Transition-based RRT for Path Planning in Continuous Cost Spaces

Léonard Jaillet, Juan Cortés and **Thierry Siméon**

LAAS-CNRS, Univerité de Toulouse

7 Av. du Colonel Roche, F-31077 Toulouse, France

Abstract

This paper presents a new method called Transition-based RRT (T-RRT) for path planning problems in continuous cost spaces. It combines the exploration strength of the RRT algorithm that rapidly grow random trees toward unexplored regions of the space, with the efficiency of stochastic optimization methods that use transition tests to accept or to reject a new potential state. This planner also relies on the notion of minimal work path that gives a quantitative way to compare path costs. The method also integrates self tuning of a parameter controlling its exploratory behavior. It yields to solution paths that efficiently follow low cost valleys and the saddle points of the cost space. Simulation results show that the method can be applied to a large set of applications including terrain costmap motions or planning low cost motions for free flying or articulated robots.

INTRODUCTION

Sampling-based path planning techniques (Choset et al. 2005; LaValle 2006) have been successfully applied to a large range of problems such as robotics, manufacturing, graphics animation or computational biology. These techniques handle complex problems in high dimensional spaces but usually operate in a binary representation of the search space made up of free regions constrained by obstacles. Due to their probabilistic nature, solution paths are generally low quality and a postprocessing phase is commonly used to improve them relatively to specific criteria, usually the length or the distance to the obstacles. The problem of computing good quality paths with respect to more general cost functions arises in many applications but remains however an open issue.

On the other side, techniques specially developed in the context of field robotics (Iagnemma and Dubowsky 2004) have focused on path planning methods on costmaps, where the assigned costs depend on different parameters (eg. robot stability, visibility, probability of presence of an obstacle). Contrarily to sampling based techniques, these methods are however limited to low dimensional problems as they are based on graph search algorithms (eg. A^* (Nilsson 1980) and its dynamic extension D^* (Stentz 1994))
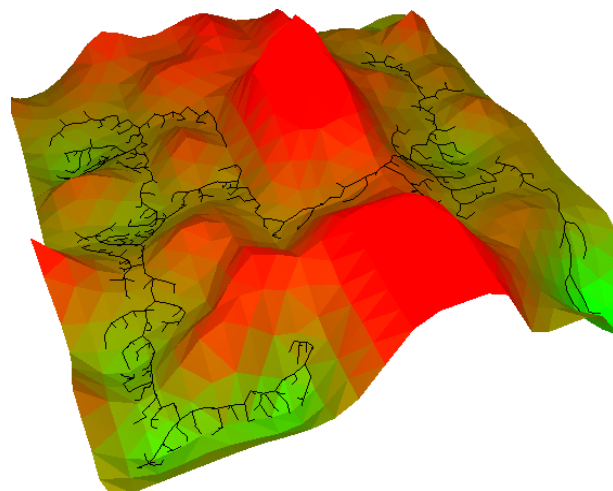

Figure 1: Transition-based RRT on a 2D costmap (the elevation corresponds to the costs). The exploration favors the expansion in valleys and saddle points connecting low cost regions.

inside a grid discretization of the space. Few works (Ettlin and Bleuler 2006b; 2006a; Urmson and Simmons 2003; Ferguson and Stentz 2006) have recently tried to bridge the gap between sampling based planning and costmap planning. However these methods using variants of the RRT diffusion technique (Kuffner and LaValle 2000), focus generally on 2D problems with cost landscapes of relatively low complexity.

This paper presents a new technique[1] called Transition-based RRT (T-RRT) for handling more general problems involving continuous cost spaces. This method extends cost planners to higher dimensional problems. More generally it is applicable to path planning for free-flying or articulated robots in cluttered 3D workspaces, in problems also requiring to integrate some additional cost criterion during the search process in order to compute low cost solution paths. The proposed planner relies on the notion of minimal work

[1]This work has been submitted for presentation at the IEEE/RSJ 2008 Int. Conf. on Intelligent Robots and Systems.

path that gives a quantitative way to compare path costs. The algorithm takes advantage of the exploration strength of RRT-like methods that steers the search toward yet unexplored regions of the space, combined with the efficiency of stochastic optimization methods (eg. Monte Carlo optimization, simulated annealing) that use transition tests to accept or to reject a new potential state. This makes the planner follow efficiently valleys and saddle points of the cost space in order to compute low cost solution paths (Figure 1). Finally, the method also integrates self tuning of a parameter controlling its exploratory behavior.

