**Algorithm 2：Dynamic-RRT-Connect**

**Input:** Xinit, Xgoal**,** i←0**;** old graph node set Gold

**Output:** path **from** Xinit to Xgoal **in** the updated map

Gnew ← ClearCollisionNode(Gold)

Ginit ←ReachableNodesAndEdges(Xinit**,** Gnew)

Ggoal ←ReachableNodesAndEdges(Xgoal**,** Gnew)

**for** i < IterMax **do**

Xrand ←RandomNode**(**C\_free, delta)

Xnearest\_init ← Nearest(Xrand**,** Ginit**)**

Xnew\_init ←GetNewNode**(**Xnearest\_init**,** Xrand**)**

**if** CheckCollision**(**Xnearest\_init**,** Xnew**) then**

Ginit.append(Xnearest\_init**,** Xnew)

Xnearest\_goaltoinit ← Nearest(Xnew\_init**,** Ggoal**)**

Xprim\_goal ←GetNewNode**(**Xnearest\_goaltoinit **,** Xnew\_init**)**

Xlast\_node ←Xprim\_goal

**if** CheckCollision**(**Xprim\_goal**,** Xgoal**) then**

Ggoal.append(Xprim\_goal**,** Xnearest\_goaltoinit)

**while not** SameNode(Xlast\_node, Xnew\_init) **do**

        Xprim\_goal\_iter ←GetNewNode**(**Xlast\_node **,** Xnew\_init**)**

**if** CheckCollision**(**Xprim\_goal**,** Xprim\_goal\_iter**) then**

Ggoal.append(Xprim\_goal\_iter**,** Xlast\_node)

Xlast\_node ←Xprim\_goal\_iter

**if** SameNode(Xlast\_node, Xnew\_init) **then**

Ggoal.append(Xlast\_node**,** Xnew\_init)

Gold ← UnionGraph(Ggoal**,** Ginit)

path ← GetFinalPath(Xinit**,** Xgoal)

**return** path

Ginit ← ClearCollisionNode(Ginit)

Ggoal ← ClearCollisionNode(Ggoal)

**if** size(Ggoal) **>** size(Ginit) then

swap(Ggoal**,** Ginit)

CheckCollision: 检查线段是否与障碍相交

ClearCollisionNode： 清空图中因相交而无效的点

ReachableNodesAndEdges： 搜集联通的点和边

RandomNode： 随机产生点

Nearest ： 最近的点

GetNewNode： 产生新的点

SameNode： 返回是否相同的点

UnionGraph： 取两个图的并集

GetFinalPath： 获取最终路径