

PLANEJAMENTO DE CAMINHO USANDO BI-ARW MELHORADO E MAPA DE ROTAS

BRUNO VILHENA ADORNO*, GEOVANY ARAÚJO BORGES*

**Laboratório de Robótica e Automação (LARA)*

*Departamento de Engenharia Elétrica (ENE) - Universidade de Brasília (UnB)
Caixa Postal 04591 - CEP 70910-900 - Asa Norte - Brasília - Brasil*

Emails: bvabr@yahoo.com.br, gaborges@ene.unb.br

Abstract— This paper proposes new features for an hybrid algorithm based on Probabilistic Roadmap (PRM) and Adaptive Random Walks (ARW) for path planning problem and its good performance was presented in previous works. Results show that the resultant algorithm has improved capacity of the configuration space exploration, low execution time and generates high quality paths.

Keywords— Path planning, mobile robotics, probabilistic roadmaps, adaptive random walks

Resumo— Este artigo propõe extensões para um algoritmo híbrido baseado no Mapa de Rotas Probabilístico (PRM) e Passeio Aleatório Adaptativo (ARW) visando resolver de forma eficiente o problema de planejamento de caminho para robôs móveis. Seu bom desempenho já havia sido apresentado em trabalhos anteriores mas resultados obtidos com o algoritmo estendido mostram a melhora em sua capacidade de exploração do espaço de configuração, baixo tempo de execução e ainda geração de caminhos com alta qualidade.

Palavras-chave— Planejamento de caminho, robótica móvel, mapa de rotas probabilístico, passeios aleatórios

1 Introdução

O problema de planejamento de caminho em robótica móvel consiste em, dadas as configurações iniciais e finais do robô, descobrir uma seqüência de configurações intermediárias que permitam que o robô saia da primeira e chegue à segunda sem colidir com obstáculos.

Este problema tem se mostrado difícil, pois sua solução requer um tempo exponencial em relação ao número de graus de liberdade do robô, sendo que as soluções determinísticas e completas se mostraram computacionalmente impraticáveis para robôs com elevado número de graus de liberdade (Barraquand et al., 1997; Kavraki et al., 1996).

Sendo assim, o foco das pesquisas tem sido as abordagens que aproximam o espaço de configuração \mathcal{C} por meio de amostras do mesmo, sendo que muitas das técnicas desenvolvidas desde a década de 90 se mostraram computacionalmente eficientes (Kavraki et al., 1996; J. J. Kuffner and LaValle, 2000; Hsu, 2000; Bekris et al., 2003; Carpin and Pilonetto, 2005b).

Dessa forma, a utilização dos algoritmos atuais extrapolou o universo da robótica móvel e alcançou outras áreas de pesquisa, tais como animação de atores artificiais, biologia molecular, problemas de montagens complexas e planejamento de movimento para objetos flexíveis (Choset et al., 2005).

Muitas vezes o planejamento de caminho ocorre no espaço de configuração do robô, e não no espaço de trabalho \mathcal{W} (e.g. quando o número de graus de liberdade do robô é maior que a dimensão do espaço de trabalho) exigindo assim uma boa formulação para o problema, pois

o alcance da solução pode consistir na busca em um espaço de dimensão arbitrária. Sendo assim, de uma forma mais geral a solução para o problema de planejamento de caminho é uma função contínua $c \in C^0$ tal que $c : [0, 1] \rightarrow \mathcal{C}_{free}$, em que $c(0) = q_{start}$, $c(1) = q_{end}$ (q corresponde à configuração do robô) e \mathcal{C}_{free} corresponde à porção do espaço de configuração que está livre de obstáculos (Choset et al., 2005; Carpin and Pilonetto, 2005a).

Além disso, os algoritmos de planejamento de caminho baseados na amostragem do espaço de configuração podem ser classificados em dois grupos: questionamento único e múltiplos questionamentos. No primeiro grupo encontram-se algoritmos que acham a solução à medida que as configurações inicial e final são entregues ao planejador (J. J. Kuffner and LaValle, 2000; Hsu, 2000; Carpin and Pilonetto, 2005b). Já no segundo grupo existe uma etapa de pré-processamento do espaço de configuração visando o aprendizado do mesmo, o que é feito por meio da criação de alguma estrutura que represente uma aproximação de \mathcal{C}_{free} (e.g. mapa de rotas) e uma etapa posterior onde o objetivo é fazer múltiplos questionamentos, sendo estes realizados por meio de uma simples busca na estrutura criada na etapa anterior (Kavraki et al., 1996). Existem ainda abordagens híbridas que se mostraram eficientes no problema de planejamento de caminho (Bekris et al., 2003; Adorno and Borges, 2006).