Related Work

This section briefly surveys the most directly related work to our approach. Only few papers deal with sampling-based planning and cost spaces. In (Ettlin and Bleuler 2006b), an adaptation of the RRT-Connect planner is used to find low cost paths in rough terrain applications. The principle is to keep new configurations only if their cost (called *obstacleness*) is under a given threshold. This threshold is initialized to a low value, and then iteratively increased during the search process. One limitation of this technique is due to the nondecreasing behavior of this threshold, which limits the efficiency of low cost search to the vicinity of the root nodes. To overcome this issue, the extension proposed in (Ettlin and Bleuler 2006a), is to grow multiple RRTs rooted at randomly sampled configurations. However, this solution still requires to tune correctly the number of initial samples in order to get sufficient number of low cost seeds among the space. Moreover, it requires a manual tuning of the parameter controlling the cost growth rate that clearly impacts on the algorithm efficiency.

In (Urmson and Simmons 2003), the heuristically-guided RRT (hRRT) bias the search toward low cost paths using a quality measure based on the cost of the path from the root node and an estimation of the optimal cost to the goal. This principle can be similarly found in (Ferguson and Stentz 2006; Diankov and Kuffner 2007). However such integration of costs along the paths tends to strongly bias the search toward the goal at the expense of less globally optimal solution paths. Moreover, the above methods have been only demonstrated on simple examples with discrete cost states (invalid, low cost, high cost). Their scalability and performance for problems involving complex cost spaces in higher dimensions remains to be established.

The T-RRT technique introduced below is inspired by Monte Carlo optimization techniques. Developed in order to find global optima in very complex function spaces (Spall 2003), they introduce randomness as a mean to avoid local minima traps. Many variants have been developed (eg. random walk, simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983)). The basic exploration process relies commonly on successive transition tests, performed using the Metropolis criterion. Also note that the Probabilistic Conformational Roadmap (Apaydin et al. 2001) developed for exploring molecular energy landscapes in computational biology applications integrates a similar transition test in the PRM framework (Kavraki et al. 1996).

Minimal Work Paths

In this section we propose a new criterion to better evaluate path costs. This criterion, based on the notion of mechanical work, gives us a quantitative way to compare path qualities.

Let us consider a robot evolving in a configuration space CS that may contain "binary" obstacles, and a continuous cost function $c : CS \rightarrow \mathbb{R}^+$ mapping this space, ie. a cost $c(q) = c_q$ can be computed for each $q \in CS$. Consider also a path $\mathcal{P}$ with a given cost profile corresponding to the various configurations costs along $\mathcal{P}$. A positive variation of cost can be seen as a force acting against the motion that produces a mechanical work with the displacement of the robot along the path. The resulting loss of "energy" due to mechanical work is actually the criterion that we propose to minimize. For negative variation of costs, the robot does not loose any energy. However, a small cost proportional to the distance is always added in order to privilege shortest paths of equal mechanical energy. Based on this principle, the cost of a path called $W(\mathcal{P})$ (in analogy with the mechanical work) can be mathematically expressed as follows:

$$W(\mathcal{P}) = \int_{s_+} \frac{\partial c_+}{\partial s} ds + \epsilon \int_s ds$$

Here, s^+ represents the path portions with positive slopes, ie. where the cost variations are strictly positive. ϵ is taken very low compared to the cost values, in order to primarily favor paths with a mechanical work as low as possible and then take among them the paths that are as short as possible. For a path representation discretized in $n + 1$ configurations linked by n edges, we have:

$$W(\mathcal{P}) = \sum_{i \in i_+} \left(c(q_i) - c(q_{i-1}) \right) d_i + \epsilon \sum_{i \in i} d_i$$

where $i \in i_+$ when $c(q_i)$-$c(q_{i-1}) > 0$, and d_i is the distance between q_{i-1} and q_i.

