations:

- $qR$, the full configuration of the robot, including the base and the arm
- $qArm \subset qR$, a configuration of the robot's arm joints excluding the joints of the hand
- $qBase \subset qR$, a configuration of the robot's base (usually in the XY plane)
- $qO$, a configuration of the object (usually 6 DOF)
- $qO_{start}$, the starting configuration of the object
- $qO_{goal}$, the goal configuration of the object
- $qR_{start}$, the starting configuration of the robot
- $qR_{Ostart}$, a collision-free configuration of the robot where the robot is grasping the object in the object's starting configuration

- $qR_{Ogoal}$, a collision-free configuration of the robot where the robot is grasping the object in the object's goal configuration

Also define $Grasp$ as a set of parameters tailored to the robot's manipulator and/or the object. The only restriction on the definition of $Grasp$ is that the parameters must somehow determine a workspace position for the robot's end-effector. For instance, a $Grasp$ could be defined by a center point and orientation for a gripper. Please see the Experiments section for a description of the $Grasp$ definition we use.

Our algorithm requires $qO_{start}$, $qO_{goal}$, and $qR_{start}$ as input. The problem is to find a collision-free path for the robot and object which takes the object from $qO_{start}$ to $qO_{goal}$.

We do *not* assume that the robot starts in a configuration where it is grasping the object. $qR_{Ostart}$ and $qR_{Ogoal}$ are initially unknown. In our experiments we assume no constraints on $qBase$, however such constraints can be incorporated into the path planning phase [1] if needed.

We also require an Inverse Kinematics (IK) algorithm for the given robot. In our context the IK algorithm need not find values for all DOF of the robot, only $qArm$. To illustrate, consider an arm mounted on a mobile base. In this case the IK algorithm would determine $qArm$ given $qBase$ and an end effector position determined by some $Grasp$ as arguments. The IK algorithm is robot-specific and can be iterative or analytical.

## IV. METHODS

Our algorithm consists of two-phases: an optimization phase and a planning phase. In the optimization phase, we use a co-evolutionary algorithm to find the optimal $qBase \subset qR_{Ostart}$, $qBase \subset qR_{Ogoal}$, and $Grasp$. After the search, we extract the accompanying $qArm$ values using IK, thus fully defining $qR_{Ostart}$ and $qR_{Ogoal}$. In the path planning phase, we use bidirectional RRTs to find a path connecting $qR_{Ostart}$ and $qR_{Ogoal}$.

### A. Scoring Function

In order to find the optimal grasp and base placement at both the start and goal configurations of the object we need a scoring function that judges the quality of a given robot configuration. Let $qR = IK(qBase, Grasp)$ be a configuration we wish to evaluate. If $qR$ is in collision it receives a score of 0. Otherwise, we consider three criteria to compute the score: *grasp quality, configuration desirability, and configuration clutter*. *Grasp quality* is often defined as force-closure [9], but can be measured with a variety of metrics [3] [4] [2]. For our purposes it will suffice that the there exists some grasp-quality metric that returns a single number as a measure of grasp quality, $G$, given $qR$. Note that if $Grasp$ is defined in such a way that all possible values of $Grasp$ parameters will yield desirable grasps, this measure is unnecessary.

*Configuration desirability* is robot-specific, referring to the cost of being in a certain configuration of the arm. If we assume that the robot's arm can be easily controlled in any feasible configuration, there is no need to consider configuration desirability. Unfortunately, an arm's configuration space often contains singular configurations which are difficult to deal with in control. The manipulability measure is useful for gauging the desirability of a configuration because it gives better scores to configurations that are farther from singular configurations. Thus we use manipulability as our configuration desirability score, calculated as follows:

$$M = min(eigs(JJ^T)) \tag{1}$$

where $M$ is the manipulability score for a certain robot configuration, $J$ is the Jacobian of the robot's arm evaluated at $qArm \subset qR$, and $eigs()$ is a function that returns the eigenvalues of a matrix.

