

Tout cela parle du prorata. J'ai mis mes pseudos code à la fin de ce pdf.

Pour répondre à votre question, j'initialise quelques agents explorateurs au début, par exemple pour deux ressources, 8 agents explorateurs, soit 4 agents explorateurs pour chaque camp pour un run ayant 100 agents. C'est-à-dire qu'il y aura toujours au moins, 4 agents explorateurs à tout moment  $t$ , ayant pour même opinion telle ressource. Cela a pour but d'éviter des trop gros écarts, par exemple un consensus (mauvais) où il n'y aurait aucun agent explorateur pour la ressource ayant la qualité la plus basse. En mettant des agents explorateurs dès le début, cela a pour conséquence de forcer l'apparition de futurs agents explorateurs pour cette ressource, car cela donne une certaine probabilité non nulle (même minime) qu'il y ait plus d'agents explorateurs pour cette ressource (puisque j'utilise le voter model et non le majority rule qui ne marcherait pas dans ce cas).

"si je comprend bien tes agents têtus sont prédéterminés...? "

Par conséquent en quelques mots, oui quelques agents têtus/explorateurs sont prédéterminés dès le début, les autres agents dits disséminateurs vont par la suite devenir des agents explorateurs, en revanche, les agents deviennent des agents explorateurs mais pour une ressource précise individuellement avec une certaine probabilité. En théorie, plus la qualité de la ressource est bonne, plus cette probabilité augmente, ce qui provoque ce "prorata". Plus la qualité est basse, moins la probabilité sera haute et moins il y aura d'agents explorateurs dessus une fois le consensus atteint.

Le risque d'obtenir un consensus trop rapide est effectivement présent, mais aussi le risque d'obtenir un consensus trop tardivement pose problèmes au cours des expérimentations. J'ai essayé quelques améliorations décrites ci-dessous.

Les autres agents dits "disséminateurs" (qui peuvent changer leur avis) vont devenir au fur et à mesure ces agents explorateurs.

1) Tout d'abord, on initialise quelques variables :

- \* Un nombre initial d'agents "explorateurs" (les agents dits "têtus") égal dans chaque équipe de ressource. Ces agents ne pourront jamais (plus) changer leur propre opinion, mais ils pourront donner leur avis aux autres agents "disséminateurs" (les agents non explorateurs, dits aussi non têtus).

- \* Un seuil max de changements d'opinions

- \* Un seuil min de changements d'opinions

- \* Une itération dite critique de décision pour le seuil min

- \* Un poids de changements d'opinions

- \* Une autre itération critique mais pour le poids

2) Je l'ai nommé "Algorithme minimal" dans le pseudo code.

Le run est fini lorsque le consensus est atteint, c'est-à-dire quand tous les agents sont des explorateurs (qui ne peuvent plus changer d'opinion). Donc tant qu'il reste des agents disséminateurs, on procède ainsi pour chaque agent disséminateur :

Chaque agent a au début, un seuil de changements d'opinions égal au seuil max de changements d'opinions, et une qualité d'une ressource. Chaque itération

correspond à un aller-retour nid - ressource. A chaque itération, les agents partagent leur opinion à leurs voisins au nid, et en récupèrent aussi de ces derniers. Ensuite ils utilisent le voter model pour choisir leur prochaine ressource à visiter. A chaque fin d'itération, ils incrémentent leur nombre de changements d'opinions global et leur nombre de changements d'opinions en faveur de la ressource qu'il vient de choisir. A la fin, dès qu'ils ont atteint leur seuil de changements d'opinions, ils choisissent chacun de devenir un agent explorateur en faveur de la ressource qu'ils ont le plus visité individuellement.

3) Ce que j'ai dit plus haut, est ce que je vous avais dit dans mes mails précédents, mais j'ai trouvé quelques ""astuces"" en plus pour essayer d'améliorer, du moins pour essayer de diminuer des imprécisions.

