



Lydia E. Kavraki et al. : Probabilistic Roadmaps for Path Planning in High-dimensional Configuration Spaces

IEEE Trans. on Robotics and Automation, Vol.12, No.4, pp.566-580 (1996)

経路計画問題とは

ロボットが何らかの作業をするために行動する際、環境との衝突を回避する必要がある。たとえば、移動ロボットの場合、目的地までぶつかることなく移動する必要がある。ロボットがものをつかもうとする場合、腕をまわりとぶつけないように、ものまで手を伸ばす必要がある。

衝突のない行動を生成する手法：経路計画 (Motion Planning) 問題は古くから研究されている。一見ロボットにしか使えない適用範囲の狭い話に見えるが、コンピュータグラフィクスや分子生物学へも応用されている。

自由度 (たとえば、動く関節の数) の小さいロボットの経路計画問題は、1990 年の頃までに有用な方法が提案されてきたが、自由度の大きいロボットに対し、そのままでは適用できないといった問題があった。腕のみのロボットアームでも関節が5つ以上あることは普通であり、30 以上の自由度を持つヒューマノイドロボットも出現している。

本稿では、多自由度 (たとえば5つ以上の関節を持つ) ロボットの経路計画法の元祖である Probabilistic Roadmap Method (PRM) を取り上げる。本稿執筆時の IEEE Xplore Digital Library での被引用数は 2,000 回近く、Google Scholar ではおよそ 4,500 回である。

前提知識

経路計画問題では、配位空間 (Configuration-Space ; C-Space と記述される) がよく使われる。配位空間とは、ロボットの状態を一意に表す空間であり、たとえば、2つの関節があるロボットでは、それぞれの関節角度を軸にとった空間、平面上を移動するロボットでは、位置 (x, y) と向き θ の3つの軸からなる空間がそれに当たる。

C-space 上のある1点で与えられるロボットの状態において、ロボットと環境が衝突していない、接している、めり込んでいるという3つの状況に分けることができる。環境に衝突していない状態の集合を free C-space と呼ぶ。

経路計画問題では、現在と目標となるロボットの状態が入力として与えられ、2つの状態を結ぶ衝突のない経路が出力される。C-Space を使って記述すると、経路計画問題は C-space 上の2点を結ぶ free C-space のみを通る曲線を求める問題であるといえる。提案手法では、環境は静的、つまり変化しないものと仮定する。これは free C-space の形が変わらないことを意味する。

経路計画問題を解決する方法として Roadmap 法が提案されている。Roadmap は頂点と辺の集合からなるグラフで表現されており、頂点は free C-space 上のある状態を、辺は2つの状態の間に衝突のない経路が存在することを表す。Roadmap 法を用いた経路計画問題は、まずスタート・ゴールの

状態と Roadmap 上の任意の頂点との衝突のない経路を求め、次にスタート・ゴールに接続された頂点間の経路をグラフ上で探索することで実現される。Roadmap の2つの頂点に対応する状態の間に衝突のない経路がある(ない)場合、Roadmap 上で経路がある(ない)ことが満たされており、スタート・ゴールとロードマップ上の任意の頂点との間に衝突のない経路のあるなしを正確に求めることができれば、すべての入力に対し、必ず衝突のない経路が得られることが保証される。

キーポイント

Free C-space の形を計算することは NP 困難な問題であることが知られている。提案手法は、以下の3つの事実を上手く活用することで、問題の困難さを回避している。

1. ある状態において環境に衝突しているかどうかは簡単に判断できる。
2. ある2つの状態を結ぶ経路が与えられたとき、その経路が環境への衝突を伴うかどうかは判断できる。
3. 距離の近い2つの状態の経路を見つける方が容易である。

また、グラフの辺は2つの状態に経路が見つかった場合にのみ結ぶこととする。元々の Roadmap 法と異なり、衝突のない経路は存在するが PRM では求められない場合があることを認める。

PRM ではランダムサンプリングと Local planner という2つのアイデアを使う。ランダムサンプリングを利用することで、環境と衝突がないロボットの状態をたくさん生成する。Local planner は生成されたロボットの状態間の経路を、あらかじめ定められた経路(たとえば C-space 上で直線に対応する経路)に対してのみ衝突の有無を判断し、Roadmap の辺を構成していく。経路が長いほど環境と

衝突する可能性が多くなるので、距離の近い2つの状態に対してのみ到達できるかどうかを試していくことによって、効率良く Roadmap を得る。

ただし、ランダムサンプリングでは細いトンネルを抜けるようなところで、十分なサンプルが得られないことが考えられる。そこで、選択的な探索によりそういった領域にもグラフを伸ばしていこうと試みたのがもう1つのポイントとなる。

