On Addressing the Run–Cost Variance in Randomized Motion Planners

Pekka Isto and Martti Mäntylä

September 30, 2002

ON ADDRESSING THE RUN-COST VARIANCE IN RANDOMIZED MOTION PLANNERS

Pekka Isto* and Martti Mäntylä**

*Helsinki University of Technology, Espoo, Finland, evp@cs.hut.fi

**Helsinki Institute for Information Technology, Helsinki, Finland, martti.mantyla@hiit.fi

ON ADDRESSING THE RUN-COST VARIANCE IN RANDOMIZED MOTION PLANNERS

Pekka Isto

Helsinki University of Technology, Espoo, Finland, evp@cs.hut.fi

Martti Mäntylä

Helsinki Institute of Information Technology, Helsinki, Finland, Martti.Mantyla@hiit.fi

Abstract

The decades of research in motion planning have resulted in numerous algorithms. Many of the most successful algorithms are randomized and can have widely differing run-times for the same problem instance from run to run. While this property is known to be undesirable from user's point of view, it has been largely ignored in past research. This paper introduces the large run-cost variance of randomized motion planners as a distinct issue to be addressed in future research. Run-cost variance is an important performance characteristic of an algorithm that should be studied together with the mean run-cost. As a positive example of possibilities for reducing the run-cost variance of a randomized motion planner, simple heuristic techniques are introduced and investigated empirically.

1. Introduction

Motion planning capability is an essential property of any autonomous robotic system [1]. When formulated as the classical "mover's" problem, it can be shown that the problem is PSPACE-hard [2]. The most basic variation of the problem assumes that all obstacles are known and static during the planning and execution of the motion for the movable device. A large number of motion planning algorithms have been presented for this and other variations of the problem [1][2][3]. Due to the complexity of the problem, practical planners must be based on heuristic and incomplete algorithms.

The complexity of the problem guarantees that for any algorithm there exists a problem instance that exhibits the intractable worst-case behavior. Heuristics can take advantage of assumed regularities in the problem, but when the heuristics is known, the construction of a "deceptive" problem instance exhibiting worst-case behavior is usually quite straightforward. Relaxing the completeness requirement has allowed the development of randomized motion planners that are complete only in probabilistic sense. Randomized planners use either some randomized search procedure, e.g. [16][22], or random sampling of the configuration space (*cspace*) for subgoals in combination of local search between the subgoals, e.g. [14][18][28][19].

The current motion planners that rely on configuration space samples for subgoals are very sensitive to the selection of the particular sampling sequence, be it a particular segment of a pseudo-random number sequence or a particular type of a quasi-random number sequence. Structured sampling techniques, e.g. [6][8][9], usually rely on assumptions on the task structure that are easily violated [20]. Since randomized algorithms tend to break the task structure, they are usually robust against "deceptive" inputs. However, this robustness comes at the expense of being able to reliably estimate the length of a particular run of the algorithm due to run-cost variance.

This paper introduces the large run-cost variance of randomized motion planners as a distinct issue to be addressed by the research community in the future work. Variance should be accounted for in the research methodology, especially if it is empirical [30]. If a new or variant randomized motion planner is presented, appropriate descriptive statistics should be presented to describe the run-cost variance of the proposed motion planner. Users are known to find run-cost variance distressing [25], therefore motion planners with low variance are preferred. The size of variance is also directly involved in comparisons between various planners or components of planners, since large variance can invalidate the statistical significance of an observed difference in average performance.

Furthermore, in order to demonstrate prospects for addressing the variance problem, simple techniques with variance reduction effect are presented. The techniques are heuristic in nature and their effects are verified empirically. The techniques are applicable to motion planners that use samples in the configuration space as subgoals to be connected with a local search procedure. In essence, they make the local search procedure more powerful when it is observed that a large number of samples or local planner calls are needed for the solution and thus the particular sampling sequence cannot be relied to be successful for the task at hand. Since the search procedure used here is deterministic, this can be seen as shifting the balance of computation away

from the randomized sampling procedure to the deterministic search procedure.

The next section presents a brief overview of randomized motion planners. Section 3 introduces PRM motion planner variants with heuristics that improve performance both in the expected run-cost and the run-cost variance. Section 4 describes the empirical methodology used in this paper and section 5 presents the results. Finally, section 6 presents the conclusions and issues for future work.