Pour cela, j'ai essayé de rendre dynamique le seuil de changements d'opinions, et de donner un poids supplémentaire lors de l'incrémentation de "leur nombre de changements d'opinions en faveur de la ressource qu'il vient de choisir". Néanmoins je vous avoue que c'est vraiment du bricolage en m'appuyant sur des résultats expérimentaux.

J'ai remarqué que dans l'algorithme minimal, Pour moduler le seuil de changements d'opinions pour chaque agent individuellement, j'ai remarqué qu'à certains moments des runs, en analysant l'évolution de la qualité du consensus au cours du temps pendant les runs (à ce moment-là il reste des agents disséminateurs, donc le consensus n'était pas encore atteint), la qualité est meilleure qu'à certains d'autres moments, ainsi je me suis dit qu'à cette itération précise, la ressource choisie à ce moment t est cruciale. J'ai donc choisi de manière expérimentale, cette itération, pour l'initialisation de l'algorithme. Ainsi lorsque l'agent disséminateur atteint cette itération critique, il affecte à son seuil actuel, la valeur du seuil min de changements d'opinions divisée par la valeur de la qualité de la ressource qu'il vient de choisir de visiter.

Ainsi, si le seuil min est égal à 8, et que la qualité du site qu'il vient de choisir est 2, ainsi son seuil de changements d'opinions se réduirait de 8 à 4, et ainsi devient un agent explorateur beaucoup plus tôt. Si la qualité du site est mauvais, par exemple égal à 1, alors il garderait son seuil actuel qui est le maximum. En général, plus la ressource choisie à cette itération critique est bonne, plus tôt l'agent devient un agent explorateur. Et donc plus la ressource choisie est mauvaise, plus l'agent disséminateur prendra du temps à devenir un agent explorateur.

De plus, j'ai pensé à donner plus de poids à une certaine incrémentation du changements d'opinions en faveur d'une ressource, en fonction de l'itération, un peu comme au-dessus, mais en donnant un poids. A une itération choisie, au lieu d'incrémenter de 1 "leur nombre de changements d'opinions en faveur de la ressource qu'il vient de choisir", on incrémenterait de  $1 + \text{poids choisi}$  (disons 1, donc égal à 2).

J'ai tracé ci-dessous les courbes montrant l'évolution de la qualité du consensus en fonction du temps, des paramètres choisis, et des qualités de ressources sur 50 runs. Par contre, je ne sais pas du tout si ma prise des résultats est correcte. En effet, pour tracer cette évolution moyenne, je prends les résultats toutes les 15min (si c'était le temps réel), de chaque run, je fais la moyenne sur celle-ci toutes ces 15min, et je trace les courbes. De plus, si le nombre réel d'agents divisé par 2, est supérieur au nombre d'agents théorique, alors je con-

sidère la qualité du consensus à ce moment-là, nulle.  
Le consensus est atteint lorsque la courbe devient "plate" à la fin.

A noter que si je mets les deux itérations critiques à 0 en paramètres, l'algorithme "amélioré" n'est autre que l'algorithme "minimal". Mettre à 0 signifie tout simplement la neutralisation de la dite amélioration, sachant que le dernier algorithme a deux améliorations, on peut donc soit en utiliser une seule, soit une autre, soit les deux, soit aucune (algorithme minimal dans ce cas).

Pour 2 ressources :

qA (qualité de la ressource a) = 1

qB (qualité de la ressource b) = 2

n (nombre initial d'agents explorateurs) = 6 (donc 3 pour chaque ressource)

smax (seuil maximal de changements d'opinions) = 8

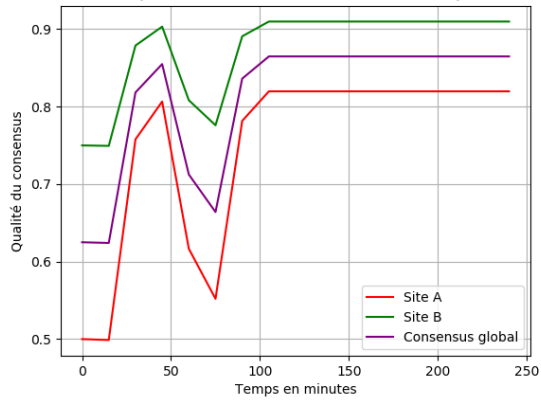
smin (seuil minimal de changements d'opinions) = 8

