Proceedings of the 2002 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
EPFL, Lausanne, Switzerland • October 2002

Constructing Probabilistic Roadmaps with Powerful Local Planning and Path Optimization

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

Abstract

This paper describes a new approach to probabilistic roadmap construction for path planning. The novel feature of the planner is that it uses a powerful local planner to produce highly connected roadmaps and path optimization to maintain the rapid query processing by a fast local operator. While most previous approaches obtain good roadmaps by advanced sampling methods, the presented approach concentrates on the method used to connect the samples. Empirical results show that the new approach outperforms the more traditional approach of using fast local planners in capability to produce roadmaps with only few connected components. Statistical analysis is used to identify features important for the efficiency of the local planners.

1. Introduction

The computation of a collision-free path for a rigid or articulated object among obstacles is an important but computationally hard problem. Most of the research in path planning has been conducted for applications in robotics, but path planners have also numerous other applications [1]. A considerable number of path planners have been proposed and a large body of literature has been accumulated during the last few decades of research [2][3]. Since the path planning problem is PSPACE-hard, heuristic planners are needed to solve real-life path planning problems within practicable time limits.

Some robotics applications such as spot welding and riveting call for a large number of paths in a static or slowly changing environment. In such a case, it may be advantageous to invest some computational resources up-front, if the subsequent paths can be planned with less effort. The probabilistic roadmap (PRM) is one such two-stage path planning technique. The first stage of PRM planning produces a probabilistic approximation of the connectivity of the collision-free configuration space in the form of a graph. This preprocessing stage consists of sampling the free configuration space (*cspace*) with a suitable sampling strategy and using a very fast local planner to connect the samples. Once the graph approximation has been constructed, path planning queries can be answered by connecting the start and goal configurations to the roadmap graph and searching the roadmap for a solution. If the roadmap represents the *cspace* sufficiently well, the queries are very fast, since they only require calling a fast local planner and searching the roadmap.

Theoretical analysis and experiments have shown that PRM approach can be very efficient, if the queries do not require passing through narrow corridors. If the *cspace* contains narrow passages, a large number of samples are needed to construct a roadmap with paths through the passages. This issue has become known as the "narrow passage" problem [1]. Various sampling and connection strategies have been proposed to get a better approximation of the true connectivity of the *cspace*.

This paper describes a new approach to probabilistic roadmap construction. The novel feature of the planner is that it uses a powerful local planner to connect the samples and transforms the path segments into a form that can be rapidly reconstructed during the query stage with a simple local operator. Thus, the queries can be answered rapidly without replanning the path segments with the potentially quite costly local planner. The method is orthogonal to the usual approaches of heuristic sampling and visibility graph connection strategy.

In this paper the proposed approach is compared to the traditional approach using empirical methods [4] and established procedures from design and analysis of experiments [5]. Results with a number of problems show that powerful local planners outperform more traditional local planners in their ability to produce accurate roadmaps within given resource limits. The differences in performance are statistically significant. Practical significance of the proposed method is demonstrated by its ability to solve Alpha Puzzle 1.0 benchmark problem. Furthermore, important features of the local planners are identified with hypothesis testing.

The next section presents an overview of the previous research in PRM planners. Section 3 presents the components of the proposed roadmap construction algorithm. Section 4 describes the experimental design and empirical results. Section 5 discusses the results and section 6 presents conclusions.

2. Probabilistic Roadmap Planners

The probabilistic roadmap planners build the roadmap by iteratively sampling the *cspace* and connecting the samples with some local method [6][7][8]. Since purely random sampling strategy performs poorly in the presence of narrow passages, a number of heuristic sampling techniques have been developed. For example, the geometric properties of the workspace can be used to guide sampling [7].

A general approach to better sample the difficult regions of the *cspace* is to use some property of the *cspace* to guide sampling. Resampling can be done in regions of *cspace* that have samples with few connections [6]. Samples at or near *cspace* obstacle surfaces can be used to increase the probability that samples will be generated in the difficult regions, especially inside the narrow passages [9][10][11][12]. The method is quite effective in many cases, but it is easy to recognize problems that break this approach. Consider the workcell in figure 3. If one of the robots has inserted the payload through the gate, the combined system is near a *cspace* obstacle surface. However, there is still the passage for the other robot through the second gate to be found.