2. Previous Work

Randomization techniques have been introduced into motion planning algorithms in order to avoid expending large amount of computation when a deterministic planner exhibits the worst-case behavior. Classical cell decomposition methods construct a *cspace* representation for the entire *cspace* although only a part may be necessary for the construction of the solution. Heuristic strategies can be used to restrict the construction of the *cspace* representation to promising areas, e.g. [10][11]. However, when the heuristics fails and guides the construction to some unimportant area, the complexity of the problem may again reveal itself with large costs in terms of time and space requirements. This renders planners utilizing straightforward best-first search strategies useless for practical problems, since the planners tend to get trapped examining dead-end regions of the *cspace*. This has been long known as the problem of local minima [1].

The same problem is also present when the guiding heuristics takes the form of a potential field over the workspace or the *cspace*. A theoretical result establishes that all global navigation functions have saddle points [12], and therefore, simple gradient procedures can terminate at some configuration other than the intended goal configuration of the problem. Again, the question becomes what to do once the (deterministic) primary technique ceases to make progress.

Randomization was introduced in order to solve the local minima problem. A randomized search procedure can be obtained by combining a heuristic search strategy with randomly generated subgoal configurations. The heuristic strategy is first attempted between the original start and the original goal configuration. If the heuristics guides the search to a local minimum, a random subgoal is generated and the heuristic strategy is attempted via the subgoal configuration. As more subgoals are generated and path segments are generated between them with the heuristic strategy, they will form a graph that approximates the connectivity of the *cspace* [14][15]. The heuristic strategy can be as simple as a straight-line interpolation between the subgoals [2]. This general idea has evolved into a randomized version of the classical roadmap approach [5][6][7] and it has been a topic for extensive research under the name of Probabilistic Roadmap Planning (PRM). Other similar motion planners use more complex heuristics for search and subgoal generation, e.g. [19][18].

An early and influential motion planner called Randomized Path Planner (RPP) combined the potential field approach with a random walk procedure to escape from the local minima [16]. Several variations of the planner were introduced, but it had difficulties in solving problems that required long walks against the potential [17]. A special random exploration procedure was later introduced to improve the efficiency of randomized search [22].

Recently, the use of quasi-random sampling has been proposed as an alternative for the more usual pseudo-random sampling [23]. While eliminating the variance caused by different seeds for the random number generator, it remains brittle with respect to small but critical changes in the workspace of the problem. Using quasi-random samples to generate the search space representation for search procedures [24] reintroduces the problem of local minima, which was effectively solved by using the samples as subgoals.

Very few user experiences with applications utilizing motion planning algorithms have been reported in the literature. Experiences with a maintainability study tool based on RPP indicate that users find the variance inherent in randomized algorithms very disturbing [25]. Although the run-cost variance has been known to be a serious deficiency of the sampling based randomized motion planners [13], no methods have been presented to directly address this problem. This observation has motivated the consideration of the amount of variance as a measure of the performance of an algorithm and the investigation in the possibilities of reducing the variance.

3. Heuristic Techniques for Variance Reduction

The general idea of the heuristic strategy presented in this paper is to increase the relative effort spent in deterministic local search procedure when it is observed that the randomized sampling procedure does not succeed rapidly in producing subgoals that cover the *cspace* adequately. Two variations of this general idea are derived with a simple PRM planner using bidirectional A^* search [26] based local planner and parameterized formulas for increasing the competence of the local planner.

Although A^* search algorithm makes a (resolution) complete motion planner when combined with one of the cell decomposition methods, it has an exponential memory consumption. The memory consumption makes it practically impossible to escape deep local minima, since the algorithm can exhaust any available central memory. Therefore, it alone is not suitable for motion planning, but must be combined with other methods, here with pseudo-random sample subgoals. The approach is to use A^* search local planner on a grid approximation of the *cspace* from one sample to the other, but to discontinue A^* search when it appears to be too costly to generate the path segment with A^*. The underlying assumption is that a sequence of subgoals can be found so that A^* search succeeds in generating all the path segments from the start to target configurations within a defined limit on the search efficiency on each segment.