O restante do artigo está organizado da seguinte forma: a Seção 2 revisa brevemente algumas características dos algoritmos ARW, PRM e MRPAA. A Seção 3 apresenta algumas extensões úteis para o algoritmo ARW, enquanto que a Se-

ção 4 mostra as melhorias aplicadas ao algoritmo MRPA. Já a Seção 5 mostra os resultados obtidos a partir do MRPA melhorado e por fim a Seção 6 apresenta as conclusões e propostas de trabalhos futuros.

2 Motivação

Em Adorno and Borges (2006) foi apresentada uma abordagem híbrida que utiliza o algoritmo ARW (do inglês, *Adaptive Random Walks*) (Carpin and Pilonetto, 2005b) para criar um mapa de rotas a ser reaproveitado em questionamentos posteriores. Sendo assim, foi mostrado que inicialmente o algoritmo funcionava como o ARW, mas depois de um número suficientemente grande de questionamentos ele passava a atuar como um PRM (do inglês, *Probabilistic Roadmap*) (Kavraki et al., 1996) em sua fase de questionamento. Os autores atribuíram o bom desempenho do algoritmo ao fato dele explorar as melhores características dos algoritmos base.

De fato, uma grande vantagem do ARW é o pequeno número de parâmetros a serem ajustados, mesmo para mapas com topologias bem diferentes umas das outras (Carpin and Pilonetto, 2005b). Em sua versão mais básica ele consiste em gerar um passeio aleatório que sai da configuração inicial e tenta alcançar a configuração final. Contudo, cada nova configuração do passeio é gerada aleatoriamente de acordo com uma distribuição gaussiana de média nula e matriz de covariâncias que se adapta de acordo com a região de \mathcal{C} em que o passeio se encontra. Além disso este algoritmo possui garantia de convergência para a solução (Carpin and Pilonetto, 2005b).

Por outro lado, o PRM já se mostrou eficiente para resolver problemas com um grande número de graus de liberdade (5 a 12), além de ser de fácil implementação (Choset et al., 2005). Este algoritmo possui duas etapas: aprendizado e questionamento.

Na etapa de aprendizado o objetivo é amostrar configurações aleatórias em \mathcal{C}_{free} e armazená-las na forma de nós N em um grafo R . À medida que as configurações são amostradas elas são conectadas às configurações vizinhas se não houver obstáculos entre elas. Essa verificação é feita por meio de um planejador local e a conexão é representada por meio de uma borda E .

Na etapa de questionamento o objetivo é, dadas as configurações inicial e final do robô, tentar conectá-las ao grafo R e, caso isso seja possível, achar um caminho no grafo que as conecte. Sendo assim, o objetivo do PRM é concentrar o processamento na etapa de aprendizado, de forma que os questionamentos possam ser feitos de maneira bem rápida.

Dessa forma, a grande vantagem do algoritmo proposto em Adorno and Borges (2006) é que não

existe etapa de aprendizado como no PRM e à medida que muitos questionamentos são feitos o mapa de rotas tende a cobrir todo o espaço livre. Os autores mostraram que o desempenho do algoritmo híbrido é melhor que o desempenho dos algoritmos base.

Contudo, existem algumas características a serem melhor exploradas, tanto em políticas de melhor utilização do grafo como na melhoria da capacidade de exploração do espaço de configuração proporcionada pelo ARW. Além disso, para amenizar o efeito da baixa qualidade dos caminhos que em geral são gerados pelos métodos que amostram \mathcal{C}_{free} , torna-se necessária a implementação de um método de suavização que além de tirar as redundâncias e ciclos desnecessários do caminho gerado, ainda aumente a distância dos obstáculos de uma distância pré-definida.

3 Extensões aplicadas ao ARW

Apesar do ARW ter se mostrado um algoritmo com bom desempenho computacional, ele não possui bom desempenho em regiões do espaço de configuração que possui passagens estreitas (Carpin and Pilonetto, 2005a). Dessa forma, Carpin and Pilonetto (2005a) propuseram melhorias a esse algoritmo, mas elas não mostraram bons resultados ou foram muito dependentes da topologia do espaço de configuração.