Figure 2 shows minimal work paths obtained for several queries on a 2D costmap example. For such low dimensional problems, these paths were simply computed using a standard A^* search performed on a grid discretizing the 2 dimensional landscape. The shapes of the optimal paths are "visually" good, in the sense that they follow as much as possible the low cost regions of the space.

However such grid based planner remains limited to low dimensional problems. The next section introduces the T-RRT algorithm for exploring possibly high dimensional free-space while biasing the search toward low cost regions such that solution paths remain close to minimal work paths. More generally, the method can also be applied to problems with "binary" obstacles for computing low cost paths according to a given cost function defined over the configuration space (see results provided below).
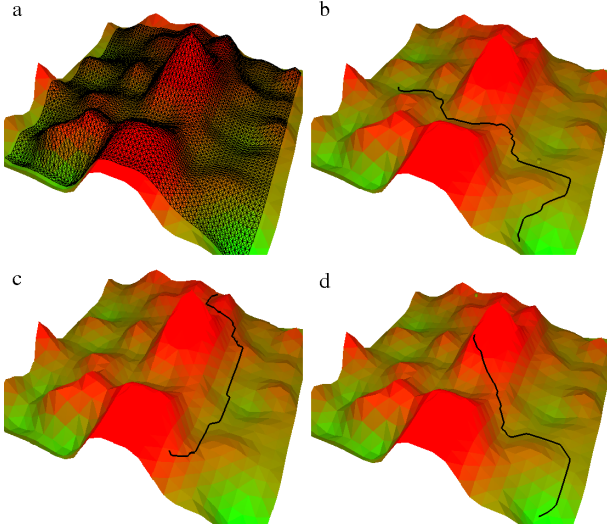
Figure 2: Optimal paths based on the minimal work criterion. The paths are computed using the A^* algorithm within a 2-dimensional grid (a) discretizing the space. The examples illustrate respectively down-to-down (b), top-to-top (c), and top-to-down (d) queries.

Transition-based RRT

General Approach

The main idea of the Transition-based RRT algorithm (T-RRT) is to take advantage of the performance of two methods. First, the exploration strength of RRT-like algorithms resulting from their exploration bias toward large Voronoi regions of the space. Second, the efficiency of stochastic optimization methods developed for computing global minima in complex spaces, and that use transition tests to accept or reject potential states.

Algorithm 1 shows the pseudo-code of the T-RRT planner. Similarly to the *Extend* version of the classical RRT algorithm (LaValle 1998), a configuration q_{rand} is randomly sampled yielding both to the nearest tree node q_{near} to be extended and to the expansion direction performed from q_{near} toward q_{rand}, with a fixed increment step δ.

The increment δ is chosen small enough to approximate well the cost variation between q_{near} and q_{new}, meaning that linear interpolation does not miss cost picks between the two configurations. Note that in the presence of "binary" obstacles, this stage also integrates some collision detection that rejects the colliding expansion nodes.

While this first stage ensures the bias toward unexplored free regions of the space, the goal of the second stage is to filter irrelevant configurations regarding the search of low cost paths before inserting q_{new} in the tree. As explained above, such filtering is performed by the `TransitionTest` function based on the Metropolis criterion commonly used in stochastic optimization methods.

Additionally, we propose an original way to automatically control the filtering power strength of this test in order to ensure the growth of the tree. Finally,