The essential property of a search algorithm based local planner is that it allows simple evaluation and control of the difficulty of the path segments that the local planner is capable of yielding. A measure of search efficiency can be obtained from the ratio of the size of the examined search space to the length of the best available path candidate towards the target sample. Noting that the local planning is performed in discrete rectangular grid representation of the *cspace* lets us realize the above measure with simple counting operations. Let $F(C)$ be the total number of collision-free configurations examined by the local planner until the examination of configuration C. Furthermore, let $g(C)$ be the distance from the start sample to the configuration C currently examined by the local planner. Now, an efficiency measure $O(C)$ can be defined as

$$O(C) = \frac{F(C)}{g(C)}.$$

The competence of the local planner can be controlled by setting an upper limit O_{th} for $O(C)$ and discontinuing the search in the local planner, if the limit is exceeded. Thus, the local planner is only competent to solve problems in which the search algorithm does not at any point violate the given efficiency limit by requiring more than the set upper limit of grid point examinations for each step proceeded away from the start sample along the currently best solution candidate path.

Note that $O(C)$ measure is similar to Nilsson's penetrance [26]. The difference is that penetrance is evaluated once after the (optimal) path is found, but $O(C)$ is evaluated continuously during the search and used as a control during the search.

The efficiency of a heuristic search algorithm such as A^* is highly dependant on the guiding heuristic function. The usual form of the guiding function is $f(C) = A{\times}g(C) +B{\times} h(C)$, where $g(C)$ is as above, $h(C)$ is a heuristic estimate for the cost from the currently examined configuration C to the target sample, and A and B are constants. The guiding function used here is greedy with smaller A=3 and larger B=5. Manhattan distance in the discrete *cspace* is used for both $g(C)$ and $h(C)$. The heuristic estimate $h(C)$ has an additional tie breaking in favor of configurations that repeat a motion of a joint along the solution path candidate.

The sampling strategy used here is simple pseudo-random sampling until a roadmap connecting all the seed configurations of the given test task is obtained. Candidate sample pairs for the local planner are produced by selecting for the newly generated sample up to k=10 closest samples from each connected component of the roadmap at the sample generation time. Euclidean distance is used as the distance metric in the selection.

An interesting problem is the question of how to determine a threshold for $O(C)$. The majority of PRM planners have static local planners. For this study, static local planners with well-defined capability can be obtained by setting a fixed upper limit o for $O(C)$.

Since the competence of the local planner can be set separately for each call, information gained during the roadmap construction can be used to determine a suitable value for $O(C)$ limit with the goal of reducing the overall roadmap construction cost. Two such adaptation strategies are proposed here and studied experimentally in the following sections. The first strategy involves increasing the $O(C)$ limit linearly with the size of the roadmap. The intuition behind this strategy is that more difficult problems require larger roadmap and a more capable local planner to adequately describe the connectivity of the *cspace*. Furthermore, as more samples are added to the roadmap, the failure probability of the local planner decreases [27]. If the failure probability is lower, failures that are more expensive can be tolerated. This strategy has a global character in the sense that it determines a single increasing $O(C)$ limit for all the samples in the roadmap. The following parameterized formula is used to determine the threshold O_{th} for $O(C)$ during roadmap construction at a particular roadmap size of S:

$$O_{th}(s) = \frac{S}{s} \times 32.$$
(1)

A strategy of local character is defined by setting the O_{th} threshold separately for each sample in

the roadmap. The strategy uses a measure of difficultiness of the *cspace* around a particular sample. For each sample *v* the fraction of successful calls of local planner is computed:

$$r_s(v) = \frac{N_s(v)+1}{N(v)+1},$$

where *N(v)* is the total number of local planner calls with the configuration space sample *v* either as start or target and $N_s(v)$ is the number of calls that succeeded in producing a path segment to or from the sample *v*. This measure is very similar to failure ratio [28].

A value of O_{th} is computed for both start and target samples with the parameterized formula:

$$O_{th}(v,n) = 1 + \frac{n}{r_s(v)}, \tag{2}$$

and the maximum is used as the current threshold value. The intuition behind this strategy is to use an estimate of the difficultiness of the *cspace* region to control the capability of the local planner. A small r_s suggests that sample resides in a difficult region of the *cspace* and thus the local planner should be given opportunity to search the region more broadly.

Parameterized heuristics present a problem of selecting the values for the heuristic parameters. If the properties of the expected motion planning problems are known, then the parameters should of course be tuned for those problems using preliminary experiments [29]. When tuning is not possible or desirable, then some on-line procedure can be used to select the value.

