

Report for AI Quadcopter Planning

Introduction to Artificial Intelligence - 4CCS1IAI



by

Nashwan Nouri - 1432099

Sedar Olmez - 1415091

Dominik Kulon - 1410203

Dorian Buksz - 1439289



Introduction

The following planning domain will be used to manage a quadcopter with package holding capabilities to be used in search and rescue missions such as delivering first-aid toolkits. The quadcopter will be able to move in a 3D environment, while delivering the loaded package to the person that need it. For maximum precision and efficiency, a gripper is attached to hold the package while the camera takes a confirmation picture upon delivery.

For safely deploying the package, the quadcopter will make use of its gripper and the coordinate system to insure that there is no collision with the person and the package dropped from directly above the person, to ensure the person receives the package, if s/he is unable to move.

After delivery, the quadcopter will be able to re-load another package and continue its mission until no more are required. The loading procedure will require the quadcopter to move to a specific location (base) at a specific altitude, and will imply the deployment of its gripper for picking up the next package.

As far as fuel is concerned, the quadcopter will run on a rechargeable battery. It will be able to recharge by landing and staying for a predetermined duration at a docking station.

The quadcopter will be able to move either horizontally or vertically, but not both at the same time.

Domain description

Our domain will make use of the following types:

PERSON	a type used to identify a target person that awaits the delivery of a package
BASE	the type used to identify a location from where the quadcopter is able to pick-up additional packages
QUADCOPTER	self-explanatory, only one at a time will be used in our scenarios
PACKAGE	a type used to identify the packages
CHARGE-DOCK	the type used to identify a location where the quad-copter can land and recharge its battery
COORDINATES	self-explanatory
GRIPPER	a type used to identify the quadcopters attached gripper

For functions, we have:

BATTERY	The current energy level of the quadcopter's battery
BATTERY-CAPACITY	The maximum capacity of the quadcopter's battery
CHARGE-DOCK-REFILL-RATE	The refill rate of the docking station

The following predicates will be used:

PATH-HORIZONTAL	Predicate used for representing the horizontal path of the quadcopter
PATH-VERTICAL-UP	Predicate used for representing the vertical ascending path of the quadcopter
PATH-VERTICAL-DOWN	Predicate used for representing the vertical descending path of the quadcopter
PACKAGE-AT	Used to evaluate if package is present at the specified coordinates
QUAD-CARRY	Used to evaluate if package is loaded in the quadcopter
QUAD-GRIPPER-FREE	Used to evaluate if the gripper is free
QUAD-AT	Used to evaluate if the quadcopter is present at the specified coordinates
CHARGE-DOCK-OCCUPIED	Used to evaluate if the specified docking station is occupied
CHARGE-DOCK-AT	Used to evaluate if the docking station is at the specified coordinates
COORDINATE-EMPTY	Used to limit the quadcopters surroundings to prevent collision with persons
QUAD-HAS-CAMERA	Used to evaluated if a camera is present on the quadcopter
PERSON-AT	Used to evaluate if the person is at the specified coordinate
PICTURE-TAKEN	Used to specify that a picture has been taken after the package-drop
BASE-AT	Used to evaluate if the base is at the specified coordinates
QUAD-AT-BASE	Used to evaluate if the quadcopter is at the specified base

As far as actions go, we will use the following durative actions:

TAKE-OFF	This action is used when the mission begins. The quadcopter is located at the base and with a first-aid kit loaded and battery fully charged.
MOVE-HORIZONTAL	For moving horizontally across the mission map, the quad-copter will be using this action
ASCEND	This action will be used for elevating the quadcopter to a higher altitude
DESCEND	Same as ASCEND, but this time for moving the quadcopter to a lower altitude
DELIVER-PACKAGE	This action will be used for delivering a first-aid kit, provided that the quadcopter has one attached and that it is near a target person
LOADPACKAGE	Action used for loading a new package, provided that a gripper that is attached to the quadcopter is available

REFILL	For recharging the quadcopters battery, the quadcopter will be using this action, provided that it is located at a docking station
LAND	This action will be used once the mission has been completed and no more persons require a package. The quadcopter will land at the base.

Problem analysis

Our initial idea was to create a solution to a problem that consisted of a single quadcopter, package and person; however, as our team progressed beyond all expectations, we managed to solve problems that consisted of up to three packages used to save 3 people at various locations (many more packages and people could be added to problem files, however, this is not advised as there will be many states visited).

Below is a table comparing the cost and time taken when the initial battery level is different for the quadcopter during the 1 package 1 person scenario.