---

Algorithm 1: Transition-based RRT

input : the configuration space CS;
the cost function $c : CS \rightarrow \mathbb{R}_*^+$;
the root q_{init} and the goal q_{goal};

output : the tree $\mathcal{T}$;

begin

$\quad \mathcal{T} \leftarrow$ InitTree(q_{init});
$\quad$ **while not** StopCondition($\mathcal{T}$, q_{goal}) **do**
$\quad\quad q_{rand} \leftarrow$ SampleConf(CS);
$\quad\quad q_{near} \leftarrow$ BestNeighbor(q_{rand}, $\mathcal{T}$);
$\quad\quad$ **if not** Extend($\mathcal{T}, q_{rand}, q_{near}, q_{new}$) **Continue**;
$\quad\quad$ **if** TransitionTest($c(q_{near}), c(q_{new}), d_{near-new}$)
$\quad\quad$ **and** MinExpandControl($\mathcal{T}, q_{near}, q_{rand}$) **then**
$\quad\quad\quad$ AddNewNode($\mathcal{T}, q_{new}$);
$\quad\quad\quad$ AddNewEdge($\mathcal{T}, q_{near}, q_{new}$);

end

---

the `MinExpandControl` function forces the planner to maintain a minimal exploration rate and thus avoid possible blocking situations during the search. The following subsections details the `TransitionTest` and `MinExpandControl` functions.

Transition Test

The `TransitionTest` function is presented on Algorithm 2. First, configurations whose cost is higher than the maximal threshold cost c_{max} are filtered. We then define a probability of acceptance of a new configuration by comparing its cost c_j relatively to the cost c_i of its parent configuration. This transition test is based on the Metropolis criterion. The transition probability p_{ij} (referred as the Bolzmann probability when applied to statistical physics or molecular modeling) is defined as follows:

$$p_{ij} = \begin{cases} exp(-\frac{\Delta c_{ij}^*}{K*T}) & \text{if } \Delta c_{ij}^* > 0 \\ 1 & \text{otherwise.} \end{cases}$$

with:

- $\Delta c_{ij}^* = \frac{c_j - c_i}{d_{ij}}$, the slope of the cost, ie. the cost variation divided by the distances between the configurations. Thus, downhill transitions are automatically accepted whereas for uphill transitions, those with the steepest slopes have the lowest chances to be accepted.

- K is a constant value used to normalize the cost variations, simply determined by the average cost of query configurations $K = \frac{c_{init} + c_{goal}}{2}$.

- T is a "temperature" parameter that controls the difficulty level of transition tests.

Temperature Parameter T is an important parameter of the algorithm since low temperatures limit the expansion to low positive slopes. On the contrary, higher temperatures permit expansion on higher slopes. Among the various applications of the Metropolis criterion, the temperature can be kept constant (eg. Monte Carlo search) or decreased gradually as the search progresses (eg. simulated annealing).

Algorithm 2: TransitionTest(c_i, c_j,d_{ij})

---

begin
    if $c_j > c_{max}$ **then** return False;
    if $c_j < c_i$ **then** return True;
    $p = exp(\frac{-(c_j - c_i)/d_{ij}}{K * T})$;
    if $Rand(0, 1) < p$ **then**
        $T = T/\alpha$;
        $nFail = 0$;
        return True;
    else
        if $nFail > nFail_{max}$ **then**
            $T = T * \alpha$;
            $nFail = 0$;
        else
            $nFail = nFail + 1$;
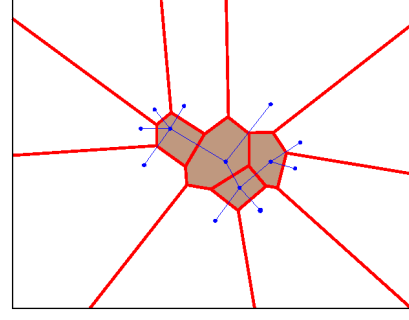   return False;
end

Figure 3: *Frontier* nodes (white regions) have a Voronoi region bounded by the space limits contrarily to *non frontier* nodes with a Voronoi region bounded by the Voronoi region of other nodes (brown/gray regions) . The distinction of these 2 types of nodes is specially useful in the case of cost spaces.
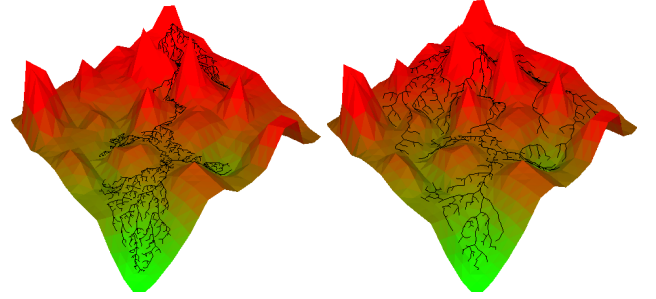

Figure 4: Impact of the minimal expansion control on the T-RRT algorithm. Without control (left), the insertion of refinement nodes tends to slow down the exploration by decreasing the temperature. By performing this control, the planner is forced to keep exploring new regions of the space.