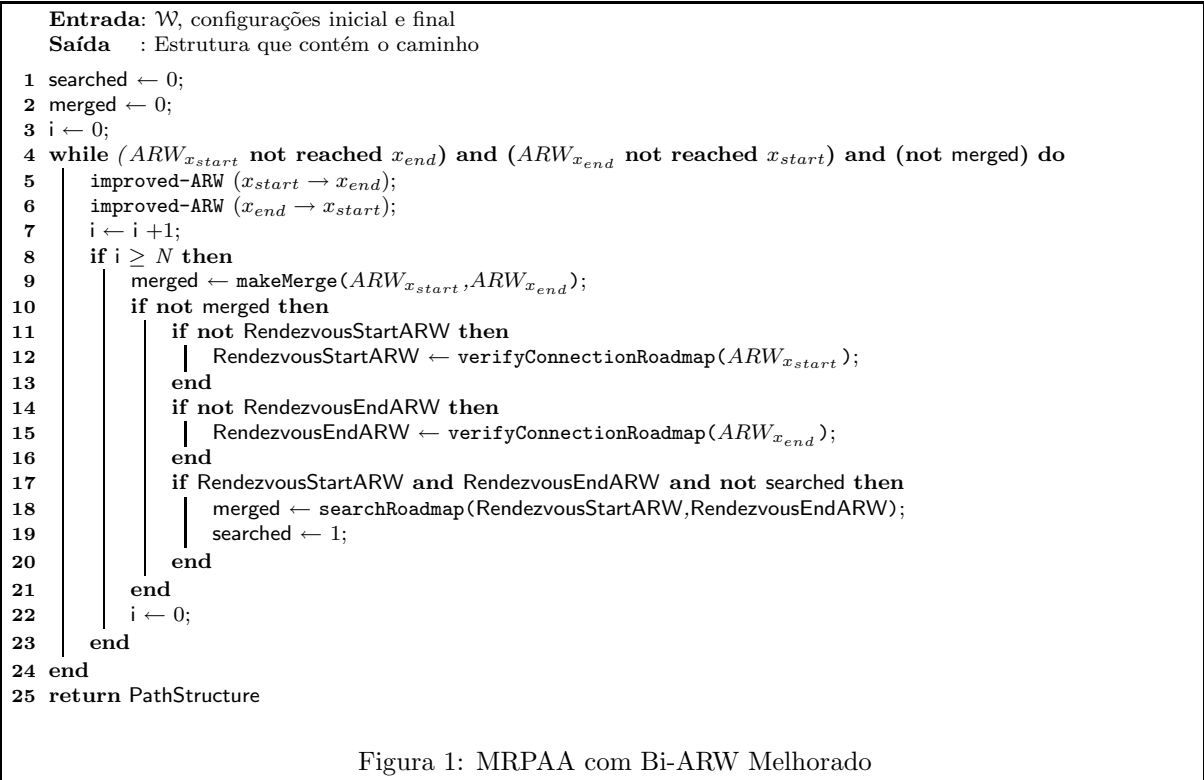
3.1 Seleção de candidatos

Em Carpin and Pilonetto (2005b) as amostras geradas eram aceitas simplesmente se elas estivessem em \mathcal{C}_{free} . Contudo, o que acontece neste tipo de abordagem é que muitas vezes o passeio aleatório passa muitas vezes pelo mesmo lugar, desperdiçando tempo e recursos computacionais.

Visando maximizar a exploração do ambiente, propõe-se que candidatas à nova configuração do passeio sejam geradas e somente a melhor delas seja escolhida, caso esteja em \mathcal{C}_{free} . A melhor candidata é aquela que possuir menor número de vizinhas, pois significa que ela foi amostrada numa região ainda pouco explorada.

3.2 Fusão dos passeios aleatórios

Em sua formulação inicial o ARW previa que a fusão dos dois passeios só deveria ocorrer caso as duas últimas configuração dos dois passeios pudessem ser conectadas pelo planejador local. O objetivo disso era tentar manter constante o tempo necessário para gerar uma nova amostra. Contudo, o que ocorre é que muitas vezes os passeios aleatórios se desencontram, sendo que a última configuração de um passeio aleatório consegue encontrar o rastro da outra antes de encontrar sua última configuração.