itCS (itération critique pour le seuil dynamique) = 2

poids (poids à ajouter) = 1

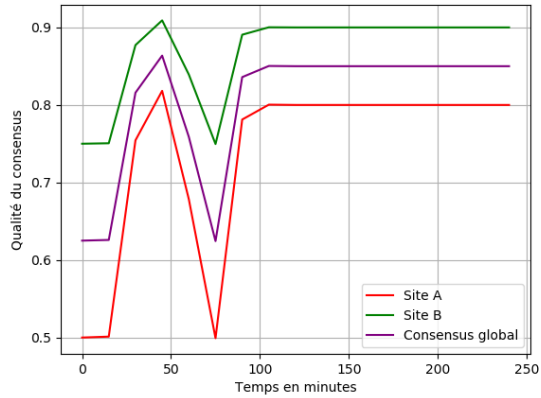
itCP (itération critique pour le poids à ajouter) = 3

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



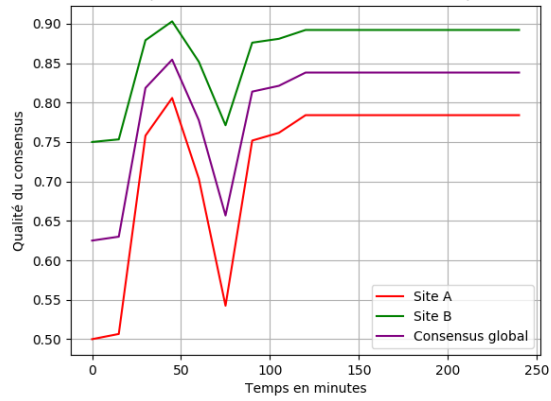
qA=1 ; qB=2 ; n=6 ; smax=8 ; smin=8 ; itCS=0 ; poids=1 ; itCP=0

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



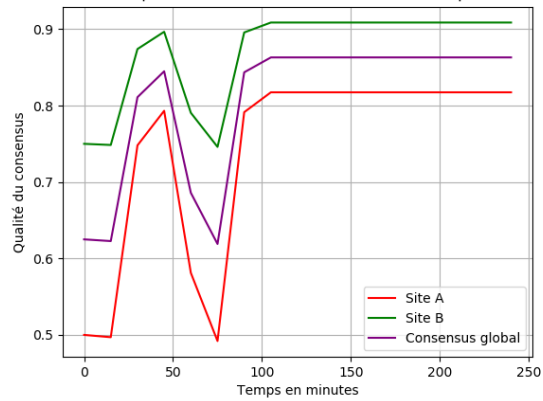
$qA=1$  ;  $qB=2$  ;  $n=6$  ;  $smax=8$  ;  $smin=8$  ;  $itCS=0$  ;  $poids=1$  ;  $itCP=3$

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



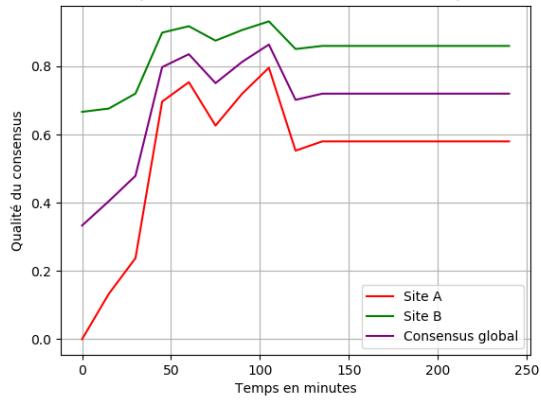
$qA=1$  ;  $qB=2$  ;  $n=6$  ;  $smax=8$  ;  $smin=8$  ;  $itCS=2$  ;  $poids=1$  ;  $itCP=0$