*Configuration clutter* measures the proximity of a given robot configuration to nearby obstacles. For path planning and safety purposes, it is desirable to choose the goal configuration for the robot that is not in a cluttered area, if possible. To measure the clutter in the vicinity of the robot, we compute the distance from each of the robot's links to the nearest obstacle. This distance, $d_l$, is computed for each link, $l$, and the total configuration clutter is given by:

$$C = \sum w_l d_l \tag{2}$$

where $w_l$ is a weighting constant for the link $l$. We use the inverse of the distance between $l$ and the robot base's center of mass as our $w_l$.

After calculating the three metrics, $G$, $M$, and $C$, we weight and sum them into an overall score for the configuration, $Score(qBase, Grasp)$. If $qR$ is in collision or does not meet the minimum requirements for grasp stability (as determined by the grasp quality metric), $Score$ is set to 0. It is important to note that these metrics are not the only ones that can be used, they can be replaced by other metrics or other metrics can be used in conjunction with them. All that is required is a function that returns an $Score$ when given a base position and grasp.

### B. Optimization

Environments with clutter and objects which can be grasped in multiple ways make the problem of finding the optimal base positions and grasp highly non-linear with many local minima. The difficulty of the problem necessitates an optimization approach that can efficiently avoid these local minima. We use a co-evolutionary algorithm (see [8] for a description of evolutionary algorithms) to search for the optimal base positions and grasp.

The evolutionary structure is set up as follows: there are three populations, a population of $qBase \subset qR_{Ostart}$ individuals, a population of $qBase \subset qR_{Ogoal}$ individuals, and a population of $Grasp$ individuals. The base position individuals are specified in polar coordinates, with the origin of the coordinate system placed at the center of mass of the object in configurations $qO_{start}$ and $qO_{goal}$, respectively. The individuals also contain genes for an offset in X and

Y limited to $\pm$20cm. These offset genes allow the co-evolutionary algorithm to improve on good solutions in later generations via mutation.

Each population is associated with its own fitness function, however the fitness of one population depends on the individuals in one or more of the other populations (this is what makes the algorithm *co*-evolutionary). The base position populations use the following fitness function:

$$F_{Base}(qBase) = \max_{0 \leq i < k} Score(qBase, Grasp_i) \quad (3)$$

$i$ indexes over the individuals in the $Grasp$ population in order of *their* fitness. $k$ determines how many $Grasp$ individuals are to be evaluated[1]. The fitness of $Grasp$ individuals is determined by

$$F_{Grasp}(Grasp) = \sum_{i=0}^{k} Score(qBase_i \subset qR_{Ostart}, Grasp) + Score(qBase_i \subset qR_{Ogoal}, Grasp)$$

In this equation, $i$ indexes over the individuals in the base position populations in order of their respective fitnesses. We also include a special provision for this fitness function: if *exactly one* of the scoring functions returns 0, (i.e. the resulting configuration is in collision), $F_{Grasp}(Grasp)$ is set to 1.0. This provision ensures that grasps which are valid in exactly one of the start/goal configurations of the base are preserved but not given any advantage over other such grasps. This is important for maintaining a pool of grasps that will eventually be successful in the $Grasp$ population without allowing one such grasp to dominate.

The effect of the fitness functions is to guide all three populations to a solution that has maximum score. Each base position population "pulls" the grasp population toward what the best base position individuals "want" via the $F_{Grasp}$ fitness function. Likewise, the $Grasp$ population "pulls" both of the base position populations toward what its best individuals want via the $F_{Base}$ fitness function. Thus the most successful individuals in the $Grasp$ population will be the ones which are preferred by *both* base position populations and the most successful individuals in the base position populations will be the ones which are preferred by the $Grasp$ population. See Figure 2.

One cycle of evolution is defined as evolving each of the base position populations for one generation and then evolving the grasp population for one generation. For each generation in the evolution of all three populations, the parents are the top 50% of the population. Children are generated from these parents by randomly choosing two parents, performing two-point crossover, and then mutating the resulting genomes with 20% mutation probability for each gene. Mutation adds a random value to the value of the mutated gene. This random value is uniformly distributed between $\pm 1/4$th of the gene's range.