No. Battery charge	States Required	No. refills	Refill at action No.	Cost	Time	Used Best First Search?
10	34	0	n/a	10.009	0.81	No
9	748	1	9	15.014	1.71	Yes
8	632	1	9	16.014	1.50	Yes
7	565	1	8	14.014	1.45	Yes
6	684	2	7,8	21.021	1.67	Yes
5	n/a	n/a	n/a	n/a	n/a	Unsolvable

There is a huge contrast from when the quadcopter can solve the problem without a refill and when it requires at least one refill. The time taken is more than doubled and the states visited is more than 20 times bigger when comparing the initial battery size of 10 and 9. This is partially due to the fact that a refill increases the battery charge by 10, which causes the quadcopter to have more states it can visit once refilled. Between battery charge 9 and 8 the quadcopter visits less states in the latter. This is due to it having less battery before and after the refill, so it has less coordinates it can explore, so less states. This also correlates to the time taken as it has decreased by 0.21 seconds. An initial Battery charge of 7 is the most optimum when considering at least 1 refill. It visits the least amount of states and takes the least amount of time. When the initial battery charge is 6, it needs to be refilled twice to reach the goal state. There is a jump in cost and time, as the quadcopter battery after refilling twice will be very large which causes it to visit many unnecessary states. When the initial battery charge is 5, the quadcopter cannot reach the docking station so the problem is unsolvable due to the limited battery.

```

; States evaluated: 565
; Cost: 14.014
; Time 1.40
0.000: (take-off quad base c0_0_0) [1.000]
0.001: (loadpackage c0_0_0 package1 quad gripper) [2.000]
2.002: (move-horizontal c0_0_0 c1_0_0 quad) [1.000]
3.003: (ascend c1_0_0 c1_0_1 quad) [1.000]
4.004: (move-horizontal c1_0_1 c1_1_1 quad) [1.000]
5.005: (move-horizontal c1_1_1 c2_1_1 quad) [1.000]
6.006: (descend c2_1_1 c2_1_0 quad) [1.000]
7.007: (refill c2_1_0 quad charge_dock) [1.000]
8.008: (ascend c2_1_0 c2_1_1 quad) [1.000]
9.009: (move-horizontal c2_1_1 c1_1_1 quad) [1.000]
10.010: (deliver-package c1_1_1 c1_1_0 package1 quad gripper person1) [2.000]
10.011: (move-horizontal c1_1_1 c0_1_1 quad) [1.000]
11.012: (move-horizontal c0_1_1 c0_0_1 quad) [1.000]
12.013: (descend c0_0_1 c0_0_0 quad) [1.000]
13.014: (land quad base c0_0_0 gripper) [1.000]