In this paper a metaplanner is used to select the values for parameters *o*, *s* and *n*. Perhaps unsurprisingly, the selection is done randomly from a set of reasonable values for each parameter. The metaplanner selects a parameter value uniformly from the set at the start of the execution of the PRM planner. Based on the preliminary experiments, the reasonable ranges of parameter values were determined and the sets defined to be *{2, 4, 8, 16, 32}* for *o*, *{300, 1000, 3000, 9000}* for *s* and *{0.01, 0.03, 0.1, 0.3}* for *n*. Thus, the planners compared in the experimental section of this paper are PRM variants with bidirectional A^* search between the samples. Planner PRM-S uses a static local planner at a competence level *o* selected by the metaplanner at the start of the run. PRM-G uses the global adaptation of the local planner according to the equation (1) with parameter value *s* selected by the metaplanner. Similarly, PRM-L uses the local adaptation strategy defined by the equation (2) and parameter *n*. The research question is whether the planners PRM-C, PRM-G and PRM-L differ in location or spread of their empirical run-cost distributions. The results are taken to indicate possible benefits of increasing the capability of the local A^* search during planning (PRM-G, PRM-L) against keeping it static (PRM-C).

4. On the Empirical Methodology

The run-cost of a Las Vegas -type randomized algorithm varies from run to run according to some distribution D. Usually, deriving the true run-cost distribution of a randomized algorithm is a formidable task and no such results exists for any of the well-known motion planning algorithms. Sometimes the true distribution is sampled empirically and the results are presented in the form of

a histogram [15]. More often the run-cost distribution is described by reporting estimates for the location and spread of the true distribution [19][21]. This implies a parametric model `D(l,s)` for the run-time distribution, where `l` describes the location of the distribution and `s` the spread of the distribution. Sample average and median are typically used for estimating the location and sample variance, standard deviation or range for estimating the spread.

Most often, however, only an estimate of the location parameter is presented by reporting the sample average of the run-cost over some number of runs. This case is unsatisfactory since it hides one of the most problematic characters of the randomized motion planning algorithms. Furthermore, it makes it impossible to evaluate the statistical significance of differences between algorithms or variations, e.g. when comparing the performance of one sampling strategy with another. An estimate of the standard deviation is needed for the most elementary test of statistical significance, namely the t-test. Of course, the outcome of the test or analysis of variance should be given as an indication of the statistical significance.

The following section presents descriptive statistics on the run-cost of the proposed PRM variants and results from the statistical tests of the significance of the differences between the variants. The test problems are two well-known benchmark problems proposed in the literature. The Hwang and Ahuja benchmark problem is a 5 degrees-of-freedom robotics motion planning problem for a SCARA-type robot [2]. The task was designed to represent a realistic but non-pathological problem for a manipulator. The task involves removing a hook from a wicket and a subsequent backtracking motion to avoid a large obstacle (see Fig. 1). No generally available geometric model for the task exists, but a difficult version of the problem called *Adept 80* [29] was produced for this study.

The second test problem is the Alpha Puzzle benchmark problem proposed by Amato *et al.* [8]. The problem is intended to represent 6 degrees-of-freedom disassembly problems and it is designed to have a narrow passage that the movable object must pass. Several versions of the Alpha Puzzle exist with varying difficultiness. This study uses Alpha Puzzle version 1.2, which is of medium difficultiness. The original Alpha Puzzle problem involves separating the two intertwined α-shaped loops. The loops can be intertwined in two different ways with the prongs of the loops either in symmetric (first image in figure 2) or anti-symmetric (last image in figure 2) orientations. Since the intention of this paper is to evaluate the performance of the various strategies in capturing the full connectivity of the test problems, both intertwined configurations together with a separated configuration (middle image in figure 2) are inserted as seed configurations to the roadmap at the beginning of the construction.

