# Problème

# • Si la quantité de quais disponibles sur une station va devenir inférieure à 10% dans les 10 prochains jours (d’après les voyages en cours et les réservations effectuées), alors des navettes doivent être transférées sur d’autres stations pour retrouver un ratio minimum de 20%.

# • Si la quantité de navettes disponibles sur une station va devenir inférieure à 10% de son nombre total de quais dans les 10 prochains jours (d’après les voyages en cours et les réservations effectuées), alors des navettes doivent être transférées vers cette station pour retrouver un ratio minium de 20%.

# • L’ordre d’urgence (quelle station nécessite le plus urgemment de dégager ses quais ou de recevoir des navettes) se fait par ordre de ratio prévisionnels croissant (ex. : une station qui n’aura que 3% de ses quais disponibles sera prioritaire par rapport à une station qui tend vers 9% de quais disponibles).

# Transfert

Transfert\_navette (Station1 , Station2 ) :

Navette 🡨 Station1.navettes [0] ;

Stations2.navattes.ajouter(Navette)

Stations1.navattes.supprimer(Navette)

//Quais dispos [Station 1 ] + +

//Quais dispos [Station 2 ] - -

# Algo

équilibrer (stations):

stations10%=ratio\_quais\_disponibles\_moins\_de\_10%(stations)

stations90%=ratio\_quais\_disponibles\_plus\_de\_90%(stations)

équilibrer\_stations10%(stations,stations10%,stations90%)

équilibrer\_stations90%(stations,stations90%,stations10%)

fin\_équilibrer

équilibrer\_stations10%(stations,stations10%,stations90%):

stations\_10%\_équilibrées = [] /\* objectif de 20% atteint \*/

retraits\_interdits = stations10% /\* stations où il n'est pas possible de retirer une navette \*/

tant\_que (stations10% - stations\_10%\_équilibrées != vide):

s\_inf = station\_moins\_occupée ( stations10% - stations\_10%\_équilibrées)

si (ratio\_apres\_ajout(s\_inf)>90%) /\*calcul prévisionnel, pas de transfert \*/

stations\_10%\_équilibrées.ajouter(s\_inf)

continue;

finsi.

tant\_que (s\_inf !€ stations\_10%\_équilibrées) :

s\_sup = station\_plus\_occupée\_priorité ( stations90%, stations,retraits\_interdits)

/\* une station à équilibrer ( € stations90%) est prioritaire si son ratio d'occupation n'a pas atteint 80% \*/

si (s\_sup == null)

retourner; /\* quitter, pas de navettes disponibles pour équilibrer \*/

sinon si (ratio\_apres\_retrait(s\_sup)<10% ) /\*calcul prévisionnel, pas de transfert \*/

retraits\_interdits.ajouter(s\_sup)

continue;

sinon

transférer\_navette(s\_sup, s\_inf)

si (ratio\_quais\_dispos(s\_inf)>=20%)

stations\_10%\_équilibrées.ajouter(s\_inf);

fin\_tant\_que

fin\_tant\_que

fin\_équilibrer\_stations10%

équilibrer\_stations90%(stations,stations90%,stations10%):

stations\_90%\_équilibrées = []

ajouts\_interdits = stations90% /\* stations où il n'est pas possible d'ajouter une navette \*/

tant\_que (stations90% - stations\_90%\_équilibrées != vide):

s\_sup = station\_plus\_occupée ( stations90% - stations\_90%\_équilibrées))

si (ratio\_apres\_retrait(s\_sup)<10%)

stations\_90%\_équilibrées.ajouter(s\_sup)

continue;

tant\_que (s\_sup !€ stations\_90%\_équilibrées) :

s\_inf = station\_moins\_occupée\_priorité ( stations10%, stations,ajouts\_interdits) /\* \*/

si (s\_inf == null)

retourner

sinon si (ratio\_apres\_ajout(s\_inf)>90% )

ajouts\_interdits.ajouter(s\_inf)

continue;

sinon

transférer\_navette(s\_sup, s\_inf)

si (ratio\_quais\_dispos(s\_sup)<=80%)

stations\_90%\_équilibrées.ajouter(s\_sup);

fin\_tant\_que

fin\_tant\_que

fin\_équilibrer\_stations90%