```

1 Package 1 Person

Fig 1 : 565 the minimal capacity our problem file can enforce on Optic is a single package and person, the time it takes for Optic to solve this particular environment is **1.40** minutes, the theoretical reachable cost from the initial state to the goal state is **14.014**, now lets up the game and increase the amount of packages and people to save.

```

(G);;; Solution Found
; States evaluated: 2910
; Cost: 46.042
; Time 7.62
0.000: (take-off quad base c0_0_0) [1.000]
0.001: (loadpackage c0_0_0 package1 quad gripper) [2.000]
2.002: (move-horizontal c0_0_0 c1_0_0 quad) [1.000]
3.003: (ascend c1_0_0 c1_0_1 quad) [1.000]
4.004: (move-horizontal c1_0_1 c2_0_1 quad) [1.000]
5.005: (move-horizontal c2_0_1 c2_1_1 quad) [1.000]
6.006: (descend c2_1_1 c2_1_0 quad) [1.000]
7.007: (refill c2_1_0 quad charge_dock) [1.000]
8.008: (ascend c2_1_0 c2_1_1 quad) [1.000]
9.009: (move-horizontal c2_1_1 c2_0_1 quad) [1.000]
10.010: (deliver-package c2_0_1 c2_0_0 package1 quad gripper person1) [2.000]
10.011: (move-horizontal c2_0_1 c2_1_1 quad) [1.000]
12.011: (descend c2_1_1 c2_1_0 quad) [1.000]
13.012: (refill c2_1_0 quad charge_dock) [1.000]
14.013: (refill c2_1_0 quad charge_dock) [1.000]
14.014: (ascend c2_1_0 c2_1_1 quad) [1.000]
15.015: (move-horizontal c2_1_1 c1_1_1 quad) [1.000]
16.016: (move-horizontal c1_1_1 c1_0_1 quad) [1.000]
17.017: (descend c1_0_1 c1_0_0 quad) [1.000]
18.018: (move-horizontal c1_0_0 c0_0_0 quad) [1.000]
19.019: (loadpackage c0_0_0 package2 quad gripper) [2.000]
21.020: (move-horizontal c0_0_0 c1_0_0 quad) [1.000]
22.021: (ascend c1_0_0 c1_0_1 quad) [1.000]
23.022: (move-horizontal c1_0_1 c1_1_1 quad) [1.000]
24.023: (deliver-package c1_1_1 c1_1_0 package2 quad gripper person2) [2.000]
26.024: (move-horizontal c1_1_1 c0_1_1 quad) [1.000]
27.025: (move-horizontal c0_1_1 c0_0_1 quad) [1.000]
28.026: (descend c0_0_1 c0_0_0 quad) [1.000]
29.027: (loadpackage c0_0_0 package3 quad gripper) [2.000]
31.028: (move-horizontal c0_0_0 c1_0_0 quad) [1.000]
32.029: (ascend c1_0_0 c1_0_1 quad) [1.000]
33.030: (move-horizontal c1_0_1 c1_1_1 quad) [1.000]
34.031: (move-horizontal c1_1_1 c1_2_1 quad) [1.000]
35.032: (move-horizontal c1_2_1 c2_2_1 quad) [1.000]
36.033: (deliver-package c2_2_1 c2_2_0 package3 quad gripper person3) [2.000]
38.034: (move-horizontal c2_2_1 c2_1_1 quad) [1.000]
39.035: (descend c2_1_1 c2_1_0 quad) [1.000]
40.036: (refill c2_1_0 quad charge_dock) [1.000]
40.037: (ascend c2_1_0 c2_1_1 quad) [1.000]
41.038: (move-horizontal c2_1_1 c1_1_1 quad) [1.000]
42.039: (move-horizontal c1_1_1 c0_1_1 quad) [1.000]
43.040: (move-horizontal c0_1_1 c0_0_1 quad) [1.000]
44.041: (descend c0_0_1 c0_0_0 quad) [1.000]
45.042: (land quad base c0_0_0 gripper) [1.000]

```

3 Packages 3 People

Fig 2: above we test a **three people** rescue mission and manage to save all three people in roughly **7.62** minutes using only a single quad copter, the theoretical reachable cost has increased to **46.042** and the amount of states the quadcopter visited was **2910**. The same quadcopter and charge-dock functions where used in addition to the same map size.

Appendix

Domain file

```
(define (domain quadcopter)
  (:requirements :strips :typing :fluents :durative-actions :equality)
  (:types person base quadcopter package charge_dock coordinates gripper)
  (:predicates
    (path-horizontal ?from ?to - coordinates)
    (path-vertical-up ?from ?to - coordinates)
    (path-vertical-down ?from ?to - coordinates)
    (package-at ?package - package ?from - coordinates)
    (quad-carry ?quadcopter - quadcopter ?package - package ?gripper - gripper)
    (quad-gripper-free ?quadcopter - quadcopter ?gripper - gripper)
    (quad-at ?quadcopter - quadcopter ?from - coordinates)
    (charge-dock-occupied ?charge_dock - charge_dock ?from - coordinates)
    (charge-dock-at ?charge_dock - charge_dock ?at - coordinates)
    (coordinate-empty ?this - coordinates)
    (quad-has-camera ?quadcopter - quadcopter)
    (person-at ?person - person ?at - coordinates)
    (picture-taken ?person - person ?package - package)
    (base-at ?base - base ?at - coordinates)
    (quad-at-base ?quadcopter - quadcopter ?base - base)
  )

  (:functions
    (battery ?quadcopter - quadcopter)
    (battery-capacity ?quadcopter - quadcopter)
    (charge-dock-refill-rate ?charge_dock - charge_dock)
  )

  (:durative-action take-off
    :parameters(?quadcopter - quadcopter ?base - base ?at - coordinates)
    :duration(= ?duration 1)
    :condition(and
      (at start (base-at ?base ?at))
      (at start (quad-at-base ?quadcopter ?base))
      (over all (> (battery ?quadcopter) 0))
    )
    :effect(and
      (at start (not (quad-at-base ?quadcopter ?base)))
      (at start (quad-at ?quadcopter ?at))
      (at end (decrease(battery ?quadcopter) 1))
    )
  )
)
```

```

(:durative-action refill
  :parameters(?at - coordinates ?quadcopter - quadcopter ?charge_dock - charge_dock)
  :duration(= ?duration 1)
  :condition(and
    (at start (charge-dock-at ?charge_dock ?at))
    (at start (quad-at ?quadcopter ?at))
    (over all (<=(battery ?quadcopter) (battery-capacity ?quadcopter)))
  )
  :effect(and
    (at start (charge-dock-occupied ?charge_dock ?at))
    (at end (increase (battery ?quadcopter)(charge-dock-refill-rate ?charge_dock)))
    (at end (not (charge-dock-occupied ?charge_dock ?at)))
  )
)