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



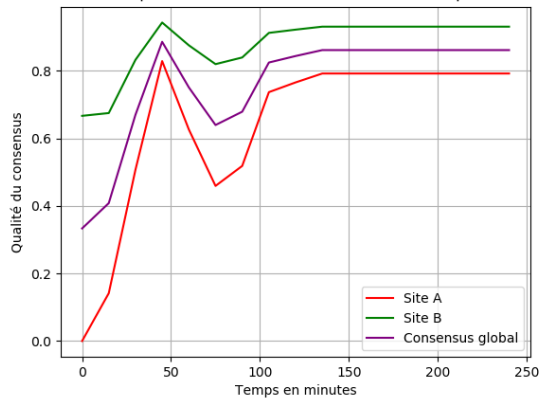
$qA=1$  ;  $qB=3$  ;  $n=6$  ;  $smax=8$  ;  $smin=8$  ;  $itCS=2$  ;  $poids=1$  ;  $itCP=3$

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



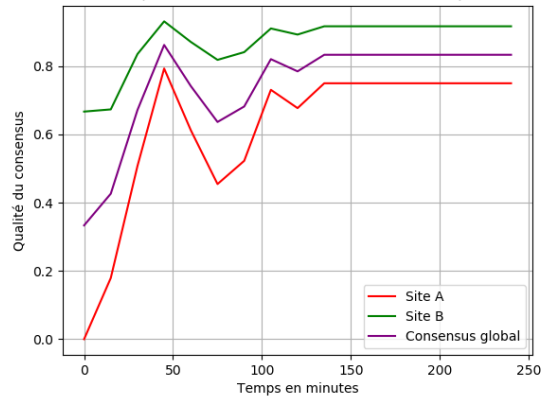
$qA=1$  ;  $qB=3$  ;  $n=6$  ;  $smax=8$  ;  $smin=8$  ;  $itCS=0$  ;  $poids=1$  ;  $itCP=0$

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



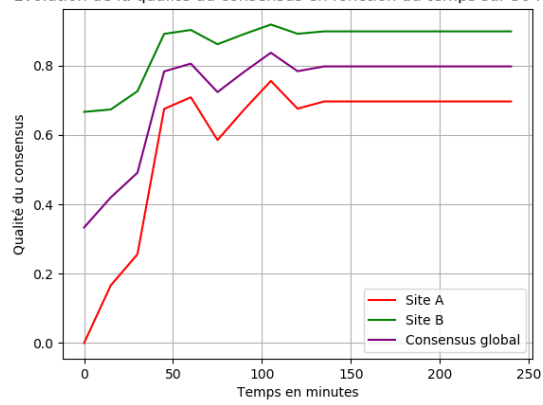
$qA=1$  ;  $qB=3$  ;  $n=6$  ;  $smax=8$  ;  $smin=8$  ;  $itCS=0$  ;  $poids=1$  ;  $itCP=3$

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



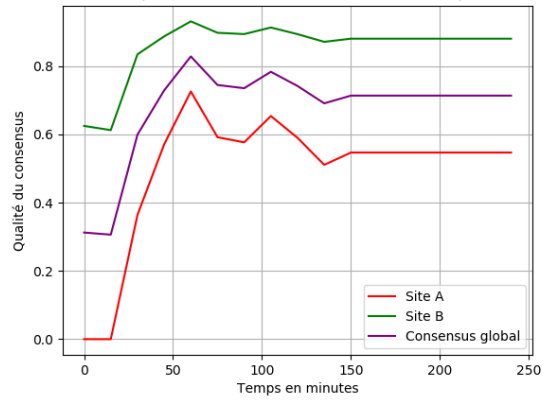
$qA=1$  ;  $qB=3$  ;  $n=6$  ;  $smax=8$  ;  $smin=8$  ;  $itCS=2$  ;  $poids=1$  ;  $itCP=0$