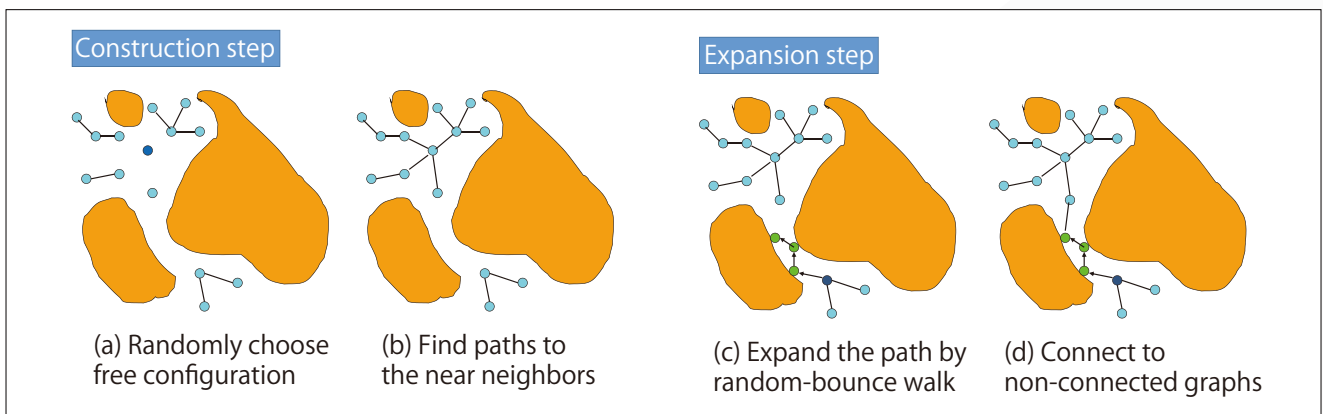
アルゴリズム

提案手法は、Learning Phase と Query Phase に分かれる。Learning Phase では Roadmap の生成を、Query Phase で実際の経路計画を行う。Learning Phase で生成される Roadmap は逐次増やしていくことが可能であり、計算時間に余裕があれば Learning Phase を再開することで、Roadmap を拡張することができる。

Learning Phase の概要を図-1 に示す。Learning Phase は、Construction step と Expansion step の2つからなる。前者は free C-space 全体をカバーすることを目的とし、後者は細いトンネルなどグラフを伸ばすのが難しいところに拡張することを目指す。具体的なアルゴリズムは以下の通り。

Learning Phase Construction step

1. 乱数を用いて状態 c を1つ生成する。
2. もし状態 c が free C-space 上であればグラフの頂点を加える(図-1 (a))。そうでなければ手順1へ戻る。
3. 状態 c との距離がある閾値以下のグラフ上の頂点の集合 N_c を求める。
4. N_c の要素を距離の小さい順に並べる。
5. N_c の先頭から要素 n を取り出し、 c と n の間に既定の経路を生成し、衝突があるか判断する。



◆ 図-1 Learning Phase の概要

6. 衝突がないと判断されれば、グラフの辺に $e = (c, n)$ を追加する (図-1 (b)).
7. 要素がなくなるまで手順 5, 6 を繰り返し、なくなれば手順 1 に戻る.

Expansion step

1. グラフの頂点においてグラフの伸ばしにくさを計算する.
2. 伸ばしにくさに従い、グラフの頂点をランダムに選択する.
3. ランダムに方向を選び、環境にぶつかるまで直進するのを繰り返す Random-bounce walk を行う (図-1 (c)).
4. 非連結なグラフの頂点との間に経路が存在するか計算する. 存在すれば、辺を追加する (図-1 (d)).

Query Phase

1. Construction step の手順 3 ~ 6 の方法にならない、スタート・ゴールの状態と Roadmap の頂点との接続を試みる.
2. 手順 1 で無理な場合は、スタート・ゴールの状態に Random-bounce walk を適用し、異なる状態を生成し、手順 1 に戻る.

実装上の工夫

Construction step の手順 3 の 2 つの状態間の距離の計算には工夫を要する. C-space 上での直線距離を単純に距離にすればよいと考えるかもしれない. たとえば、ヒューマノイドロボットのようなものを想定した場合、手先に近いところの関節が動くのと、腰あたりの関節が動くのでは、ロボット全体の移動量が変わってくる. そこで提案手法では、適度な時間で計算可能な尺度として、ロボット上の任意の点のうち最も大きく動いた個所の変位を 2 つの状態間の距離として定義している.

Expansion step の手順 1 の伸ばしにくさを正確に計算するためには、free C-space の形を知る必要があるため、何らかの推定方法を考える必要がある. 提案手法では、各頂点において Local planner が経路の発見に失敗した確率をもとに計算する方法を提案している. 伸ばしにくい個所では失敗する確率が高いことが期待される.

経路計画問題の発展

経路計画に興味を持たれた方がいれば、The



Open Motion Planning Library ^{☆1} を試してみることをお勧めする。Available planner のところを見れば、PRM ともう1つのランダムサンプリングに基づく経路計画法の雄 Rapidly-exploring Random Trees (RRT)、PRM や RRT から派生したさまざまなアルゴリズムが実装されていることが分かる。この Available planner の数を見れば、この論文の与えた影響の大きさを想像することができると思う。論文でも指摘されているように、衝突のない経路が存在するにもかかわらず、アルゴリズムが経路の発見に失敗する確率は、計算に無限に時間をかければ0に近づく (random completeness) という魅力的な性質を持っており、今後も PRM や RRT をベースとした経路計画手法は提案されていくと考えている。

筆者は組立て作業における高速な動作生成とい

^{☆1} <http://ompl.kavrakilab.org/>

う方法を提案しているのであるが、その査読の際にも、「RPM を使えばできるのだから、そのような特殊な場面でしか使えないものを提案する意味はあるのか?」というコメントをいただき RPM の偉大さを知ったことを、いまさらながらに思い出した (笑)。残念ながらほかの物体との接触を維持しながら動かすことの多い組立て作業には、RPM は単純には適用できない。ランダムサンプルの仕方と既定の経路 (Construction step 5) の生成に工夫を要する。動作に拘束の多い作業に対する適用、および経路計画の計算の高速化は今後も重要なテーマであると考えている。

(2018年8月28日受付)

.....
高松 淳 (正会員) j-taka@is.naist.jp

2004年東京大学大学院・情報科学研究科博士課程修了。博士 (情報理工学)。2008年より奈良先端科学技術大学院大学に所属。マニピュレーション、動作計画、ロボットビジョンに関する研究に従事。

