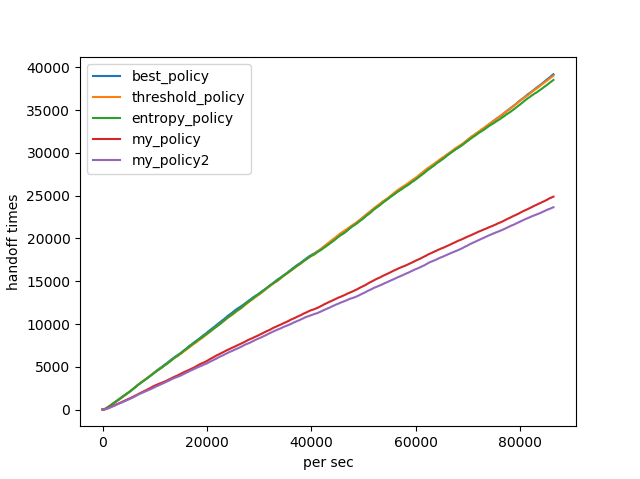
F74062010 劉松霖

1.圖表



2.source code

**def best\_policy(car):**

# hand off ...

old = car.hold

pold = car.power

pnew = pold

new = old

for a in range(0, 4): # find p max

bs = bs1 if a == 0 else bs2 if a == 1 else bs3 if a == 2 else bs4

if pnew < bs[car.y][car.x]:

pnew = bs[car.y][car.x]

new = a + 1

if pnew > pold or pold < pmin: # old not the max

car.change(new)

return True

else:

return False

pass

**def threshold\_policy(car):**

# hand off ...

old = car.hold

pold = car.power

pnew = pold

new = old

for a in range(0, 4): # find p max

bs = bs1 if a == 0 else bs2 if a == 1 else bs3 if a == 2 else bs4

if pnew < bs[car.y][car.x]:

pnew = bs[car.y][car.x]

new = a + 1

if (pnew > pold and pold < threshold) or pold < pmin: # old not the max and < threshold

car.change(new)

return True

else:

return False

pass

**def entropy\_policy(car):**

# hand off ...

old = car.hold

pold = car.power

pnew = pold

new = old

for a in range(0, 4): # find p max

bs = bs1 if a == 0 else bs2 if a == 1 else bs3 if a == 2 else bs4

if pnew < bs[car.y][car.x]:

pnew = bs[car.y][car.x]

new = a + 1

if pold + entro < pnew or pold < pmin: # old not the max and diff entropy

car.change(new)

return True

else:

return False

pass

pass

**def my\_policy(car):**

# hand off ...

old = car.hold

pold = car.power

pnew = pold

new = old

for a in range(0, 4): # find p max

bs = bs1 if a == 0 else bs2 if a == 1 else bs3 if a == 2 else bs4

if pnew < bs[car.y][car.x]:

pnew = bs[car.y][car.x]

new = a + 1

if (pnew > pold and car.duration > 75\*3) or pold < pmin: # old not the max and last 75\*3 sec

car.change(new)

return True

elif pold < pnew: # time for pold not the max

car.elapse()

return False

else:

return False

pass

pass

**def my\_policy2(car):**

# hand off ...

old = car.hold

pold = car.power

pnew = pold

new = old

for a in range(0, 4): # find p max

bs = bs1 if a == 0 else bs2 if a == 1 else bs3 if a == 2 else bs4

if pnew < bs[car.y][car.x]:

pnew = bs[car.y][car.x]

new = a + 1

if pold < pmin: # until pmin

car.change(new)

return True

else:

return False

pass

pass

3.introduction to your policy

best\_policy average power:-114.72121321773818

threshold\_policy average power:-114.6940126347598

entropy\_policy average power:-114.91339595752127

my\_policy average power:-117.05076337331167

my\_policy2 average power:-117.1380324903485

mypolicy:當車子累積超過75\*3秒原本負責BS功率不是最大再handoff，可減少部分剛好繞出一圈(走三段)又回到原本1/4區域的handoff，比best/threshold policy handoff較少次但average power比較低，彈性沒entropy policy高(由於參數問題在此例entropy policy有較多次handoff)。

mypolicy2:直到功率要小於pmin時才handoff，handoff必最少次，但average power最低。