[1]We use $k = 7$ in our experiments

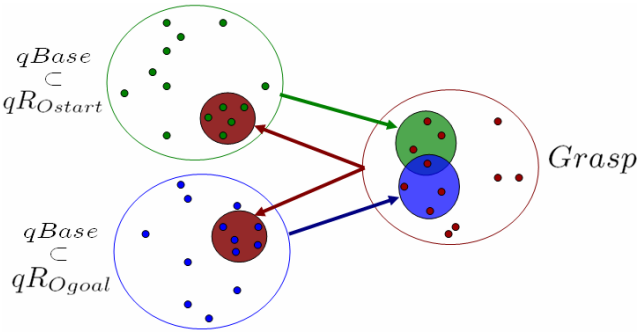


Fig. 2. Depiction of the three populations. The best individuals in the $qBase \subset qR_{Ostart}$ population prefer the grasps in the green circle. The best individuals in the $qBase \subset qR_{Ogoal}$ population prefer the grasps in the blue circle. The best individuals in the $Grasp$ population prefer the base positions in the red circles. The grasp in the intersection of the green and blue circles is preferred by both base position populations and will receive the highest fitness in the $Grasp$ population.

To initialize the base position populations, we need to sample positions from regions of the configuration space that are likely to yield IK solutions. We define an annulus around the object with inner radius equal to the radius of the robot's base and an outer radius equal to the distance from the robot's base to its end effector when the arm is fully extended. Individuals in the initial start and goal base position populations are sampled from annuluses centered at the center of mass of the object in $qO_{start}$ and $qO_{goal}$, respectively. Each sample is tested for collision between the base and environment obstacles and any samples in collision are rejected and re-sampled.

After running the co-evolutionary algorithm, we extract the best $Grasp$ in the population of grasps and the best $qBase \subset qR_{Ostart}$ and $qBase \subset qR_{Ogoal}$ for that grasp in the base position populations. We then compute the IK and arrive at the fully specified $qR_{Ostart}$ and $qR_{Ogoal}$ configurations.

### C. Path Planning

Once we have determined $qR_{Ostart}$ and $qR_{Ogoal}$, we plan a path for the robot and object to move the object from $qO_{start}$ to $qO_{goal}$. We divide this task into two intuitive subtasks, moving the robot from $qR_{start}$ to $qR_{Ostart}$ and then moving the robot and object from $qR_{Ostart}$ to $qR_{Ogoal}$. We know that moving the robot to $qR_{Ogoal}$ while grasping the object must place the object into $qO_{goal}$ because of the way we found $qR_{Ogoal}$. We use a Bidirectional RRT to plan these two paths, which is very effective for mobile manipulators.

## V. EXPERIMENTS

We conducted several experiments with a Puma robot on a cylindrical mobile base in simulation. The robot has 10 DOF: 6 DOF in the arm, 2 DOF in the hand, and 2 DOF of translation (X and Y) of the base. We define $qBase$ to be the X and Y translation and define $qArm$ to be the DOF of the arm. We consider two pick-and-place problems: moving a wineglass from a kitchen counter into a dishwasher and moving a plate from the dishwasher onto a cabinet shelf (see Figure 3).