Adaptive Tuning The `TransitionTest` function performs an adaptive tuning of the temperature during the search process (second stage of Algorithm 2). At the initialization of the algorithm, T is first set to a very low value (eg. 10^{-6}) in order to only authorize very low positive slopes (and negative ones). Then, during the exploration, the number $nFail$ of consecutive times the Metropolis criterion discards a configuration is memorized and used for temperature tuning. When the T-RRT search reaches a maximal number of rejections $nFail_{max}$, then the temperature is multiplied by a given factor α. On the contrary, each time the tree successes to climb a slope, the temperature is divided by the same factor α. Proceeding in this way, the temperature automatically adapts itself such that in average a new node climbing a slope is inserted every $nFail_{max}$ times. For all our experiments, we have taken $\alpha = 2$ and $nFail_{max} = 100$.

Minimal Expansion Control

The adaptive temperature tuning introduced above for the transition tests ensures a given rate of slope climbing successes. However in some cases, a possible side effect can be a too slow expansion of the tree versus an excessive refinement of explored regions. We further discuss below this issue and explain how the `MinExpandControl` function overcomes this problem.

Exploration versus Refinement The behavior of the RRT expansion can be explained by distinguishing two types of nodes (Yershova et al. 2005): *frontier* nodes (with a Voronoi region bounded by the space limits) and *non frontier* nodes (with Voronoi region entirely bounded by the Voronoi region of the other nodes). Whereas the extension of a *frontier* node tends to explore new regions of the space, the extension of *non frontier* nodes only leads to the refinement of the existing tree (see Figure 3). In the case of the T-RRT method, it happens that whereas the temperature needs to grow in order to develop *frontier* nodes and make the tree

explore new regions, it is stabilized by the insertion of *non frontier* nodes refining the tree in easier regions of the space. The problem of unbalanced refinements and explorations modes was already addressed in (Yershova et al. 2005; Jaillet et al. 2005) for standard RRTs. However the problem with T-RRT is different since the insertion of refinement nodes directly impacts on the temperature and thus on the probability to create new exploration nodes. Figure 4 illustrates this issue with an example of tree (left) whose expansion has been stopped by the too frequent insertion of *non frontier* nodes. It also shows the solution to the problem obtained using the minimal expansion control that is detailed below (right).

Minimal Exploration Rate The proposed solution is to force the planner to keep exploring new regions of the space by controlling the ratio between exploration and refinement steps. It is important to note that *non frontier* nodes have a Voronoi region usually much smaller than the one of *frontier* nodes (bounded by the Voronoi regions of other nodes). Based on this property, extension steps can be estimated as refinements or expansions in function of the distance be-

tween q_{near} and q_{rand} (the rationale is that a small distance corresponds to configurations having a higher probability to be *non frontier* nodes). In practice, q_{new} is considered as participating to the tree expansion if this distance is higher than the expansion step δ. On the contrary, q_{new} is supposed to participate to the tree refinement when the distance is lower than δ. The control of minimal exploration rate is performed by the `MinExpandControl` function that rejects configurations accepted by the transition-based test if they make the ratio between exploration and refinement mode lower than a given minimal value (set to 1 in our implementation).

Experimental Results

The basic Extend-RRT algorithm and its T-RRT variant have been implemented into the path planning software *Move3D* (Siméon, Laumond, and Lamiraux 2001). The experiments reported below were performed on a 1GHz AMD Opteron 148 processor. All performance results summarized in the tables correspond to average values computed over 10 runs.