The samples can be generated on the medial axis either in the *cspace* or in the workspace [13][14]. Since the medial axis is 2-equidistant to the obstacles (in workspace or *cspace*), sampling at it increases the probability of finding a path through cluttered areas. The computation of medial axis in the *cspace* is costly and the generalization of the workspace medial axis method to more complicated workspaces is non-trivial (consider again figure 3).

The various components of a PRM planner can have a significant performance impact. Amato *et al.* present exploratory studies of the performance of a number of distance metrics, local planners and connection strategies for rigid bodies in cluttered environments [15][10]. A visibility criterion can be used to limit the number of nodes in the PRM and thus reduce the construction time [16].

Chen presents an incremental learning algorithm that bears similarity to the PRM approach [17]. The algorithm produces a sparse collection of subgoals to form an experience graph. The experience graph is constructed while solving planning tasks by augmenting it with paths generated by a powerful planner, but abstracted into a sequence of subgoals that can be connected by a fast planner to solve the same task. The abstraction step is conceptually quite similar to the transformation performed in the proposed PRM construction method, although the proposed method uses interpolation to reconstruct the paths rather than a proper path planner.

With the exception of the component reduction stage in [6], PRM planners typically use very simple local planners. This stems from the fact that the local planner is used in queries to reconstruct the roadmap connections. However, there is no obligation for the use of the local planner during query stage, if the connections between nodes can be reconstructed by other means.

3. Roadmap Construction

The roadmap construction algorithm has several components. A sampling strategy must be used to place collision-free points in the *cspace*. A set of point pairs must be selected as candidate edges in the roadmap graph. These pairs are given to the local planner that attempts to generate a path segment between the points. If the local planner is successful, the path segments are transformed into a form that can be rapidly reconstructed by the local operator during the query stage.

The sampling strategy used in the experiments is the most basic one of pseudo-random sampling until a stopping criterion is met. Candidate sample pairs are produced by selecting for the new sample up to $k=10$ closest nodes from each connected component of the roadmap at the sample generation time. The aim is to facilitate generation of a roadmap with few components rather than to minimize run-time. Euclidean distance is used as the distance metric as it has been shown to have good performance with minimal computational cost [15].

The powerful local planner used in the experiments originates in the resolution complete free-space enumeration algorithm presented by Kondo [18]. The planner has been modified into an incomplete local planner [19]. The critical difference is the use of a cut-off rule to discontinue the execution of the planner to allow the planner to fail in a controlled manner. The heuristics employed here use Manhattan distance with fixed weights for the degrees-of-freedom instead of Kondo's Euclidean distance with randomized weights.

The local planner uses bi-directional A^* algorithm guided by four different heuristics to search a grid representation of the *cspace*. The heuristic evaluation of a configuration uses the guiding function $f(C) = A \times g(C) + B \times h(C)$, where $A=3$, $B=1$, and $g(C)$ is the distance from the start configuration to the current configuration C in grid steps (Manhattan distance). Function $h(C)$ is an estimate for the cost from the evaluated configuration to the target configuration of the local planner:

$$h(C) = \left(\sum_{i=1}^{DOF} a_i D_i(C,T) - \rho a_j \right)$$

$D_i(C,T)$ is the distance between the current configuration and the target configuration T in grid steps along axis i. The tie-breaking term ρ has a value of 0.5, if the evaluated node was expanded in the same direction as the parent node along the axis j or 0 otherwise.

Based on previous research [20], the following four weight sets are used in the estimate:

"Manipulator" heuristics:

$$a_i = \left\lceil 9\frac{DOF+1-i}{DOF} \right\rceil, \quad i = 1,\ldots,DOF.$$

"Position" heuristics:

$$a_i = \begin{cases} 9, & i = 1,\ldots,d \\ 1, & i = d+1,\ldots,DOF \end{cases}$$

"Rotation" heuristics:

$$a_i = \begin{cases} 1, & i = 1,\ldots,d \\ 9, & i = d+1,\ldots,DOF \end{cases}$$

"Even" heuristics:

$$a_i = 5, \quad i = 1,\ldots,DOF.$$

$d = \lfloor (DOF + 0.5)/2 \rfloor$.