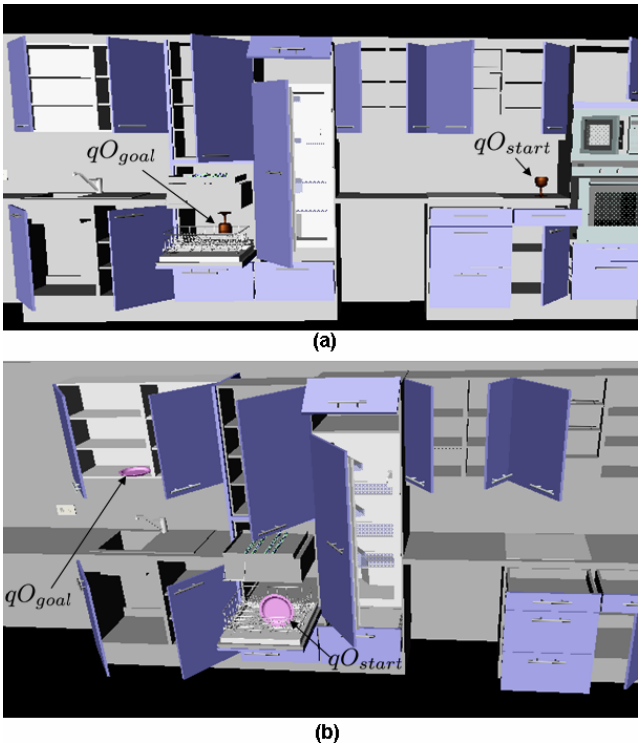


Fig. 3. Start and goal positions for the object to be moved in the wineglass problem(a) and plate problem (b).

Currently, we must parameterize the $Grasp$ differently for each object that is to be grasped. For the wineglass problem, the $Grasp$ was parameterized in terms of end effector orientation and target point on the wineglass' principle axis. The target point specifies where the center of the Puma's gripper should be placed. The end effector orientation was restricted to be perpendicular to the principle axis of the wineglass. Thus the grasp is defined using two variables: 1 for the location of the target point along the principle axis and 1 for the approach direction.

The $Grasp$ for the plate was defined similarly. The target point for the gripper is allowed to lie on the outer lip of the plate, with the approach direction allowed to lie in the plane parallel to the plate. Note that in both problems, all grasps will yield force closure, so grasp-quality is not evaluated.

To gauge our co-evolutionary optimization algorithm, we performed a benchmark test against random sampling of base placement and grasp parameters. Co-evolution was run for 6 cycles with population sizes of 84 for the base position populations and 64 for the grasp population. Each run entailed 6940 scoring function evaluations. To be fair, the random sampling was restricted to base positions within the same annulus as the co-evolutionary algorithm and both random sampling and co-evolution were allowed the same number of scoring function evaluations. Both problems were run 200 times for both random sampling and co-evolution, the statistics are shown in Table I. Both algorithms were able to find a feasible solution in an equal number of runs. However, co-evolution clearly outperforms random sampling

| | Percent Success | Avg. Score (100 max) | Score Std. Dev. |
|---|---|---|---|
| **Wineglass Problem** | | | |
| Co-Evolution | 99.0% | 91.74 | 3.77 |
| Random Sampling | 99.0% | 72.57 | 6.13 |
| **Plate Problem** | | | |
| Co Evolution | 100% | 90.18 | 3.94 |
| Random Sampling | 100% | 78.06 | 4.52 |

TABLE I

RESULTS OF 200 RUNS OF THE WINEGLASS AND PLATE PROBLEMS

in terms of both average score[2] and consistency (standard deviation).

The runtime of optimization depends almost entirely on the time needed for scoring function evaluation. Since the models we used were extremely detailed, composed of complex shapes with hundreds of thousands of triangles, evaluating distance from the robot's links to the environment was quite slow. The distance evaluation time depended on the configuration of the robot and its proximity to obstacles. This is because our distance check used the PQP collision library, which uses oriented bounding boxes to prune the search for the shortest distance. The analytical IK computation for the Puma was fast, however, if there was more than one IK solution for a given base/end effector pair, the scoring function evaluated all the solutions and took the best. Thus it is difficult to give a runtime for one evaluation of the scoring function. In total, co-evolution took an average of 149 and 133 seconds on a 3.0 GHz Pentium 4 with 1GB of memory for the wineglass and plate problems, respectively. Random sampling took an average of 97.4 and 112 seconds on the same computer for the wineglass and plate problems, respectively. While the difference between co-evolution and random sampling runtimes may seem large, it is important to note that random sampling spends most of its time evaluating solutions that result in collision, which are quickly rejected, while co-evolution spends most of its time evaluating high-scoring solutions, which takes more time.