The first set of experiments evaluates T-RRT in the case of a 2-dimensional cost space (Figure 5). In this example, the solution paths have to get through a saddle point to link the query configurations located at two opposite corners of the environment. The figure shows snapshots of the exploration tree during the search process and the solution path found (bottom-left) which is close to the optimal one (bottom-right). Path qualities using T-RRT are compared in Table 1 to those using a basic RRT algorithm and to the optimal path P^* computed with an A^* search within a 128×128 grid discretizing the space.
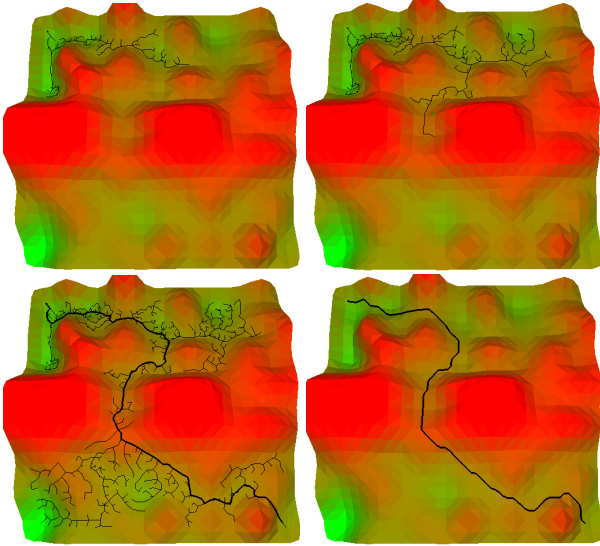

Figure 5: Expansion process of the Transition-based RRT finding a low cost solution path (bottom-left), close to the optimal one (bottom-right) computed from a space discretization.

Table 1: 2D cost space, comparative results for RRT, T-RRT ($nFail_{max}$ = 100, 10) and the optimal path $\mathcal{P}^*$.

	$length$	c_{min}	c_{max}	c_{ave}	W
RRT	148	8	36	21	**32.7**
T-RRT	214	8	23	17	**19.5**
T-RRT$_{10}$	182	8	25	18	**21.9**
$\mathcal{P}^*$	178	10	23	17	**13.3**

Solution paths for the classic RRT algorithm (almost straight line paths) are 2.5 times more costly than the optimal solution path (32.7 vs. 13.3). In comparison T-RRT path costs are only 45% higher than the optimal one. Most importantly, the overall shape of solution paths is very close to the optimal one and follows the same low cost regions. Therefore, a simple post processing (local optimization) would certainly be sufficient to tend toward the optimal solution path. Concerning the influence of the $nFail_{max}$ parameter, note that a smaller value ($nFail_{max}$ = 10 instead of 100) yields to very similar solution paths with slightly higher costs, but balanced by a large improvement of computing time performance (1.4s vs. 28.6s).

Next experiments illustrate the generality of the algorithm, which is not limited to 2D costmap planning, but also able to integrate cost functions for planning motions of more complex robot systems. First example involves a 6-dof manipulator arm carrying a rod in 3D environment cluttered with obstacles (see Figure 6). The goal of the planner is to extract the rod from a hole, keeping it as far as possible from the obstacles. Thus the cost function considered for this case corresponds to the inverse of the distance of the rod to the obstacles. Results are presented on Table 2. The cost of solution paths found with T-RRT are considerably lower than the ones found with RRT (with respective computing times of $19.2s$ and $3.3s$). This difference of quality is much higher here since there is no limitation on the cost function (contact configurations have an infinite cost) contrarily to the costmap example of Figure 5. The last row of Table 2 illustrates an interesting feature of T-RRT. The standard deviation σ_W computed over the solution path costs shows the regularity of the paths costs obtained with the T-RRT method. Contrarily to the erratic motions of the classical RRT, T-RRT paths tends to keep the rod horizontal during its extraction from the hole in order to stay at a fixed maximal distance from the obstacles.

Last experiment was performed on the molecular model shown Figure 7. The task is to compute the pathway extract-

Table 2: 6-dof manipulator arm with binary obstacles, comparative results for RRT and T-RRT.