```

```

(:durative-action land
  :parameters(?quadcopter - quadcopter ?base - base ?at - coordinates ?gripper -gripper)
  :duration(= ?duration 1)
  :condition(and
    (at start (quad-at ?quadcopter ?at))
    (at start (base-at ?base ?at))
    (at start (quad-gripper-free ?quadcopter ?gripper))
    (over all (> (battery ?quadcopter) 0))
  )
  :effect(and
    (at end (quad-at-base ?quadcopter ?base))
    (at start (not (quad-at ?quadcopter ?at)))
    (at end (decrease(battery ?quadcopter) 1))
  )
)

```

```

(:durative-action loadpackage
  :parameters(?from - coordinates ?package - package ?quadcopter - quadcopter
    ?gripper - gripper )
  :duration(= ?duration 2)
  :condition(and
    (at start (quad-at ?quadcopter ?from))
    (at start (package-at ?package ?from))
    (at start (quad-gripper-free ?quadcopter ?gripper))
    (over all (> (battery ?quadcopter) 0))
  )
  :effect(and
    (at start (not(package-at ?package ?from)))
    (at start (not(quad-gripper-free ?quadcopter ?gripper)))
    (at end (decrease(battery ?quadcopter) 1))
    (at start (quad-carry ?quadcopter ?package ?gripper))
  )
)

```

```

(:durative-action deliver-package
  :parameters(?from ?at - coordinates ?package - package ?quadcopter - quadcopter
              ?gripper - gripper ?person - person)
  :duration(= ?duration 2)
  :condition(and
    (at start(quad-at ?quadcopter ?from))
    (at start(quad-carry ?quadcopter ?package ?gripper))
    (at start (path-vertical-down ?from ?at))
    (at start (person-at ?person ?at))
    (over all (quad-has-camera ?quadcopter))
    (over all (> (battery ?quadcopter) 0))

  )
  :effect(and
    (at start (package-at ?package ?at))
    (at start (quad-gripper-free ?quadcopter ?gripper))
    (at start (not(quad-carry ?quadcopter ?package ?gripper)))
    (at end (picture-taken ?person ?package))
    (at end (decrease(battery ?quadcopter) 1))

  )
)

```

```

(:durative-action move-horizontal
  :parameters(?from ?to - coordinates ?quadcopter - quadcopter)
  :duration(= ?duration 1)
  :condition(and
    (at start (path-horizontal ?from ?to))
    (at start (quad-at ?quadcopter ?from))
    (at start (coordinate-empty ?to))
    (over all (> (battery ?quadcopter) 0))

  )
  :effect(and
    (at start (not (quad-at ?quadcopter ?from)))
    (at end (quad-at ?quadcopter ?to))
    (at end (not (coordinate-empty ?to)))
    (at end (coordinate-empty ?from))
    (at end (decrease(battery ?quadcopter) 1))

  )
)

```

```

(:durative-action ascend
  :parameters(?from ?to - coordinates ?quadcopter - quadcopter)
  :duration(= ?duration 1)
  :condition(and
    (at start (path-vertical-up ?from ?to))
    (at start (quad-at ?quadcopter ?from))
    (at start (coordinate-empty ?to))
    (over all (> (battery ?quadcopter) 0))
  )
)

```



```

    )
    :effect(and
      (at start (not(quad-at ?quadcopter ?from)))
      (at end (quad-at ?quadcopter ?to))
      (at end (not (coordinate-empty ?to)))
      (at end (coordinate-empty ?from))
      (at end (decrease(battery ?quadcopter)1))
    )
  )

  (:durative-action descend
    :parameters(?from ?to - coordinates ?quadcopter - quadcopter)
    :duration(= ?duration 1)
    :condition(and
      (at start (path-vertical-down ?from ?to))
      (at start (quad-at ?quadcopter ?from))
      (at start (coordinate-empty ?to))
      (over all (> (battery ?quadcopter) 0))
    )
    :effect(and
      (at start(not(quad-at ?quadcopter ?from)))
      (at end(quad-at ?quadcopter ?to))
      (at end (not (coordinate-empty ?to)))
      (at end (coordinate-empty ?from))
      (at end (decrease(battery ?quadcopter) 1))
    )
  )
)