After generating the $qR_{Ostart}$ and $qR_{Ogoal}$ configurations using the co-evolutionary algorithm, we plan a path from $qR_{start}$ to $qR_{Ostart}$. We then grasp the object by closing the gripper at the target point. After the grasp is complete, we compute a plan to move from $qR_{Ostart}$ to $qR_{Ogoal}$ with the exception that the fingers are not moved so that the grasp is maintained. Planning these two trajectories took roughly 30 seconds on the aforementioned computer, which is reasonable considering the complexity of the models used and the clutter of the space. See Figure 1 and Figure 4 for an example of the $qR_{Ostart}$ and $qR_{Ogoal}$ found for the wineglass and plate problems, respectively. See Figure 5 for several snapshots from the execution of the plan for the wineglass problem.

## VI. CONCLUSION AND FUTURE WORK

We have presented an optimization-based approach to grasping and path planning for mobile manipulators per-

[2]Note, the scores are normalized to 100 by the best score found after hundreds of runs of each problem.

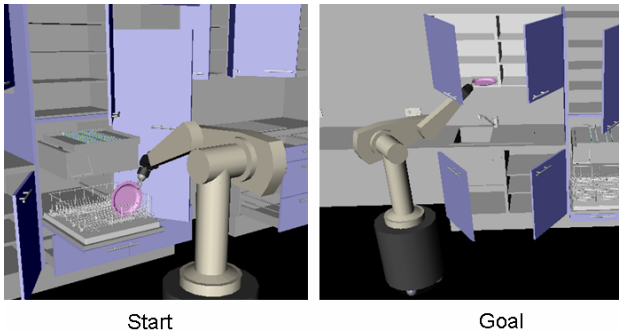


Fig. 4. An example of the $qR_{Ostart}$ and $qR_{Ogoal}$ found by the optimization phase for the plate problem. Note that in the goal configuration the optimizer prefers to keep the base away from the lower cabinet doors more than it prefers to keep the arm away from the upper cabinet doors. This is because the distance from the *base* link to the nearest obstacle is weighted higher than the distance from the *forearm* link to the nearest obstacle in our scoring function.

forming pick-and-place tasks. The algorithm consists of two phases: optimization and planning. In the optimization phase, we use a co-evolutionary algorithm to find the optimal robot configurations and grasp for the object in its start and goal configurations. In the planning phase, a path connecting the two robot configurations determined by optimization is found using bidirectional RRTs. Our optimization algorithm has proven effective at finding high scoring grasp/base position combinations and is capable of constructing plans to perform complex pick-and-place tasks.

In future work, we would like to implement our algorithm as a module for manipulation among movable obstacles, as well as applying it to mobile manipulation by humanoid robots. Also, if our optimization algorithm is unable to find a $Grasp$ and positions that are valid for $qO_{start}$ and $qO_{goal}$ it is likely that no such combination exists and that regrasping is necessary to move the object from $qO_{start}$ to $qO_{goal}$. In that case, the optimization results are still useful because we will have grasps and base positions that are valid for both $qO_{start}$ and $qO_{goal}$ in the populations (but no grasp valid for both). The remaining task is to find an intermediate regrasp that links two grasps in the grasp population preferred by different base positions.

## VII. ACKNOWLEDGMENTS

We gratefully acknowledge the students and faculty involved with the ARMAR humanoid platform at the University of Karlsruhe for the use of the kitchen environment model. We would also like to thank Rosen Diankov for his assistance with the simulator and other valuable discussions.

## REFERENCES

[1] LaValle, S., and Kuffner, J., "Randomized Kinodynamic Planning," *Int. Journal of Robotics Research*, Vol. 20, 2001.

[2] Pollard, N., "Closure and Quality Equivalence for Efficient Synthesis of Grasps from Examples," *International Journal of Robotics Research* 23(6), pages 595 - 614, June 2004.

[3] Zhu, X., Ding, H., and Li, H. "A quantitative measure for multi-fingered grasps," *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2001.

[4] Li, Z., and Sastry, S., "Task oriented optimal grasping by multifingered robot hands," *IEEE Trans. on Robotics and Automation*, 4(1):32-44, 1988.