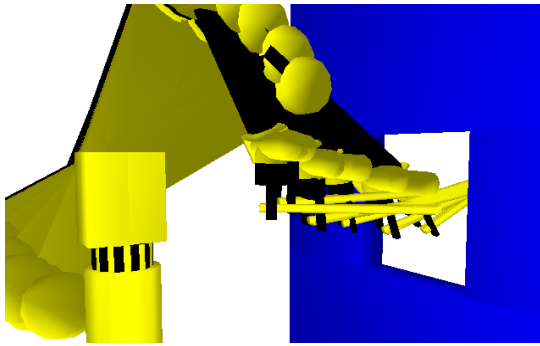|	$length$	c_{min}	c_{max}	c_{ave}	W	σ_W
RRT	45	0.1	17150	88	**3279**	3151
T-RRT	49	0.1	40.4	1	**15.2**	5.22

Figure 6: 6-dof manipulator arm extracting a rod from a hole thanks to the T-RRT planner. The solution path tends to keep the rod horizontal during the motion in order to stay as far as possible from the obstacles.
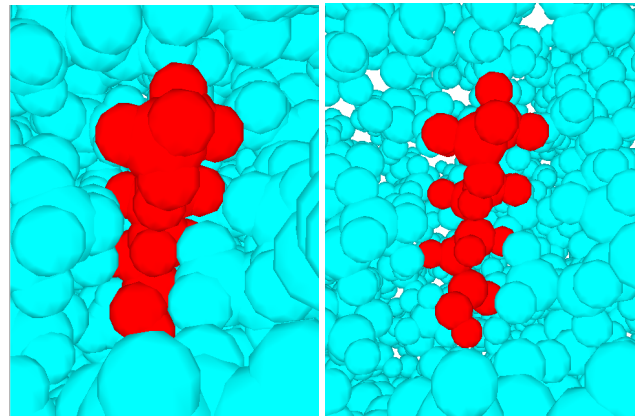

Figure 7: Two representations of the same ligand protein problem, with different van der Waals radii: maximal radius (left) and shrunk radius (right).

ing the ligand (small molecule in red/dark) from the active site located inside a protein. This problem can be seen as a mechanical disassembly path planning problem for the free-flying ligand (Cortés, Jaillet, and Siméon 2008). Energetic constraints are translated into geometric ones by considering a steric model of the molecules and applying a collision detection (de Angulo, Cortés, and Siméon 2005) as a geometric filter that rejects conformations with prohibitively high van der Waals (VdW) energy. Like for the manipulator example, the cost function considered in this case is defined as the inverse of distance between the ligand and the protein. The interest of this molecular model is to provide a simple way to quantify the quality of the solution path. Hence, a dilatation of the ligand free space can be simply obtained by shrinking the atoms radii. The results reported in Table 3 were obtained by applying both RRT and T-RRT algorithms on the shrunk model shown in the right image in Figure 7 (25% of VdW radii). T-RRT solution path has a much lower cost compared to the one computed by RRT. The higher clearance of the T-RRT solution is also quantified by the maximal VdW ratio indicated in the last row of the table. This maximal ratio was simply obtained by testing solution paths with VdW radii increasingly grown until a collision was detected between the ligand and the protein. While no growing was possible for the RRT solution, the T-RRT path (computed with a 25% ratio) remains valid up to 66%, which is relatively close to the maximal value allowing the ligand to exit (80%). This high clearance of the T-RRT computed path shows its good quality with respect to the distance-based cost considered for this example.

Table 3: Ligand-Protein example with an initial VdW ratio of 25%, comparative results between RRT and T-RRT.

	$length$	c_{min}	c_{max}	c_{ave}	W	VdW
RRT	59	0.1	2236	14	**471**	0.25
T-RRT	70	0.1	1.0	0.3	**0.3**	0.66

CONCLUSIONS AND FUTURE WORKS

We have presented the Transition-based RRT algorithm (T-RRT) that takes advantage of both the exploration strength of the RRT and the efficiency of stochastic optimization methods. It integrates a self tuning of the temperature parameter to adapt the difficulty of the transition tests in function of failures and successes during the search process. An explicit definition of the optimality criterion allows us to compare solution path with the optimal ones and helps us to quantify the experimental results. The algorithm is general enough to be applied to high-dimensional spaces also constrained by obstacles or to classical costmaps in rough terrain applications.