Dessa forma, visando melhorar este mecanismo de fusão, periodicamente o algoritmo tenta fazer a conexão entre a última configuração de um passeio e um número fixo de configurações escolhidas aleatoriamente no outro passeio.

3.3 Geração de amostras baseadas em informações extraídas do espaço de trabalho

O ARW original utiliza amostragem gaussiana para geração de novas configurações. Contudo, a distribuição gaussiana pode fazer com que o algoritmo passe um bom tempo explorando longos corredores no espaço de configuração sem conseguir sair destas regiões caso a única opção seja alguma passagem estreita no meio destes corredores.

Sendo assim, visando aumentar a explorabilidade do espaço de configuração, optou-se por outra técnica que amostra novas configurações baseada em informações do próprio espaço de trabalho \mathcal{W} . De fato, a idéia de utilizar informações do espaço de trabalho para amostrar no espaço de configuração não é nova (Berg and Overmars, 2004; Kurniawati and Hsu, 2004).

Neste trabalho, o espaço de trabalho é discretizado em torno do robô de acordo com uma região cujo tamanho é adaptativo, de forma que uma função de probabilidade possa ser levantada com informações extraídas de \mathcal{W} para então uma amostra ser gerada em \mathcal{C} .

O tamanho da região é adaptado de acordo com a variância do passeio aleatório e é dado por

$$D = 2 \sum_{i=1}^N \sigma(i), \quad (1)$$

em que $\sigma(i)$ é o desvio padrão para o eixo i do espaço de configuração do robô e D é o tamanho do intervalo a ser amostrado para todos os eixos do espaço de configuração.

Sendo assim, dado um vetor $cell[N]$ com as N células na região vizinha do espaço de trabalho ocupado pelo robô (sendo que N vai depender do tamanho da região), a probabilidade de uma célula $cell[i]$ ser sorteada é dada por

$$Pr_{cell[i]} = \frac{cell[i]}{acc}, \quad (2)$$

em que acc e $cell[i]$ são dadas por

$$acc = \sum_{i=1}^{i=N} cell[i], \quad (3)$$

$$cell[i] = \begin{cases} 1, & \text{se } cell[i] \in \mathcal{W}_{free} \\ 0, & \text{c.c.} \end{cases}$$

Uma vantagem deste método é que regiões de \mathcal{C}_{free} em geral possuem uma maior chance de serem amostradas em relação à amostragem uniforme ao mesmo tempo que há uma maior dispersão que na amostragem gaussiana. Assim há um menor desperdício de amostras, uma vez que aquelas que não se encontram em \mathcal{C}_{free} teriam que

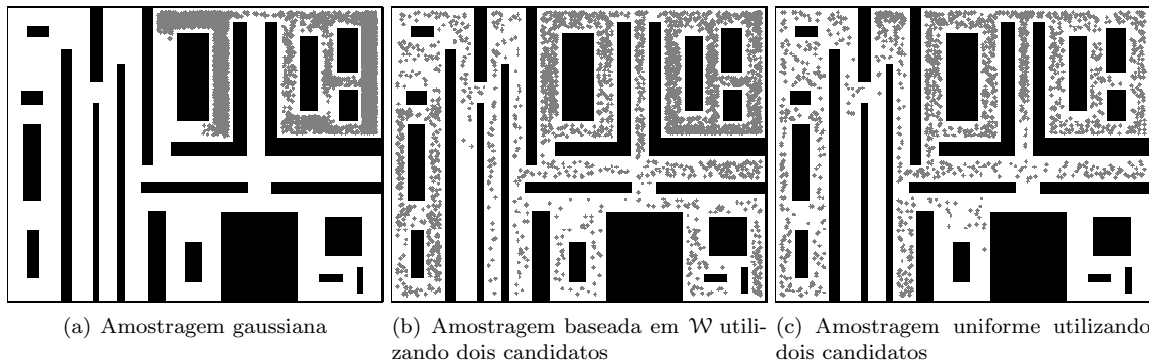


Figura 2: Exploração do ambiente

ser descartadas. Já uma desvantagem evidente é que apenas os graus de liberdade correspondentes à translação do robô podem ser amostrados facilmente com esta abordagem, uma vez que nem sempre é trivial fazer um mapeamento de \mathcal{W} em \mathcal{C} caso a dimensão deste espaço seja maior que a daquele. Além disso, esta discretização tem que ser grande o suficiente para que o planejador seja de resolução completa (Choset et al., 2005).