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



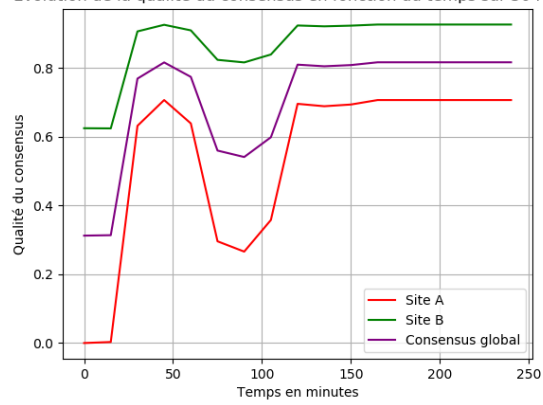
$qA=1$  ;  $qB=4$  ;  $n=6$  ;  $smax=8$  ;  $smin=8$  ;  $itCS=2$  ;  $poids=1$  ;  $itCP=3$

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



$qA=1$  ;  $qB=4$  ;  $n=6$  ;  $smax=8$  ;  $smin=8$  ;  $itCS=0$  ;  $poids=1$  ;  $itCP=0$

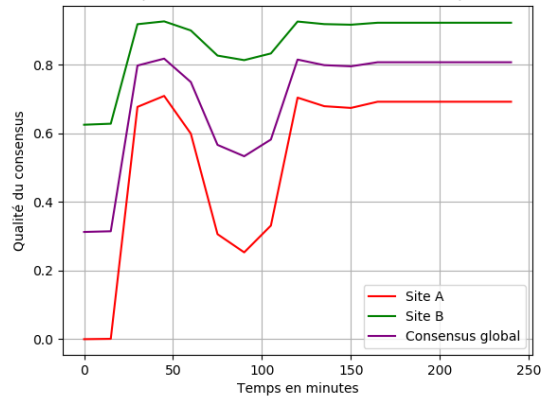
Evolution de la qualité du consensus en fonction du temps sur 50 runs :





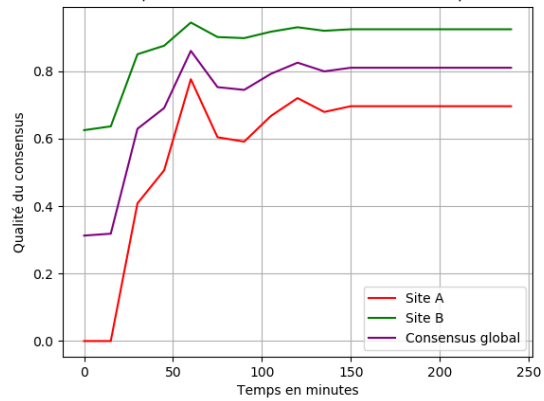
qA=1 ; qB=4 ; n=6 ; smax=8 ; smin=8 ; itCS=0 ; poids=1 ; itCP=3

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



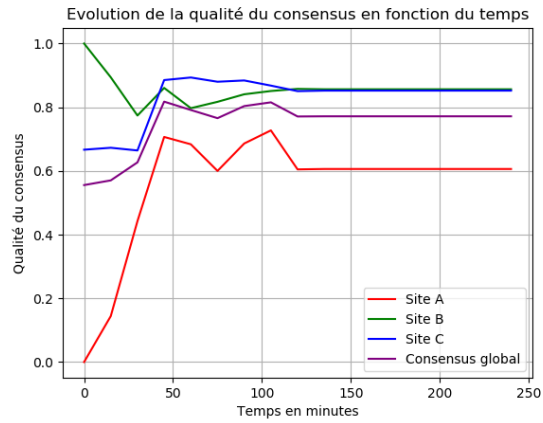
qA=1 ; qB=4 ; n=6 ; smax=8 ; smin=8 ; itCS=2 ; poids=1 ; itCP=0

