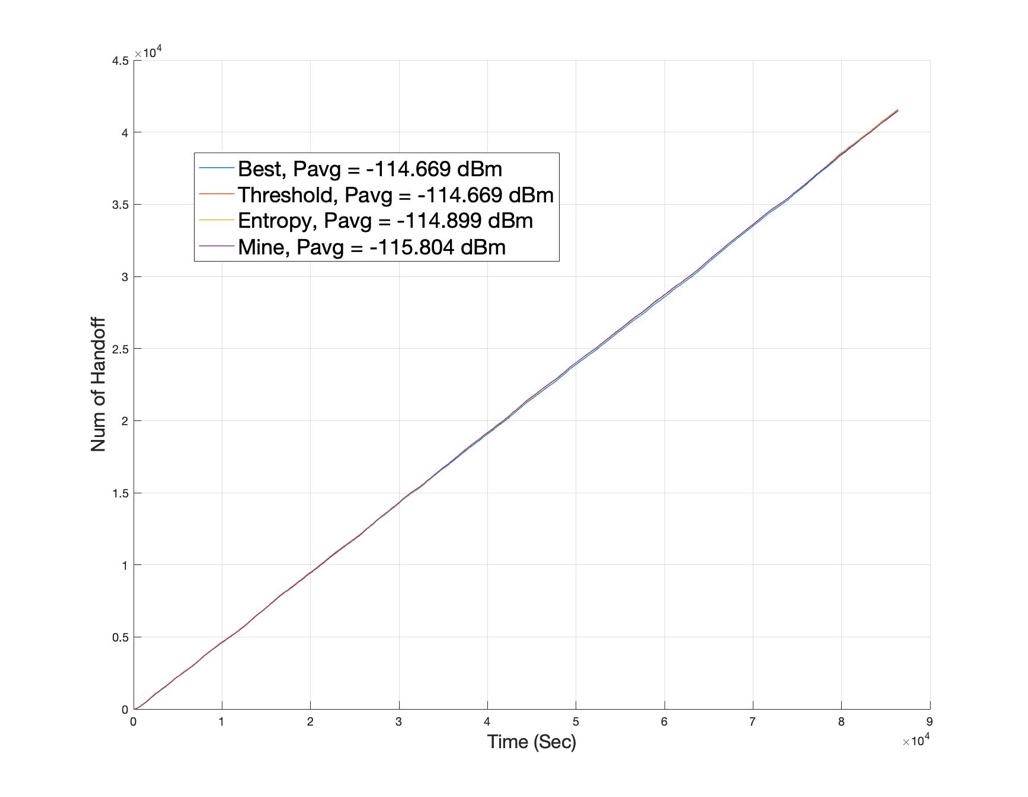