3.4 Suavização do caminho resultante

Uma característica presente em praticamente todos os planejadores de caminho baseados em amostragem do espaço de configuração é que o caminho resultante possui muitas redundâncias e ciclos desnecessários, havendo então a necessidade de suavização (Geraerts and Overmars, 2004; Geraerts and Overmars, 2006; Geraerts, 2006).

Em (Carpin and Pillonetto, 2005b) foi utilizado o método “dividir e conquistar” para reduzir o tamanho dos caminhos gerados pelo ARW. Contudo, em muitos casos ainda podem restar ciclos e ainda os caminhos podem passar muito perto dos obstáculos, o que em algumas aplicações pode ser inaceitável (e.g. robótica móvel).

Sendo assim, foi implementada uma busca de atalhos no caminho por meio de um método ganancioso (Choset et al., 2005). Neste método o caminho é percorrido e para cada configuração atual há a tentativa de conectá-la a uma configuração o mais próxima possível da final.

Por último, visando afastar o caminho dos obstáculos foi implementado um algoritmo semelhante ao proposto em Geraerts and Overmars (2004), cuja idéia básica é levar as configurações para o eixo médio de \mathcal{C}_{free} numa etapa posterior à construção do caminho.

De fato, a idéia de amostrar no eixo médio de \mathcal{C}_{free} não é nova e já tinha sido implementada em Wilmarth et al. (1999), porém na própria geração de configurações aleatórias.

4 Melhorias aplicadas ao MRPA

O algoritmo Mapa de Rotas criado por Passeio Aleatório Adaptativo (MRPAA) proposto em Adorno and Borges (2006) se mostrou eficiente ao usar o ARW para explorar o ambiente e armazenar os caminhos encontrados em um mapa de rotas. Contudo, este algoritmo usava um ARW bidirecional padrão e ainda não possuía políticas muito claras de utilização do grafo. Visando melhorar a explorabilidade do MRPAA e a qualidade dos caminhos resultantes, o ARW bidirecional padrão foi substituído por um com as extensões apresentadas na Seção 3.

Além disso, foram definidas políticas mais claras de utilização do mapa de rotas, pois em Adorno and Borges (2006) a idéia era tentar ligar as configurações inicial e final ao mapa de rotas. Caso isso não fosse possível ou se não houvesse um caminho no grafo, um ARW bidirecional padrão era acionado. O objetivo era chegar a uma solução do problema de planejamento de caminho e armazená-la no mapa de rotas para uso posterior. Já neste trabalho é feita uma verificação de caminho no grafo periodicamente de forma a tentar fazer uma fusão com o mesmo, de uma maneira semelhante ao que é feito na fusão dos dois passeios aleatórios apresentada na Seção 3.2.

Sendo assim, o algoritmo tenta conectar a última configuração armazenada em um passeio aleatório ao grafo. Caso isso seja possível, o ponto de encontro é armazenado e não há mais tentativa de fusão entre este passeio aleatório e o grafo durante o questionamento de caminho em questão. Se os dois passeios aleatórios puderem se conectar ao grafo, então uma busca é feita no mesmo para tentar achar um caminho entre os dois pontos de encontro. Caso isso seja possível, uma solução foi encontrada.

Contudo, apenas o trecho referente aos pontos de partida dos passeios até os respectivos pontos de encontro no grafo são suavizados e armazenados. Com isto há uma redução no tempo de execução do algoritmo, pois há aproveitamento de soluções descobertas em questionamentos anterior-

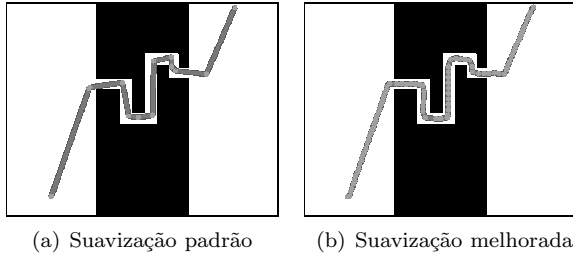


Figura 3: Suavização do caminho resultante

res, assim como também há redução no caminho a ser pós-processado.

O algoritmo da Figura 1 mostra o MRPAA antes de entrar na etapa de suavização. As linhas 5 e 6 fazem a evolução dos passeios que saem da configuração inicial e final, respectivamente. O parâmetro N da linha 8 indica a periodicidade em que haverá tentativa de fusão, sendo que na linha 9 é feita a tentativa de fusão entre os dois passeios e da linha 10 até a linha 21 é feita a tentativa de fusão dos passeios com o mapa de rotas.