```

Problem file

```
(define (problem Problem3)
  (:domain quadcopter)
  (:objects
    Quad - quadcopter
    Package1 Package2 Package3 - package
    Base - base
    Person1 Person2 Person3 - person
    gripper - gripper
    charge_dock - charge_dock
    Camera - camera
    c0_0_0 c0_0_1 c0_0_2 c0_1_0 c0_1_1 c0_1_2 c0_2_0 c0_2_1 c0_2_2 c1_0_0 c1_0_1
    c1_0_2 c1_1_0 c1_1_1 c1_1_2 c1_2_0 c1_2_1 c1_2_2 c2_0_0 c2_0_1 c2_0_2 c2_1_0 c2_1_1 c2_1_2
    c2_2_0 c2_2_1 c2_2_2 - coordinates
  )
  (:init
    (base-at Base c0_0_0)
    (quad-at-base Quad Base)
    (camera-at Quad Camera)
    (quad-gripper-free Quad gripper)
    (charge-dock-at charge_dock c2_1_0)

    (package-at Package1 c0_0_0)
    (package-at Package2 c0_0_0)
    (package-at Package3 c0_0_0)
    (person-at Person1 c2_0_0)
    (person-at Person2 c1_1_0)
    (person-at Person3 c2_2_0)

    (= (battery Quad) 7)
    (= (battery-capacity Quad) 20)
    (= (charge-dock-battery charge_dock) 20)
    (= (charge-dock-refill-rate charge_dock) 10)

    (path-horizontal c0_0_0 c1_0_0) (path-horizontal c0_0_0 c0_1_0) (path-horizontal c0_0_1 c0_1_1)
    (path-horizontal c0_0_1 c1_0_1) (path-horizontal c0_0_2 c1_0_2) (path-horizontal c0_0_2 c0_1_2)
    (path-horizontal c0_1_0 c0_0_0) (path-horizontal c0_1_0 c0_2_0) (path-horizontal c0_1_0 c1_1_0)
    (path-horizontal c0_1_1 c0_2_1) (path-horizontal c0_1_1 c0_0_1) (path-horizontal c0_1_1 c1_1_1)
    (path-horizontal c0_1_2 c0_0_2) (path-horizontal c0_1_2 c0_2_2) (path-horizontal c0_1_2 c1_1_2)
    (path-horizontal c0_2_0 c1_2_0) (path-horizontal c0_2_0 c0_1_0) (path-horizontal c0_2_1 c0_1_1)
    (path-horizontal c0_2_1 c1_2_1) (path-horizontal c0_2_2 c1_2_2) (path-horizontal c0_2_2 c0_1_2)
    (path-horizontal c1_0_0 c1_1_0) (path-horizontal c1_0_0 c0_0_0) (path-horizontal c1_0_0 c2_0_0)
    (path-horizontal c1_0_1 c1_1_1) (path-horizontal c1_0_1 c0_0_1) (path-horizontal c1_0_1 c2_0_1)
    (path-horizontal c1_0_2 c1_1_2) (path-horizontal c1_0_2 c0_0_2) (path-horizontal c1_0_2 c2_0_2)
    (path-horizontal c1_1_0 c1_0_0) (path-horizontal c1_1_0 c1_2_0) (path-horizontal c1_1_0 c0_1_0)
    (path-horizontal c1_1_0 c2_1_0) (path-horizontal c1_1_1 c1_2_1) (path-horizontal c1_1_1 c1_0_1)
    (path-horizontal c1_1_1 c0_1_1) (path-horizontal c1_1_1 c2_1_1) (path-horizontal c1_1_2 c1_2_2)
```

```

(path-horizontal c1_1_2 c1_0_2) (path-horizontal c1_1_2 c0_1_2) (path-horizontal c1_1_2 c2_1_2)
(path-horizontal c1_2_0 c1_1_0) (path-horizontal c1_2_0 c0_2_0) (path-horizontal c1_2_0 c2_2_0)
(path-horizontal c1_2_1 c1_1_1) (path-horizontal c1_2_1 c0_2_1) (path-horizontal c1_2_1 c2_2_1)
(path-horizontal c1_2_2 c1_1_2) (path-horizontal c1_2_2 c0_2_2) (path-horizontal c1_2_2 c2_2_2)