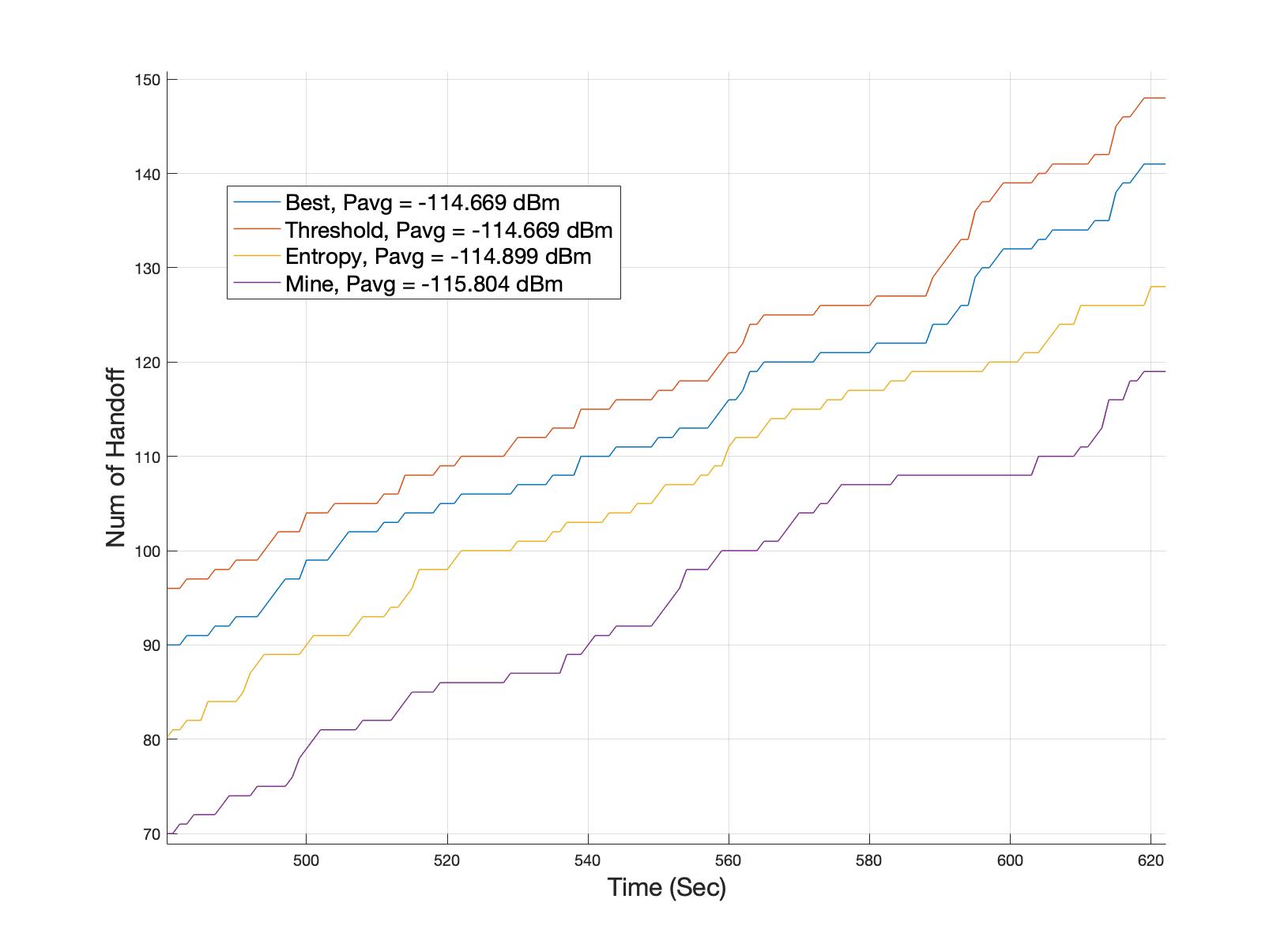
**無線通訊網路Project F04066028詹子毅**