The roadmap construction with the PRM variants is continued until a roadmap connecting all the seed configurations of the task is obtained and the number of performed collision checks is taken as the run-cost measure. The grid approximation of *cspace* for A^* search has resolution of 128 points for each degree-of-freedom for the Hwang and Ahuja benchmark problem and 512 steps for the Alpha Puzzle problem.
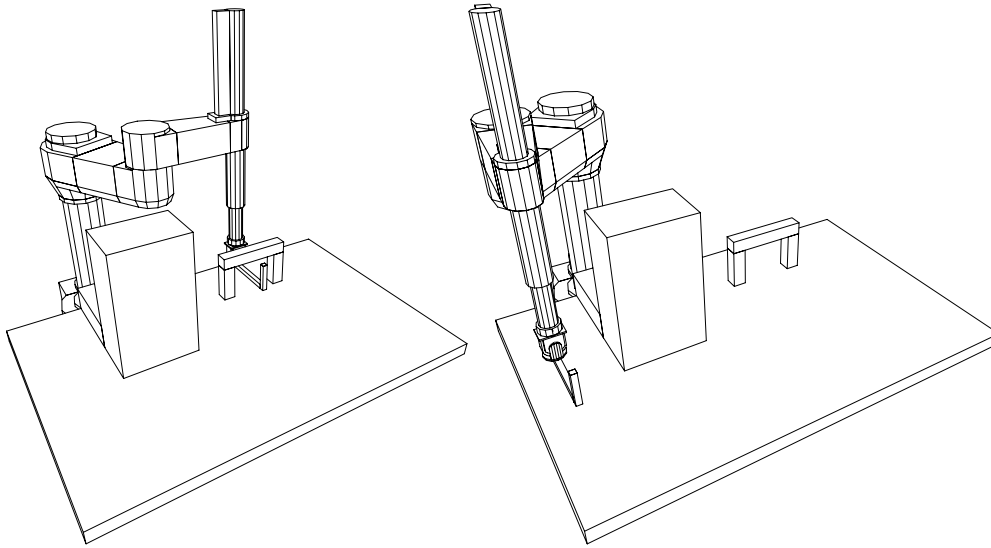
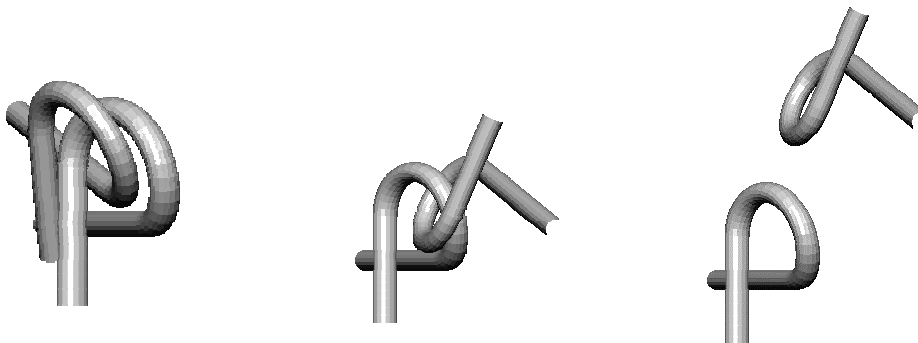**Figure 1:** *The seed configurations for a version of the benchmark problem proposed by Hwang and Ahuja.*

Figure 2: *The seed configurations for the Alpha Puzzle 1.2 benchmark task.*

5. Empirical Results

Table 1 gives descriptive statistics for the experiments with the PRM variants. The table gives run-cost mean, standard deviation and the coefficient of variation for a sample of 240 runs. The coefficient of variation expresses standard deviation as a percentage of mean, so it can reveal if the standard deviation changes together with the mean.

As can be seen in table, PRM-G and PRM-L have considerably better performance than PRM-C both in terms of mean run-cost and the standard deviation of the run-cost for the version of the Hwang and Ahuja benchmark problem. PRM-L not only has the absolute standard deviation improved but also the coefficient of variation is smaller. This indicates that not only has the scale of the run-cost distribution changed but also its shape. Ryan-Einot-Gabriel-Welsch multiple comparison procedure [31] can declare the difference between static (PRM-C) and adaptive

(PRM-G, PRM-L) planner variants statistically significant at α=0.01 (experimentwise), but fails to detect statistically significant difference between the two proposed heuristics. Levene's test of homogeneity of variances [32] detects a very significant difference in the standard deviations (p=0.0071). For the Alpha Puzzle 1.2 problem the results are not as good. There is a small improvement in standard deviation when using the proposed heuristics, but that difference is not statistically significant. Neither are the differences in the means.