5 Resultados

Os resultados foram avaliados em simulação utilizando espaços de trabalho bidimensionais e um robô planar holonômico. A execução do ambiente de avaliação foi feita em um computador compatível com IBM-PC, 1GB de memória RAM e processador Pentium 4 de 3,06GHz.

A análise foi dividida em duas partes, sendo uma qualitativa e outra quantitativa. Na etapa qualitativa é avaliada a explorabilidade de diferentes tipos de amostragem. Já na quantitativa são avaliados os tempos de execução em diversos tipos de espaço de trabalho, sendo que alguns não são mostrados neste trabalho por falta de espaço.

Além disso, como o mecanismo principal de construção do mapa de rotas é basicamente o passeio aleatório adaptativo, então a avaliação de tempo de execução será feita através da comparação entre o BI-ARW padrão e várias configurações do BI-ARW melhorado.

O intervalo de amostragem da distribuição uniforme utilizada para avaliação dos resultados também é adaptativo e é dado pela Equação 1.

Na Figura 2 observa-se que a explorabilidade proporcionada tanto pela distribuição uniforme quanto pela função de probabilidade levantada a partir de \mathcal{W} foi melhor que aquela proporcionada pela distribuição gaussiana.

Entretanto, observa-se também que mais configurações foram amostradas em \mathcal{W}_{free} quando foi utilizada a função de probabilidade levantada a partir de \mathcal{W} (3003 configurações) do que quando foi utilizada a distribuição uniforme (1705 configurações). Isto acontece porque naquela amostragem a probabilidade de amostrar em \mathcal{C}_{free} é

maior. Sendo assim, como a topologia de \mathcal{C} é igual à topologia de \mathcal{W} , então também há uma maximização da probabilidade de amostrar em \mathcal{W}_{free} .

A Tabela 1 mostra os tempos de execução médio do BI-ARW padrão e várias configurações do BI-ARW melhorado. A medição de tempo foi feita sem levar em consideração o tempo de suavização, pois esta se dá numa etapa posterior no MRPAA. Os ambientes utilizados foram os da Figura 2 e Figura 3, um mapa real gerado em (Borges, 2002) e um grande labirinto cujo tamanho é quatro vezes maior que o da Figura 2, porém com um número consideravelmente maior de caminhos sem saída. Para cada ambiente foram feitos oito questionamentos de acordo com a metodologia proposta em Adorno and Borges (2006).

Pelos resultados da Tabela 1 é possível notar que todas as extensões propostas mostraram vantagens em relação ao BI-ARW padrão no que se refere ao tempo de execução. Além disso, a amostragem uniforme com apenas um candidato se mostrou eficiente, mas para ambientes grandes com algumas passagens bem estreitas foi necessária a seleção de candidatos. Isto porque esta seleção acaba empurrando o passeio para dentro dessas passagens, dado que o passeio consegue cobrir rapidamente os grandes ambientes e as passagens estreitas passam a ser regiões menos exploradas e portanto funcionam como um atrator local.

Além disso, apesar da função de probabilidade gerada a partir de \mathcal{W} permitir um maior aproveitamento de amostras, este método é computacionalmente mais custoso que a geração de amostras por distribuição uniforme. Sendo assim, mesmo que sua explorabilidade seja um pouco maior, esta acontece mais lentamente.

Foi constatado também que a etapa de fusão apresentada na Seção 3.2 não mostrou a eficiência esperada. Apesar de ter sido observado que um passeio aleatório em geral encontra primeiro o rastro do outro ao invés de encontrar sua última configuração, este processo se mostrou computacionalmente mais lento que uma simples busca pela última configuração.

A Figura 3(a) mostra um caminho suavizado pelo método apresentado em (Carpin and Pilonetto, 2005b), sendo que é possível perceber que as configurações quase encostam nos obstáculos. Já a Figura 3(b) mostra este mesmo caminho, mas com as etapas posteriores de suavização apresentadas na Seção 3.4. Nota-se que as configurações situadas no meio do corredor se situam no eixo médio, enquanto que as configurações das grandes áreas abertas não ficam no eixo médio, mas a uma distância segura dos obstáculos.