1. **Charts**

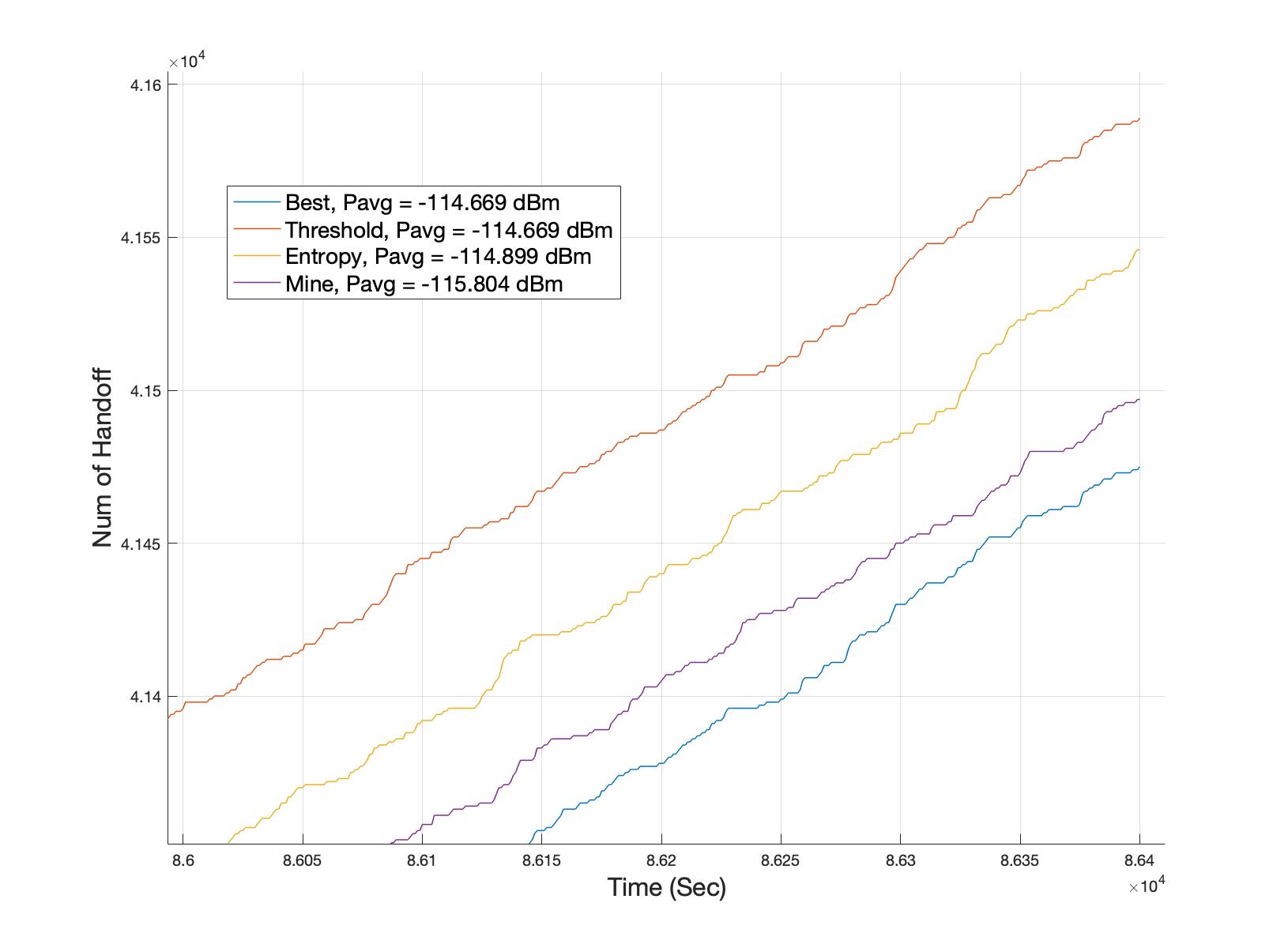
**Overview & Pavg**



**T**



**T**



1. **Source Code**

**Best Signal Method**

% (1) Best Relative Signal Method

function bestSigMethod(obj)

SignalPower = sigPower(obj);

best\_signal = max(SignalPower);

% idx of base stations that have best signal

BestSigBaseIdx = find(SignalPower == best\_signal);

% change base station

if numel(BestSigBaseIdx) == 2 && ~ismember(obj.Base(1),

BestSigBaseIdx)

obj.Base(1) = randi(BestSigBaseIdx);

obj.Handoff(1) = obj.Handoff(1) + 1;

elseif numel(BestSigBaseIdx) == 1 && obj.Base(1) ~=

BestSigBaseIdx

obj.Base(1) = BestSigBaseIdx;

obj.Handoff(1) = obj.Handoff(1) + 1;

end

% assign updated signal power to car

obj.Signal\_power(1) = best\_signal;

end

**Pseudo Code**

1. Find best out of the four BSs based on car’s current location
2. Get index of best signal BSs as BestSigBaseIdx
3. If two BSs give Pnew > Pold, randomly assign car to one of 2 BSs; add handoff
4. If only one BS gives Pnew > Pold, assign car to BS; add handoff
5. Update

**Threshold Method**

% (2) Threshold Method

function thresholdMethod(obj)

% Pold < Threshold

SignalPower = sigPower(obj);

if obj.Signal\_power(2) < obj.T

best\_signal = max(SignalPower);

% idx of base stations that have best signal

BestSigBaseIdx = find(SignalPower == best\_signal);