The failure of the heuristics to yield statistically significant differences for the Alpha Puzzle 1.2 may be explained by the fact that it is very difficult to generate "good" subgoal samples for this task. Finding critical samples for this problem from a pseudo-random sampling sequence is a rare event and the increase in the capability of the local planner fails to make it sufficiently more frequent. It can be stated that the behavior of the planners on this test problem is determined by the "narrow passage" nature of the problem, as samples are required in a small bottleneck area in the *cspace*. Narrow passages are a well-known difficulty for all PRM type planners [4] and the proposed heuristics does not overcome this problem.

6. Conclusions And Future Work

This paper introduced the run-cost variance of randomized motion planners as a distinct research issue. In future research in randomized motion planners the run-cost variance should be considered as an important performance characteristic of the planner. Descriptive statistics and statistical tests used in this paper will provide researchers with tools to address the issue of variance and strengthen the research methodology.

As a positive example of possibilities for reducing the run-cost variance of a randomized motion planner, simple heuristic techniques were introduced and investigated empirically. A typical analysis of the run-cost means shows that the techniques can provide a significant improvement. An analysis of the run-cost variance shows an additional benefit of the techniques and helps to select between two techniques that have similar expected run-cost. But like all heuristics, also the techniques presented here may fail to provide the expected benefit.

The heuristic techniques for variance reduction presented in this paper can also be used with other search procedures that have a means to control the extent of search in the local planner. Possible search procedures include randomized ones such as RPP [16] and RRT [22]. But due to the lack of a theoretical model of the phenomena, the effectiveness of the proposed techniques with other search procedures must be investigated empirically.

Acknowledgments

References

[1]   J. C. Latombe, Robot Motion Planning, Kluwer Academic Publishers, Norwell, Mass. 1991.

[2]   Y. K. Hwang, N. Ahuja, Gross Motion Planning - A Survey, ACM Computing Surveys, Vol. 24, No. 3, Sep. 1992, 219-291.

[3]   K. Gupta, A. P. del Pobil (eds.), Practical Motion Planning in Robotics: Current Approaches and Future Directions, John Wiley & Sons, West Sussex, 1998.

[4]   J.-C. Latombe, Motion Planning: A Journey of Robots, Molecules, Digital Actors, and Other Artifacts, Int. Journal of Robotics Research, Vol. 18, No. 11, November 1999, 1119-1128.

[5]   L. Kavraki, J.-C. Latombe, Randomized Preprocessing of Configuration Space for Fast Path Planning, Proc. of the 1994 IEEE Int. Conf. on Robotics and Automation, IEEE, 2138-2145.

[6]   M. H. Overmars, P. Svestka, A Probabilistic Learning Approach to Motion Planning, Technical Report UU-CS-1994-03, Department of Computer Science, Utrecht University, The Netherlands, January 1994.

[7]   Th. Horsch, F. Schwarz, H. Tolle, Motion Planning with Many Degrees of Freedom - Random Reflections at C-Space Obstacles, Proc. of the 1994 IEEE Int. Conf. on Robotics and Automation, IEEE, 3318-3323.

[8]   N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, D. Vallejo, OBPRM: An Obstacle-Based PRM for 3D Workspaces, Workshop on Algorithmic Foundations of Robotics, 1998, 155-168.

[9]   C. Holleman, L. Kavraki, A Framework for Using the Workspace Medial Axis in PRM Planners, Proc. of the 2000 IEEE Int. Conf. on Robotics and Automation, IEEE.

[10]   B. R. Donald, A Search Algorithm for Motion Planning with Six Degrees of Freedom, Artificial Intelligence, Vol. 31, No. 3, March 1987, 295-353.

[11]   K. Kondo, Motion Planning with Six Degrees of Freedom by Multistrategic Bidirectional Heuristic Free-Space Enumeration. IEEE Transactions on Robotics and Automation, Vol. 7, No. 3, June 1991, 267-277.

[12]   D. E. Koditschek, Exact Robot Navigation by Means of Potential Functions: Some Topological Considerations, Proc. of the 1987 IEEE Int. Conf. on Robotics and Automation Conf., IEEE, 1-6.

[13]   F. Lamiraux, L. E. Kavraki, Planning Paths for Elastic Objects under Manipulation Constraints, Int. Journal of Robotics Research, Vol. 20, No. 3, 2001, 188-208.

