Algorithm 1 Jax Rendezvous (heurisitic based multirobot rendezvous)

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

procedure OVERALL STEP FUNCTION
recvUpdates:
    $neighborInfo \leftarrow robot_i.update$
    $map_{\mathrm{x,y}} \leftarrow robot_i.id$
checkForStopCriteria:
    if $length(group) == length(Robots)$ && $self.cell == leader.cell$ then return
refineGroup:
    for $neighbors \in group$ do
        if $euclideanDistance(robot, neighbor_i) > maxDistFromNeighbor$ then
            $group.remove(neighbor_i)$
computeOptions:
    $computeCentroid()$
    $computeGroupCentroid()$
    $weightDirections()$
    if $length(group) == length(Robots)$ then
        $weights = 0, 0, 1$
        $weightByXY(leader.cell)$
        if $leader.cell \in self.path$ then
            $self.backTrack = true$
    else if $length(group) > 0$ then
        $weights = 2, 1, 0$
        $weightByNeighbor()$
        $weightByGroup()$
    else
        $weights = 1, 0, 0$
        $weightByNeighbor()$
move:
    $checkFreedom()$
    $checkForPause()$
    $sortDirections()$
    for $rankedDirections \in possibleDirections$ do
        if $self.backTrack$ then break
        if $cell_{\mathrm{t+1}}.ownerID != self.id$ then
            $group.add(cell_{\mathrm{t+1}}.ownerID)$
            $move()$
    if $hasNotMoved$ then
        $backTrack()$
        $self.backTrack = false$
checkForSuccessCriteria:
    if $length(group) == length(Robots)$&&$euclideanDistance(robot_i, self) == 0$ then
        $success = true$
        $exit()$
```