% change base station

if numel(BestSigBaseIdx) == 2 && ~ismember(obj.Base(2),

BestSigBaseIdx)

obj.Base(2) = randi(BestSigBaseIdx);

obj.Handoff(2) = obj.Handoff(2) + 1;

elseif numel(BestSigBaseIdx) == 1 && obj.Base(2) ~=

BestSigBaseIdx

obj.Base(2) = BestSigBaseIdx;

obj.Handoff(2) = obj.Handoff(2) + 1;

end

% assign updated signal power to car

obj.Signal\_power(2) = best\_signal;

% Pold > Threshold

else

obj.Signal\_power(2) = SignalPower(obj.Base(2));

end

end

**Pseudo Code**

1. Find corresponding s out of the four BSs based on car’s current location
2. If Pold < Threshold, get index of best signal BSs as BestSigBaseIdx
3. If two BSs give Pnew > Pold, randomly assign car to one of 2 BSs; add handoff
4. If only one BS gives Pnew > Pold, assign car to BS; add handoff
5. If Pold >= Threshold, no handoff
6. Update

**Entropy Method**

% (3) Entropy Method

function entropyMethod(obj)

SignalPower = sigPower(obj);

best\_signal = max(SignalPower);

% Pnew > Pold + E

if best\_signal > obj.Signal\_power(3) + obj.E

% idx of base stations that have best signal

BestSigBaseIdx = find(SignalPower == best\_signal);

% change base station

if numel(BestSigBaseIdx) == 2 && ~ismember(obj.Base(3),

BestSigBaseIdx)

obj.Base(3) = randi(BestSigBaseIdx);

obj.Handoff(3) = obj.Handoff(3) + 1;

elseif numel(BestSigBaseIdx) == 1 && obj.Base(3) ~=

BestSigBaseIdx

obj.Base(3) = BestSigBaseIdx;

obj.Handoff(3) = obj.Handoff(3) + 1;

end

% assign updated signal power to car

obj.Signal\_power(3) = best\_signal;

% Pnew <= Pold + E

else

obj.Signal\_power(3) = SignalPower(obj.Base(3));

end

end

**Pseudo Code**

1. Find best out of the four BSs based on car’s current location
2. If Pnew > Pold + Entropy, get index of best signal BSs as BestSigBaseIdx
3. If two BSs give Pnew > Pold, randomly assign car to one of 2 BSs; add handoff
4. If only one BS gives Pnew > Pold, assign car to BS; add handoff
5. If Pnew <= Pold + Entropy, no handoff
6. Update

**My Method (Distance Method)**

function myMethod(obj)

SignalPower = sigPower(obj);

best\_signal = max(SignalPower);

% handoff when dist to BSold > 1500m

if norm([obj.x, obj.y] - obj.BS\_Coor\_Array(obj.Base(4),:)) >

1500

% idx of base stations that have best signal

BestSigBaseIdx = find(SignalPower == best\_signal);

% change base station

if numel(BestSigBaseIdx) == 2 && ~ismember(obj.Base(4),

BestSigBaseIdx)

obj.Base(4) = randi(BestSigBaseIdx);

obj.Handoff(4) = obj.Handoff(4) + 1;

elseif numel(BestSigBaseIdx) == 1 && obj.Base(4) ~=

BestSigBaseIdx

obj.Base(4) = BestSigBaseIdx;

obj.Handoff(4) = obj.Handoff(4) + 1;

end

% assign updated signal power to car

obj.Signal\_power(4) = best\_signal;

% dist to BSold < 1500

else

obj.Signal\_power(4) = SignalPower(obj.Base(4));

end

end

**Pseudo Code**

1. Find best out of the four BSs based on car’s current location
2. If distance to original BS > 1500m, get indexes of best signal BSs
3. If two BSs give Pnew > Pold, randomly assign car to one of 2 BSs; add handoff
4. If only one BS gives Pnew > Pold, assign car to BS; add handoff
5. If distance to original BS <= 1500m, no handoff
6. Update
7. **Intro to My Policy**

My policy uses the distance between the car and the BS to determine if handoff should happen. This might sound identical to the Threshold Method in the case of free space ideal propagation, but if applied to a realistic scenario, the background influence of the environment would differ the two.

My policy has a lower handoff rate compared to the previous methods because the distance at which handoff occurs is greater than those. However, it also suffers from lower Pavg since on average the distance between transmitter and receiver is also greater.