The heuristics are executed in round-robin fashion and the efficiency of each heuristics determines the number of node expansions it is allocated. Complete details of the scheduling between the heuristics can be found elsewhere [19]. In order to control the minimal efficiency of the local planner an evaluation value is calculated for each examined configuration:

$$O_t(C) = \frac{\sum_t F_t(C)}{g(C)}$$

$F_t(C)$ is the total number of collision-free configurations examined by the heuristics t until the examination of C. If the evaluation value $O_t(C)$ for the heuristics t increases above a given threshold value O_{th}, the execution of the heuristics is discontinued for that round. If all the heuristics are discontinued, the local planner fails. Additionally, a hard limit of 10^6 configuration nodes for each call of the local planner was used in the experiments to keep the local planner from exhausting the main memory.

Once the local planner generates a path segment, it is transformed with a polygonal optimizer [21]. The optimizer deletes a configuration from the solution path if a collision-free straight-line connection can be made from the preceding configuration to the successor configuration. The remaining configuration triplets are considered as triangles and the corners of the triangles are 'cut off' as much as possible to further reduce the length of the path segment. Finally, retracting the remaining configurations toward the straight line connecting the preceding and successor configurations smoothes the path.

The optimizer returns the solution as a sequence of configurations that can be connected by straight-line interpolation. These configurations are stored with the roadmap edge. If needed at query time, the path segment is reconstructed by interpolation without performing any collision checks.

4. Empirical Results

The experiments are designed to compare the quality of the roadmaps produced by several local planners. The multi-heuristic A^* local planner with path optimization is compared with more conventional local planners. Linear *straight-line* (*SL*) and *rotate-at-½* (*RAS*) local planners are selected as base line, since they were the recommended fast local planners in the Amato *et al.* study [15]. A simple greedy search local planner (*G*) is obtained from the multi-heuristic local planner by setting $A=0$, $O_{th}=1$ and executing only the "even" heuristics. These planners have distinctively different power as ability to proceed after they make contact with a *cspace* obstacle surface. *SL* and *RAS* fail as soon as contact is made. The greedy local planner can proceed by "sliding" along the *cspace* obstacle surface as long as any of the neighboring configurations improves the heuristic estimate and no backtracking is required. The multi-heuristic local planner can escape local minima by backtracking, but the extent of backtracking is limited by the threshold value O_{th}. Two values for the threshold are used in the experiments: 2 (M_2) and 32 (M_{32}). *SL* is arbitrary set to be more powerful than *RAS* in the analysis below.

The test problems have been designed to have a *cspace* with only one connected component, but with sections that are connected by narrow passages. In order to force roadmaps that cover different sections of the *cspace*, the roadmaps are seeded by placing a predetermined node into each section of the workspaces.

The test problems with predetermined configurations are shown in figures 1-4. The first test problem is a version of the 5 DOF benchmark task proposed by Hwang and Ahuja [2]. The start and goal configurations of the problem are used to initialize the roadmap. The Alpha Puzzle family of benchmark problems is designed to represent 6 DOF disassembly tasks with a narrow passage that must be passed to solve the problem [10]. This study uses Alpha Puzzle 1.0, which is the most difficult version in the family. A high dimensionality problem is obtained with two Puma robots by combining them serially into one 12 DOF system. The *cspace* of the problem has four sections as neither, one or both of the robots have inserted the payload through the gate. Finally, a 6 DOF problem with a workspace consisting of a number of sections is constructed from a Puma type robot with a large payload.

The grid sizes for the multi-heuristic local planner are 512^{DOF} for the Alpha Puzzle 1.0 and 128^{DOF} for the other tasks. The step size for the *SL* and *RAS* was selected so that the discretations are comparable.

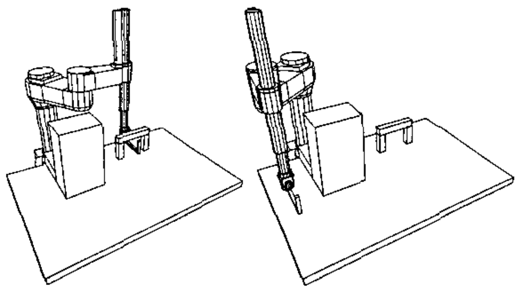


Figure 1: The seed configurations for a version of the benchmark problem proposed by Hwang and Ahuja.