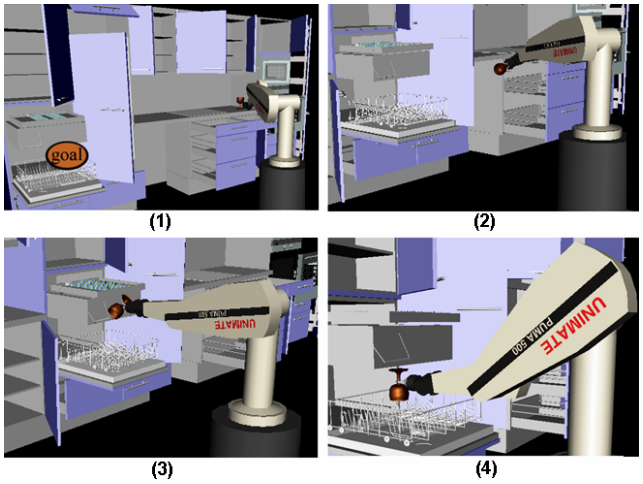


Fig. 5. Snapshots from the execution of a trajectory for the wineglass problem.

[5] Min Zhao, Ansari, N., and Hou, E.S.H., "Mobile Manipulator Path Planning By A Genetic algorithm," *Proceedings of the 1992 lEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 681 - 688, July 1992.

[6] Chen, M., and Zalzala, A., "A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration," *Journal of Robotic Systems*, Vol. 14, No. Y, pages 529 - 544, 1997.

[7] Vannoy, J., and Xiao, J., "Real-time planning of mobile manipulation in dynamic environments of unknown changes," *Proceedings of RSS 2006 Workshop: Manipulation for Human Environments*, August 2006.

[8] Mitchell, M., "An Introduction to Genetic Algorithms," Cambridge, MA: MIT Press, 1996.

[9] Mason, M.T., "Mechanics of Robotic Manipulation," Cambridge, MA: MIT Press, August 2001.

[10] Lozano-Perez, T., Jones, J., Mazer, E., O'Donnell, P., Grimson, E., Tournassoud, P., and Lanusse, A., "HANDY: A Robot System that Recognizes, Plans, and Manipulates," *Proceedings of the 1987 IEEE lnternational Conference on Robotics and Automation*, March 1987.

[11] Koga, Y., Kondo, K., Kuffner, J., and Latombe, J-C., "Planning Motions with Intentions," *Proceedings of SIGGRAPH 94* pp. 395-408, 1994.

[12] Alami, R., Laumond, J.P., and Simeon, T., "Two manipulation planning algorithms," *Workshop on Algorithmic Foundations of Robotics (WAFR94)*, 1995.

[13] Simeon, T., Cortes, J., Sahbani, A., and Laumond, J.P., "A Manipulation Planner for Pick and Place Operations under Continuous Grasps and Placements," *Proceedings of the 2002 IEEE lnternational Conference on Robotics and Automation*, May 2002.

[14] Cambon, S., Gravot, F., and Alami, R., "A robot task planner that merges symbolic and geometric reasonning." ECAI, 2004.

[15] Stilman, M., Schamburek, J., Kuffner, J., and Asfour, T., "Manipulation Planning Among Movable Obstacles" *Proceedings of the IEEE International Conference on Robotics and Automation*, April, 2007.

[16] Nagatani, K., Yuta, S., "Designing strategy and implementation of mobile manipulator control system for opening doors," *IEEE International Conference on Robotics and Automation*, April, 1996.

[17] Yamamoto, Y., Yun, X., "Coordinating locomotion and manipulation of a mobile manipulator," *IEEE Conference on Decision and Control*, 1992.

[18] Seraji, H., "A Unified Approach to Motion Control of Mobile Manipulators," *The International Journal of Robotics Research*, Vol. 17, No. 2, pages 107 - 118, 1998.

[19] Petrovskaya, A., and Ng, A., "Probabilistic Mobile Manipulation in Dynamic Environments, with Application to Opening Doors," *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[20] Hsu, D., Latcombe, J.C., and Sorkin, S., "Placing a robot manipulator amid obstacles for optimized execution" *IEEE International Symposium on Assembly and Task Planning*, 1999.

[21] Brock, O., Khatib, O., and Viji, S., "Task-Consistent Obstacle Avoidance and Motion Behavior for Mobile Manipulation," *IEEE International Conference on Robotics and Automation*, Washington, 2002.