

Algorithm 1: RRT*

```

Rad = r
G (V, E) //Graph containing edges and vertices
for itr in range (0...n)
    Xnew = RandomPosition ()
    If Obstacle (Xnew) == True, try again
    Xnearest = Nearest (G (V, E), Xnew)
    Xneighbors = findNeighbors (G (V, E), Xnew, Rad)
    //Select  $u_i$  to move from Xneighbors towards Xnew (Based on Kinematics)
    for x' in Xneighbors
        Xbest = Xneighbors(i) + [f (xi, ui) 1]TΔt
    //Update Xnew based on ui
    If Obstacle(Xbest) == True:
        Link = Chain(Xnew,Xbest)
        Parent(x') = Xnew
        G += {Xnew,x'}
        G += Link
Return G (V, E)

```