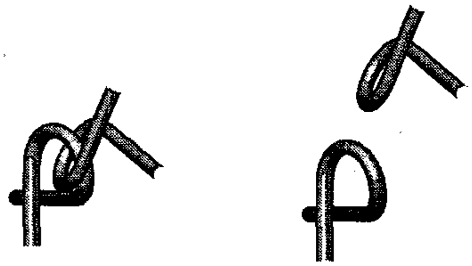


Figure 2: The seed configurations for the Alpha Puzzle 1.0 benchmark task.

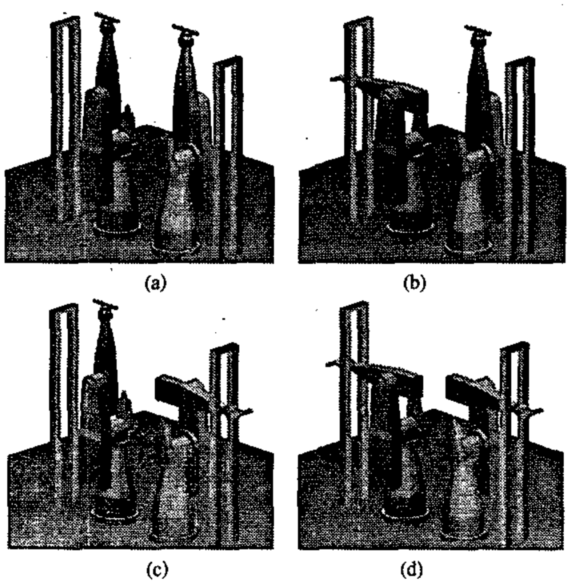


Figure 3: The seed configurations for the 12 degrees-of-freedom task with two kinematic chains.

The final roadmap should have just one connected component for each connected component of the *cspace* [22]. Therefore, the performance metrics for this study are the number of components in the final roadmap and the frequency of roadmaps that have paths between all of the seed configurations. The task for the planner is to construct a roadmap for a workcell within roadmap size limit of 1000 nodes and construction time limit of 60 minutes for the Hwang and Ahuja task and 360 minutes for the other tasks starting with the predetermined seed configurations. Fifteen replicates are produced, each with a different pseudo-random sampling sequence. When applicable in the analysis, the sampling sequences are used as a blocking factor to eliminate the variability among the sequences. The experimental design is randomized complete block design with workcell, local planner and sampling sequence as fixed factors

The average number of components in the final roadmap and the number of successfully connecting the seed configurations are presented in the table 1 for both size and time constrained preprocessing runs. Analysis of variance (ANOVA) was performed to the data with the number of final components as a response. The statistical model is significant ($p<0.0001$) and it explains 99% of the variance. The analysis reveals that the local planner is a highly significant factor ($p<0.0001$) with a highly significant interaction ($p<0.0001$) with the task in both size and time constrained experiments.

ANOVA can detect that there are significant differences among the local planners, but other statistical techniques are needed to reveal which local planners differ. A set of contrasts is used to test several focused hypotheses. The fast local planners as a group are compared to the powerful local planners as a group (*SL* and *RAS* vs. *G*, M_2, and M_{32}). Similarly, the backtracking local planners are compared to the non-backtracking local planners (M_2, M_{32} vs. *SL*, *RAS*, *G*). In order to study the significance of "sliding" performed by *G*, it is compared to *SL* and *RAS* as a group. In each case the null hypotheses of no difference between group means can be rejected in favor of an alternative hypothesis of group means differing significantly ($p<0.0001$) for both sets of experiments.

The Ryan-Einot-Gabriel-Welsch multiple comparison procedure [22] can declare a very significant ($\alpha=0.01$) difference between the means of each pair of local planners in the size constrained runs, but fails to declare the difference between the means of M_2 and M_{32} in the time constrained runs statistically significant at $\alpha=0.05$ (experimentwise). Jonckheere-Terprstra test [24] provides strong support ($p<0.0001$) for the significance of the decreasing trend in the number of final components as the local planner gets more powerful. The significance of the differences between average numbers of final components are tested separately for each task with the least significant difference test at $\alpha=0.05$ (comparisonwise) and indicated in the table 1.

Cochran Q test [25] detects highly significant ($p<0.0001$) differences among the success frequencies of the local planners for both sets of experiments. In particular, McNemar's test [25] detects significant differences between M_2 and M_{32} in both the size ($p<0.0001$) and time ($p=0.0004$) constrained experiments. Furthermore, Jonckheere-Terprstra test declares the increasing trends in success frequency highly significant ($p<0.0001$) for both sets.