Evolution de la qualité du consensus en fonction du temps sur 50 runs :

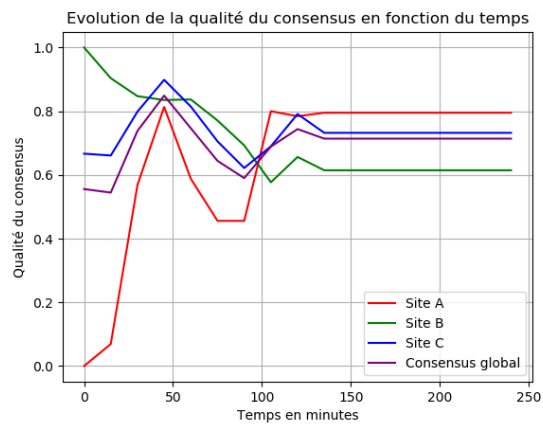


Pour 3 ressources :

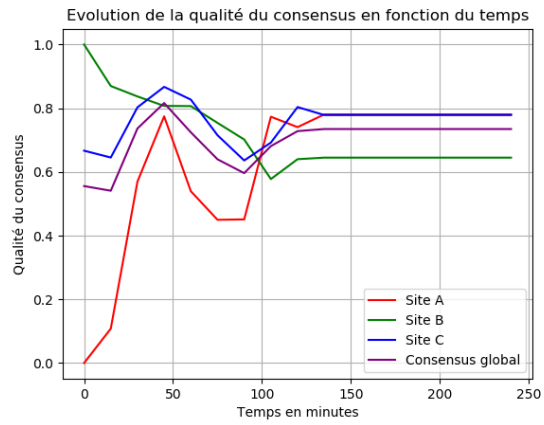
$qA=1$  ;  $qB=2$  ;  $qC=3$  ;  $n=12$  ;  $smax=8$  ;  $smin=8$  ;  $itCS=2$  ;  $poids=1$  ;  
 $itCP=3$



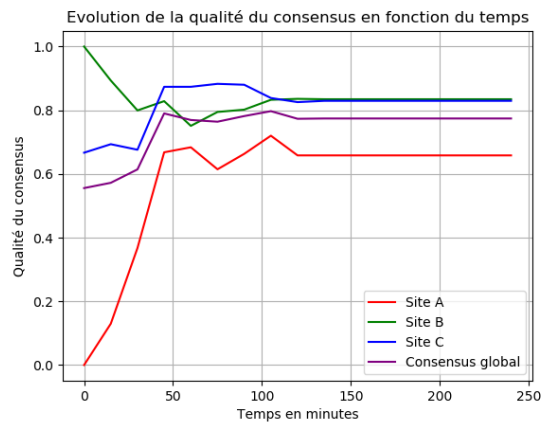
$qA=1$  ;  $qB=2$  ;  $qC=3$  ;  $n=12$  ;  $smax=8$  ;  $smin=8$  ;  $itCS=0$  ;  $poids=1$  ;  
 $itCP=0$