The first simulation results reported in the paper show the effectiveness of the method for solving both classes of problems. However there remain several possible improvements, in particular to consider higher dimensional problems or to increase the optimality with some postprocessing of the T-RRT computed path. We also plan to investigate possible applications of the technique: the integration of human-robot interaction constraints into computed motions and also the exploration of energy landscapes in computational biology problems.

ACKNOWLEDGMENTS

References

Apaydin, M.; Singh, A.; Brutlag, D.; and Latombe, J.-C. 2001. Capturing molecular energy landscapes with probabilistic conformational roadmaps. *Proc. IEEE Int. Conf. on Robotics and Automation* 932–939.

Choset, H.; Lynch, K.; Hutchinson, S.; Kantor, G.; Burgard, W.; Kavraki, L.; and Thrun, S. 2005. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge: MIT Press.

Cortés, J.; Jaillet, L.; and Siméon, T. 2008. Disassembly path planning for complex articulated objects. *IEEE Transactions on Robotics and Automation*. In press.

de Angulo, V. R.; Cortés, J.; and Siméon, T. 2005. BioCD: An efficient algorithm for self-collision and distance computation between highly articulated molecular models. In snd G. Sukhatme, S. T.; Schaal, S.; and Brock, O., eds., *Robotics: Science and Systems*. Cambridge: MIT Press. 6–11.

Diankov, R., and Kuffner, J. 2007. Randomized statistical path planning. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*.

Ettlin, A., and Bleuler, H. 2006a. Randomised rough-terrain robot motion planning. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* 5798–5803.

Ettlin, A., and Bleuler, H. 2006b. Rough-terrain robot motion planning based on obstacleness. *Proc. Int. Conf. on Control, Automation, Robotics and Vision* 1–6.

Ferguson, D., and Stentz, A. 2006. Anytime rrts. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* 5369 – 5375.

Iagnemma, K., and Dubowsky, S. 2004. *Mobile Robots in Rough Terrain: Estimation, Motion Planning, and Control with application to Planetary Rovers*. Springer Tracts in Advanced Robotics (STAR) Series, Springer.

Jaillet, L.; Yershova, A.; LaValle, S.; and Siméon, T. 2005. Adaptive tuning of the sampling domain for dynamic-domain rrts. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* 4086–4091.

Kavraki, L.; Svestka, P.; Latombe, J.-C.; and Overmars, M. 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* 12(4):566–580.

Kirkpatrick, S.; Gelatt, C.; and Vecchi, M. 1983. Optimization by simulated annealing. *Science* 220:671–680.

Kuffner, J., and LaValle, S. 2000. Rrt-connect: An efficient approach to single-query path planning. *Proc. IEEE Int. Conf. on Robotics and Automation* 995–1001.

LaValle, S. 1998. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*. TR 98-11, Computer Science Dept., Iowa State University.

LaValle, S. 2006. *Planning Algorithms*. New York: Cambridge University Press.

Nilsson, N. 1980. *Principles of Artificial Intelligence*. Palo Alto, CA: Tioga Publishing Company.

Siméon, T.; Laumond, J.-P.; and Lamiraux, F. 2001. Move3D: A generic platform for path planning. *Proc. IEEE Int. Symp. on Assembly & Task Planning* 25–30.

Spall, J. 2003. *Introduction to Stochastic Search and Optimization*. New York: Wiley.

Stentz, A. 1994. Optimal and efficient path planning for partially-known environments. *Proc. IEEE Int. Conf. on Robotics and Automation* 3310–3317.

Urmson, C., and Simmons, R. 2003. Approaches for heuristically biasing RRT growth. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* 1178–1183.

Yershova, A.; Jaillet, L.; Siméon, T.; and LaValle, S. 2005. Dynamic-domain RRTs: Efficient exploration by controlling the sampling domain. *Proc. IEEE Int. Conf. on Robotics and Automation* 3867–3872.