[14]   B. Glavina, Solving Findpath by Combination of Goal-Directed and Randomized Search, Proc. of the 1990 IEEE Int. Conf. on Robotics and Automation, IEEE, 1990, 1718-1723.

[15]   B. Glavina, A Fast Motion Planner for 6-DOF Manipulators in 3-D Environments. Proc. of the Fifth Int. Conf. on Advanced Robotics, IEEE Press, 1991, 1176-1181.

[16]   J. Barraquand, J.C. Latombe, Robot Motion Planning: A Distributed Representation Approach, Int. Journal of Robotics Research, vol. 10, no. 6, Dec. 1991, 628-649.

[17]   X. Zhu and K. Gupta, On Local Minima and Random Search in Robot Motion Planning, Unpublished Manuscript, 1993.

[18]   E. Mazer, J. Ahuactzin, G. Talbi and P. Bessiere, The Ariadne's Clew Algorithm, Journal of Artificial Intelligence Research, Vol. 9, 1998, 295-316.

[19] P. Isto, A Two-level Search Algorithm for Motion Planning, Proc. of the 1997 IEEE Int. Conf. on Robotics and Automation, IEEE Press, 2025-2031.

[20] P. Isto, Constructing Probabilistic Roadmaps with Powerful Local Planning and Path Optimization. To appear in: Proc. of the 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.

[21] D. J. Challou, D. Boley, M. Gini, V. Kumar, C. Olson, Parallel Search Algorithms for Robot Motion Planning, In [3], 115-131.

[22] S. M. LaValle, J. J. Kuffner, Jr., Rapidly-Exploring Random Trees: Progress and Prospects, Workshop on Algorithmic Foundations of Robotics, A K Peters, Wellesley, 2001, 293-308.

[23] M. Branicky, S. M. LaValle, K. Olsen, and L. Yang. Quasi-Randomized Path Planning. Proc. of the 2001 IEEE Int. Conf. on Robotics and Automation, IEEE, 1481-1487.

[24] S. M. LaValle, M. S. Branicky, On the Relationship Between Classical Grid Search and Probabilistic Roadmaps. To appear in WAFR 2002.

[25] H. Chang, T.-Y. Lai, Assembly Maintainability Study with Motion Planning, Proc. of the 1995 IEEE Int. Conf. on Robotics and Automation, IEEE, Los Alamitos, CA, 1012-1019.

[26] N. J. Nilsson, Principles of Artificial Intelligence, Springer-Verlag, Berlin, 1982.

[27] L. Kavraki, J.C. Latombe, R. Motwani, P. Raghavan, Randomized Query Processing in Robot Motion Planning, Proceeding of the ACM SIGACT Symposium on Theory of Computing (STOC), ACM, Las Vegas, 1995, 353-362.

[28] L. E. Kavraki, P. Svestka, J.C. Latombe, M. Overmars, Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces, IEEE Transactions on Robotics and Automation, Vol. 12, No. 4, 1996, 566-580.

[29] P. Isto, Adaptive Strategies for Probabilistic Roadmap Construction. Submitted to WAFR'02.

[30] P. R. Cohen, Empirical Methods for Artificial Intelligence, The MIT Press, Cambridge, 1995.

[31] I. Einot, K. R. Gabriel, A Study of the Powers of Several Methods of Multiple Comparisons, Journal of the American Statistical Association, Vol. 70, No. 351, September 1975, 574-583.

[32] R. E. Glaser, Levene's Robust Test of Homogeneity of Variances, In Kotz & Johnson (eds.), Encyclopedia of Statistical Sciences, Vol 4, John Wiley & Sons, 1983, 608-610.

	Hwang and Ahuja problem			Alpha Puzzle 1.2		
	Mean	Std. Dev.	Coeff. Var.	Mean	Std. Dev.	Coeff. Var.
PRM-C	2,021,367	3,308,323	164	1,431,266	1,061,356	74
PRM-G	804,865	1,357,438	169	1,473,386	924,775	62
PRM-L	816,762	800,089	98	1,357,358	921,015	68

Table 1: *The mean and standard deviation of the run-cost in collision checks and the coefficient of variation for the problems of figure 1 and figure 2. The sample size is 240 runs.*