

# Documentació de la Pràctica 1: Cerca local

Oriol Pascual  
Jordi Donadeu  
Oriol Vidal

QT1 2018-2019

## Descripció de la implementació de l'estat

Per tal d'implementar l'estat que ens demanava el problema vam crear una classe anomenada ProbCentralBoard amb els següents atributs:

- Un objecte de tipus Centrales.
- Un objecte de tipus Clientes.
- El número de clients i de centrals.
- Un vector d'enters que farem servir per representar les assignacions, on l'índex de la posició és el client i el valor d'una posició és la central a la que aquest client està assignat.
- Els paràmetres necessaris per cridar les constructores de Clientes i Centrales (propc, propg i cent).
- Un vector de pairs, amb una posició per cada central. Serveix per emmagatzemar el consum d'una central i el consum disponible.

## Descripció dels operadors escollits

- Moure client: Movem un client i a una central j i l'heurístic valora si aquest canvi és millor o pitjor en termes de benefici total.
- Buidar central: Intentem buidar una central c, intentant repartir tots els seus clients assignats a altres centrals que no estiguin plenes.
- Buidar central amb més potencial: Entenem com a central amb més potencial aquella que té molta energia disponible no assignada, és a dir, aquella que està produint molta energia però en té assignada molt poca a clients. Aquesta funció intenta assignar aquests clients a altres centrals, per intentar evitar posar en marxa aquesta central.
- Intercanviar clients: Intercanviem dos clients per intentar optimitzar les assignacions de clients a centrals.

## Descripció de les estratègies per trobar la solució inicial

En el primer operador intentem ajuntar clients garantits amb la central que pugui assumir l'energia demanada, sense tenir en compte les distàncies entre clients i centrals.

En el segon operador intentem ajuntar clients garantits amb la central més propera que pugui assumir l'energia demanda pel client.

## Descripció de la funció heurística

```
double consum_clients = board.getConsumClients();  
double consum_central= board.getConsumCentrals();  
double compensacio_clients = board.getCompensacio();  
heur=(consum_clients - consum_central – compensacio_clients);  
return -1*(heur);
```

Primer obtenim el consum dels clients que estan assignats (positiu), a continuació el consum de les centrals actives (negatiu) i finalment la compensació que haurem de pagar als clients que no tenim assignats. Llavors multipliquem el valor heurístic perquè l'heurístic intenta minimitzar el valor, però nosaltres volem maximitzar el valor absolut ja que estem parlant de benefici, per tant, haurem de fer que el valor estigui en negatiu.

## Experiments