5. Discussion

As expected, more powerful local planners generate fewer components and have higher success rate than weaker local planners when applied to the same samples without a run-time limit. The main result of the size-constrained experiments is an empirical confirmation for the assignment of relative power to each local planner, especially for *SL* and *RAS*.

On the other hand, the time-constrained experiments provide relevant information about the time efficiency of the local planners in producing roadmaps with few components. According to the statistical analysis, the local planner should be able to "slide". Backtracking capability further improves the time efficiency of the local planner, at least when combined with efficient heuristics to guide the search. As a group the powerful local planners are more efficient than the traditional fast local planners. Furthermore, there is a statistically significant trend of increasing efficiency as the local planner gets more powerful.

The success frequencies of the planners increase with their power. Since connecting the seed configurations requires paths through narrow space, these results indicate the effectiveness of the planners in addressing the "narrow passage" problem. The ceiling effect observed with the Hwang and Ahuja benchmark problem demonstrates that also the simple local planners can deal with relatively easy problems. The difficult problems, however, are solved only with backtracking search.

The results on the relative effectiveness of *SL* and *RAS* observed here complement the earlier results of Amato *et al.* [15]. They constrain roadmap size and use the number of connections as the performance metric. When comparing the average number of connections for size-constrained Alpha Puzzle rigid body task, the results agree, but here the difference is not statistically significant ($p=0.74$). For the robot tasks *SL* is significantly better than *RAS*, except for the time constrained 2 pumas task. In that workcell, *RAS* behaves like *SL* for each robot separately. The general observations of Amato *et al.* are similar to those presented here: powerful local planners are best for difficult problems. However, the path transformation technique presented here allows the use of local planners sufficiently powerful to solve even the most difficult problems in this study, while still maintaining rapid query processing.

The ability of the multi-heuristic local planner to solve the Alpha Puzzle 1.0 even with very simplistic sampling and connection strategies is quite significant. Traditional local planners require the use of sophisticated sampling strategy or visibility graph approach to solve the Alpha Puzzle family of benchmarks.

6. Conclusions

A new approach for probabilistic roadmap planning was presented that allows the use of powerful local planners during the roadmap construction. The approach separates the local planner used during roadmap construction and the local operator used to reconstruct the roadmap connections. Experiments with a multi-heuristic local planner and difficult problems demonstrated the effectiveness of the method in producing roadmaps that capture the connectivity of the *cspace* even in the absence of a sophisticated sampling strategy. The results indicate that the use of powerful local planners should play a role in solving the narrow passage problem.

Acknowledgments

Alpha Puzzle 1.0 was designed by Boris Yamrom, GE Corporate Research & Development Center. The model was provided by the DSMFT research group at Texas A&M University. Johannes Lehtinen provided the geometric model for the Hwang and Ahuja task. Janne Ravantti provided access to the cluster computer at the Bamford Laboratory, University of Helsinki. The author thanks Prof. Juha Tuominen for his useful comments.

References

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

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

[5] D. C. Montgomery, Design and Analysis of Experiments, John Wiley & Sons, New York, 1997.

[6] L. Kavraki, J.-C. Latombe, Randomized Preprocessing of Configuration Space for Fast Path Planning, IEEE International Conference on Robotics and Automation, IEEE, 1994, 2138-2145.

[7] 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.

[8] Th. Horsch, F. Schwarz, H. Tolle, Motion Planning with

Many Degrees of Freedom - Random Reflections at C-Space Obstacles, IEEE International Conference on Robotics and Automation, IEEE, 1994, 3318-3323.

[9] N. M. Amato and Y. Wu, A Randomized Roadmap Method for Path and Manipulation Planning, IEEE International Conference on Robotics and Automation, IEEE, 1996, 113-120.

[10] 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.

[11] V. Boor, M. H. Overmars and F. van der Stappen, The Gaussian Sampling Strategy for for Probabilistic Roadmap Planners, IEEE International Conference on Robotics and Automation, IEEE, 1999, 1018-1023.

[12] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, S. Sorkin, On Finding Narrow Passages with Probabilistic Roadmap Planners, Workshop on Algorithmic Foundations of Robotics, 1998, 141-154.