qA=1 ; qB=2 ; qC=3 ; n=12 ; smax=8 ; smin=8 ; itCS=0 ; poids=1 ;  
itCP=3

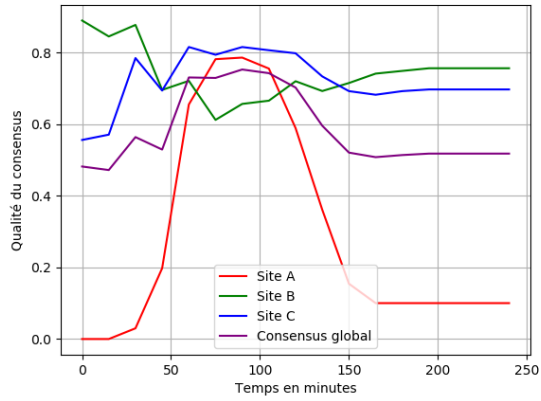


qA=1 ; qB=2 ; qC=3 ; n=12 ; smax=8 ; smin=8 ; itCS=2 ; poids=1 ;  
itCP=0



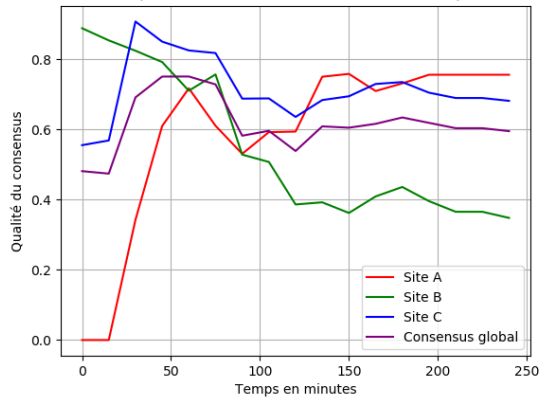
qA=1 ; qB=3 ; qC=6 ; n=12 ; smax=8 ; smin=8 ; itCS=2 ; poids=1 ;  
itCP=3

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



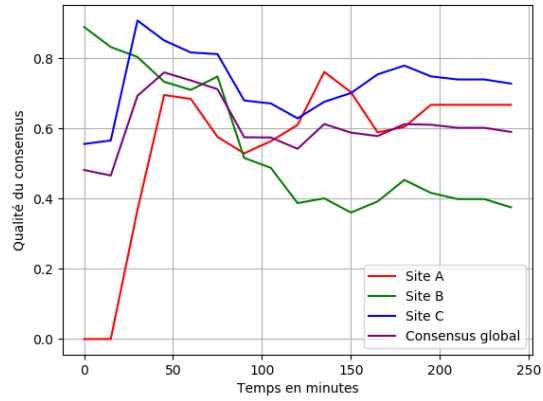
qA=1 ; qB=3 ; qC=6 ; n=12 ; smax=8 ; smin=8 ; itCS=0 ; poids=1 ;  
itCP=0

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



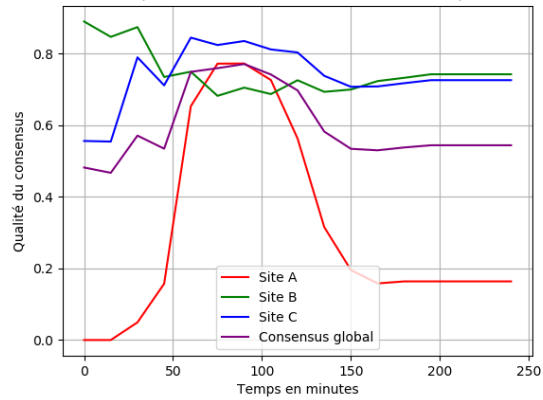
qA=1 ; qB=3 ; qC=6 ; n=12 ; smax=8 ; smin=8 ; itCS=0 ; poids=1 ;  
itCP=3

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



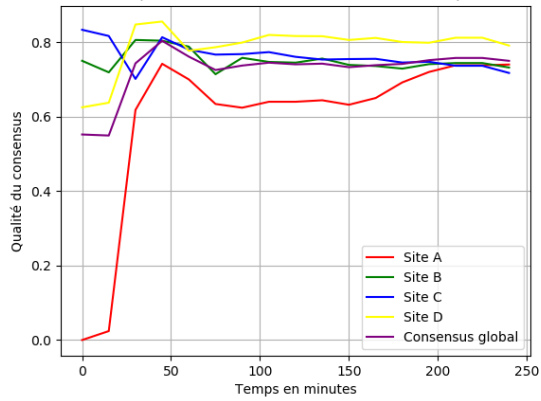
qA=1 ; qB=3 ; qC=6 ; n=12 ; smax=8 ; smin=8 ; itCS=2 ; poids=1 ;  
itCP=0

