

---

**Algorithm 1** Jax Rendezvous (heuristic based multirobot rendezvous)

---

```
1: Let:  $map = \{cell_{x,y,ownerID} \in R^2\}$ 
2: Let:  $path = \{cells \in map_{visited} | length \leq maxPathLength\}$ 
3: Let:  $group = \{robot_i \in Robots |$ 
    $\{robot_i.path \cap self.path\} > 0 \ \&\& \ dist < maxDistFromNeighbor\}$ 
4: Let:  $groupLeader = \{min(robot_i.id) \in group\}$ 
5: Let:  $centroids = \{centroidofgroup, overallcentroid\}$ 
6: Let:  $weights = \{neighbor, groupCentroid, xy\}$ 

7: procedure OVERALL STEP FUNCTION
8: recvUpdates:
9:    $neighborInfo \leftarrow robot_i.update$ 
10:   $map_{x,y} \leftarrow robot_i.id$ 
11: checkForStopCriteria:
12:  if  $length(group) == length(Robots) \ \&\& \ self.cell == leader.cell$  then return
13: refineGroup:
14:  for  $neighbors \in group$  do
15:    if  $euclideanDistance(robot, neighbor_i) > maxDistFromNeighbor$  then
16:       $group.remove(neighbor_i)$ 
17: computeOptions:
18:   $computeCentroid()$ 
19:   $computeGroupCentroid()$ 
20:   $weightDirections()$ 
21:  if  $length(group) == length(Robots)$  then
22:     $weights = 0, 0, 1$ 
23:     $weightByXY(leader.cell)$ 
24:    if  $leader.cell \in self.path$  then
25:       $self.backTrack = true$ 
26:  else if  $length(group) > 0$  then
27:     $weights = 2, 1, 0$ 
28:     $weightByNeighbor()$ 
29:     $weightByGroup()$ 
30:  else
31:     $weights = 1, 0, 0$ 
32:     $weightByNeighbor()$ 
33: move:
34:   $checkFreedom()$ 
35:   $checkForPause()$ 
36:   $sortDirections()$ 
37:  for  $rankedDirections \in possibleDirections$  do
38:    if  $self.backTrack$  then break
39:    if  $cell_{t+1}.ownerID \neq self.id$  then
40:       $group.add(cell_{t+1}.ownerID)$ 
41:       $move()$ 
42:  if  $hasNotMoved$  then
43:     $backTrack()$ 
44:     $self.backTrack = false$ 
45: checkForSuccessCriteria:
46:  if  $length(group) == length(Robots) \ \&\& \ euclideanDistance(robot_i, self) == 0$  then
47:     $success = true$ 
48:     $exit()$ 
```

---