[13] S. A. Wilmarth, N. M. Amato and P. F. Stiller, MAPRM: A Probabilistic Roadmap Planner with Sampling on the Medial Axis of the Free Space, International Conference on Robotics and Automation, IEEE, 1999, 1024-1031.

[14] C. Holleman, L. Kavraki, A Framework for Using the Workspace Medial Axis in PRM Planners, IEEE International Conference on Robotics and Automation, IEEE, 2000.

[15] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, D.Vallejo, Choosing Good Distance Metrics and Local Planners for Probabilistic Roadmap Methods, IEEE Transactions on Robotics and Automation, Vol. 16, No. 4, August 2000, 442 -447.

[16] C. Nissoux, T. Simeon, J.P. Laumond, Visibility Based Probabilistic Roadmaps, IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 1999, 1316-1321.

[17] P. C. Chen, Improving Path Planning with Learning, Machine Learning 9, Morgan Kaufmann Publishers, 1992, 55-61.

[18] 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.

[19] P. Isto, A Parallel Motion Planner for Systems with Many Degrees of Freedom, International Conference on Advanced Robotics, 2001, 339-344.

[20] P. Isto, Path Planning by Multiheuristic Search via Subgoals, Proceedings of the 27th International Symposium on Industrial Robots, CEU, Milan 1996, 721-726.

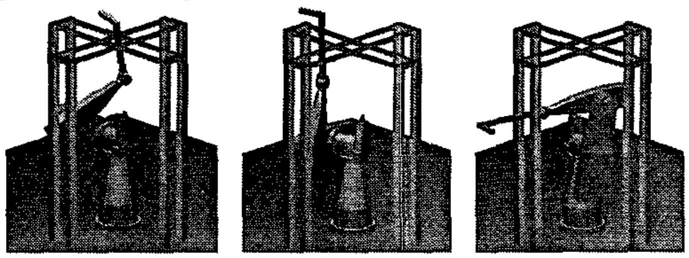
[21] S. Berchtold, B. Glavina, Scalable Optimizer for Automatically Generated Manipulator Motions, IEEE/RSJ/GI International Conference on Intelligent Robots and Systems, IEEE, 1994, 1796-1802.

[22] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, Capturing the Connectivity of High-Dimensional Geometric Spaces by Parallelizable Random Sampling Techniques. In P. M. Pardalos and S. Rajasekaran (eds.), Advances in Randomized Parallel Computing, Kluwer Academic Publishers, Boston, 159-182, 1999.

[23] 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.

[24] W. Pirie, Jonckheere Tests for Ordered Alternatives, In Kotz & Johnson (eds.), Encyclopedia of Statistical Sciences, Vol 4, John Wiley & Sons, 1983, 315-318.

[25] S. Siegel, Nonparametric Statistics, McGraw-Hill, Tokyo, 1956.



(a) (b) (c)

Figure 4: *Some of the seed configurations for the 6 degrees-of-freedom task. The total number of seed configurations is 9, since all the configurations symmetrical to (b) and (c) by rotation of the base joint are used as seeds.*

Size Constrained											Time Constrained										
	Figure 1		Figure 2		Figure 3		Figure 4		Ave.	Σ		Figure 1		Figure 2		Figure 3		Figure 4		Ave.	Σ
RAS	4.9	2	3.1	0	16.0	0	73.7	0	24.5	2	*RAS*	7.7	15	41.2	0	28.7	0	127.3	0	51.3	15
SL	3.5	10	3.2	0	14.3	0	56.1	0	19.3	10	*SL*	3.1	15	20.9	0	28.6	0	102.7	0	38.8	15
G	1.5	15	2.5	0	4.7	0	19.7	0	7.1	15	*G*	1.2	15	6.1	0	5.1	0	27.1	0	9.9	15
M_2	1.1	15	2.3	1	2.9	12	8.3	0	3.6	28	M_2	1.3	15	2.9	1	2.5	12	7.2	0	3.5	28
M_{32}	1.1	15	1.6	7	2.3	15	2.2	15	1.8	52	M_{32}	1.7	15	2.0	6	1.9	10	2.2	15	2.0	46

Table 1: *Average number of final components and the number of successful merges of the seed configurations into the roadmap. Grand averages and total sums over all tasks are presented for the final number of components and the number of successes, respectively. For the averages, grid lines are removed between values that are not significantly different at α=0.05, comparisonwise for task averages and experimentwise for the grand averages. Sample size is 15 runs.*