(path-horizontal c2_0_0 c1_0_0) (path-horizontal c2_0_0 c2_1_0) (path-horizontal c2_0_1 c2_1_1)
(path-horizontal c2_0_1 c1_0_1) (path-horizontal c2_0_2 c1_0_2) (path-horizontal c2_0_2 c2_1_2)
(path-horizontal c2_1_0 c2_2_0) (path-horizontal c2_1_0 c2_0_0) (path-horizontal c2_1_0 c1_1_0)
(path-horizontal c2_1_1 c2_2_1) (path-horizontal c2_1_1 c1_1_1) (path-horizontal c2_1_1 c2_0_1)
(path-horizontal c2_1_2 c2_2_2) (path-horizontal c2_1_2 c2_0_2) (path-horizontal c2_1_2 c1_1_2)
(path-horizontal c2_2_0 c1_2_0) (path-horizontal c2_2_0 c2_1_0) (path-horizontal c2_2_1 c2_1_1)
(path-horizontal c2_2_1 c1_2_1) (path-horizontal c2_2_2 c1_2_2) (path-horizontal c2_2_2 c2_1_2)

(path-vertical-up c0_0_0 c0_0_1) (path-vertical-up c0_0_1 c0_0_2) (path-vertical-up c0_1_0 c0_1_1)
(path-vertical-up c0_1_1 c0_1_2) (path-vertical-up c0_2_0 c0_2_1) (path-vertical-up c0_2_1 c0_2_2)
(path-vertical-up c1_0_0 c1_0_1) (path-vertical-up c1_0_1 c1_0_2) (path-vertical-up c1_1_0 c1_1_1)
(path-vertical-up c1_1_1 c1_1_2) (path-vertical-up c1_2_0 c1_2_1) (path-vertical-up c1_2_1 c1_2_2)
(path-vertical-down c1_2_2 c1_2_1) (path-vertical-up c2_0_0 c2_0_1) (path-vertical-up c2_0_1 c2_0_2)
(path-vertical-up c2_1_0 c2_1_1) (path-vertical-up c2_1_1 c2_1_2) (path-vertical-up c2_2_0 c2_2_1)
(path-vertical-up c2_2_1 c2_2_2)

(path-vertical-down c0_0_1 c0_0_0) (path-vertical-down c0_0_2 c0_0_1) (path-vertical-down c0_1_1
c0_1_0) (path-vertical-down c0_1_2 c0_1_1) (path-vertical-down c0_2_1 c0_2_0) (path-vertical-down
c0_2_2 c0_2_1) (path-vertical-down c1_0_1 c1_0_0) (path-vertical-down c1_0_2 c1_0_1) (path-vertical-
down c1_1_1 c1_1_0) (path-vertical-down c1_1_2 c1_1_1) (path-vertical-down c1_2_1 c1_2_0) (path-
vertical-down c2_0_1 c2_0_0) (path-vertical-down c2_0_2 c2_0_1) (path-vertical-down c2_1_1 c2_1_0)
(path-vertical-down c2_1_2 c2_1_1) (path-vertical-down c2_2_1 c2_2_0) (path-vertical-down c2_2_2
c2_2_1)

(coordinate-empty c0_0_0) (coordinate-empty c0_0_1) (coordinate-empty c0_0_2) (coordinate-empty
c0_1_1) (coordinate-empty c0_1_2) (coordinate-empty c0_2_0) (coordinate-empty c0_2_1) (coordinate-
empty c0_2_2) (coordinate-empty c1_0_0) (coordinate-empty c1_0_1) (coordinate-empty c1_0_2)
(coordinate-empty c1_1_1) (coordinate-empty c1_1_2) (coordinate-empty c1_2_0) (coordinate-empty
c1_2_1) (coordinate-empty c1_2_2) (coordinate-empty c2_0_1) (coordinate-empty c2_0_2) (coordinate-
empty c2_1_0) (coordinate-empty c2_1_1) (coordinate-empty c2_1_2) (coordinate-empty c2_2_0)
(coordinate-empty c2_2_1) (coordinate-empty c2_2_2)
)
(:goal (and
      (picture-taken Person1 Package1)
      (picture-taken Person2 Package2)
      (picture-taken Person3 Package3)
      (quad-at-base Quad Base))
)
(:metric minimize (total-time))
)

```