6 Conclusões e Trabalhos Futuros

Neste artigo foram propostas extensões para o algoritmo ARW de forma a melhorar as caracterís-

Tabela 1: Tempos de execução dos BI-ARW com diversos mecanismos de exploração (*ms*).

	*duas candidatas	**duas candidatas e fusão com 20 configurações no passeio oposto	Informações de W^{**}		
	BI-ARW padrão	Uniforme	Uniforme*	Uniforme**	
Figura 2	861,8	322,4	462,5	870,9	879,5
Figura 3	21549,5	3127,5	611,485	1950,4	1828,0
Mapa real	178,4	166,0	161,2	186,6	153,1
Grande labirinto	18821,1	3997,7	4896,6	5267,1	7783,2

ticas do algoritmo híbrido MRPA. Foi feita uma avaliação de três formas de amostragem do espaço de configuração, seleção de candidatos, fusão dos passeios aleatórios e suavização do caminho resultante. Além disso, foram propostas políticas de melhor utilização do mapa de rotas. Resultados mostraram que a amostragem uniforme adaptativa com seleção de candidatos se mostrou a mais eficiente, pois apesar de simples ainda consegue explorar de forma eficiente o espaço de configuração. Já o método de fusão dos passeios aleatórios não obteve a eficiência esperada. Por último, o método de suavização apresentado se mostrou eficiente ao afastar dos obstáculos o caminho resultante e ainda ao conseguir buscar atalhos no mesmo. Sugere-se como propostas de trabalhos futuros uma melhor avaliação na escolha de candidatos para amostragem do espaço de configuração, além da avaliação do algoritmo para robôs com grandes números de graus de liberdade.

Referências

- Adorno, B. V. and Borges, G. A. (2006). Um método de planejamento de trajetória para robôs móveis através de passeios aleatórios adaptativos e mapa de rotas, *XVI Congresso Brasileiro de Automática* pp. 1–6.
- Barraquand, J., Kavraki, L., Latombe, J. C. and Li, T. (1997). A random sampling scheme for path planning, *The International Journal of Robotics Research* **16**: 759–774.
- Bekris, K. E., Chen, B. Y., Ladd, A. M., Plaku, E. and Kavraki, L. E. (2003). Multiple query motion planning using single query primitives, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, Nevada, USA, pp. 656–661.
- Berg, J. P. and Overmars, M. H. (2004). Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners, *IEEE International Conference on Robotics and Automation* pp. 453–460.
- Borges, G. A. (2002). *Cartographie de l’environnement et localisation robuste pour la navigation de robots mobiles*, PhD thesis, Université Montpellier II, LIRMM, 161 rue ADA, 34392, Montpellier, Cedex 5, France.
- Carpin, S. and Pillonetto, G. (2005a). Merging the adaptive random walks planner with the randomized potential field planner, *Fifth International Workshop on Robot Motion and Control* pp. 151–156.
- Carpin, S. and Pillonetto, G. (2005b). Motion planning using adaptive random walks, *IEEE Transactions on Robotics* **21**(1).
- Choset, H., Lynch, K. M., Hutchinson, S., G. Kantor, W. B., Kavraki, L. E. and Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press.
- Geraerts, R. (2006). *Sampling-based Motion Planning: Analysis and Path Quality*, PhD thesis, Utrecht University.
- Geraerts, R. and Overmars, M. H. (2004). Clearance based path optimization for motion planning, *Proceedings of the 2004 IEEE Proceedings of the 2004 IEEE International Conference on Robotics & Automation* pp. 2386–2392.
- Geraerts, R. and Overmars, M. H. (2006). Creating high-quality roadmaps for motion planning in virtual environments, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’06)* pp. 4355–4361.
- Hsu, D. (2000). *Randomized Single-Query Motion Planning in Expansive Spaces*, PhD thesis, Department of Computer Science, Stanford University.
- J. J. Kuffner, J. and LaValle, S. M. (2000). Rrt-connect: An efficient approach to single-query path planning, *IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 995–1001.
- Kavraki, L. E., Kolountzakis, M. N., Latombe, J. C. and Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Transactions on Robotics and Automation* **12**(4).
- Kurniawati, H. and Hsu, D. (2004). Workspace importance sampling for probabilistic roadmap planning, *In Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.
- Wilmarth, S. A., Amato, N. M. and Stiller, P. E. (1999). Maprm: A probabilistic roadmap planner with sampling on the medial axis of the free space, *IEEE International Conference on Robotics & Automation*, pp. 1024–1031.