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



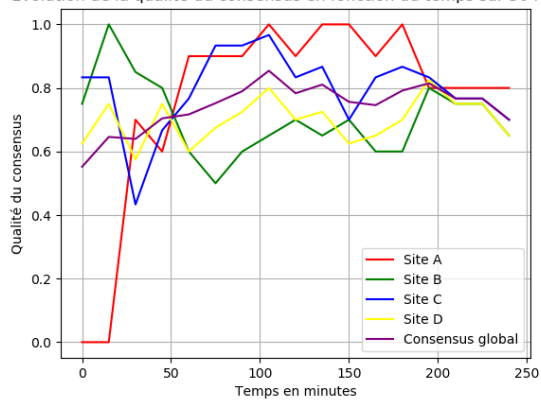
Pour 4 ressources :  
 $qA=1$  ;  $qB=2$  ;  $qC=3$  ;  $qD=4$  ;  $n=16$  ;  $smax=12$  ;  $smin=12$  ;  $itCS=2$  ;  
 $poids=1$  ;  $itCP=3$

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



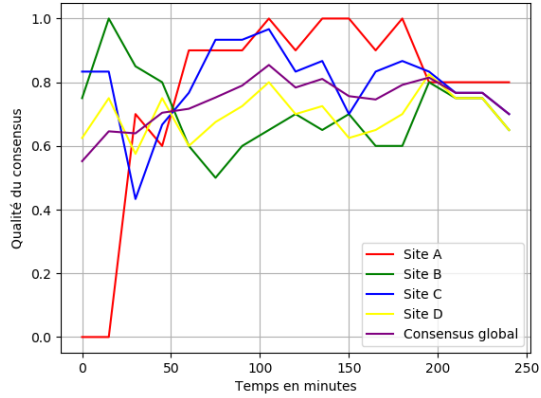
$qA=1$  ;  $qB=2$  ;  $qC=3$  ;  $qD=4$  ;  $n=16$  ;  $smax=12$  ;  $smin=12$  ;  $itCS=0$  ;  
 $poids=1$  ;  $itCP=0$

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



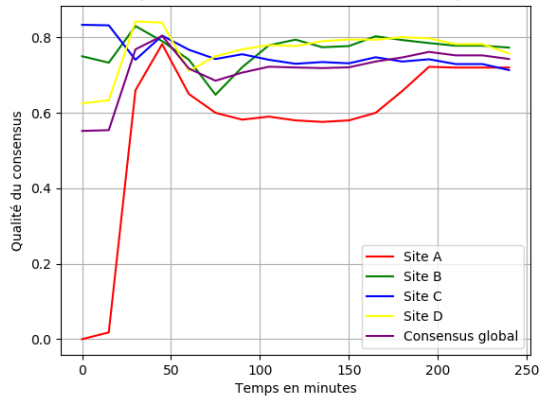
qA=1 ; qB=2 ; qC=3 ; qD=4 ; n=16 ; smax=12 ; smin=12 ; itCS=0 ;  
poids=1 ; itCP=3

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



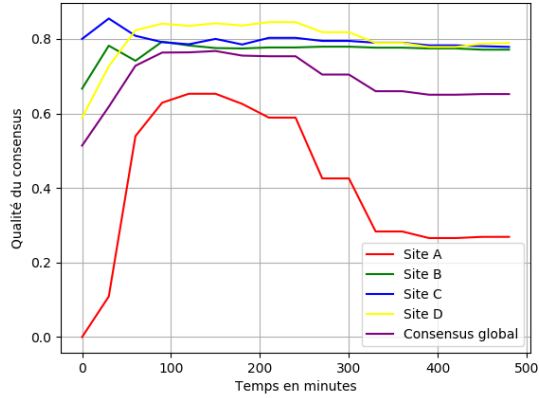
qA=1 ; qB=2 ; qC=3 ; qD=4 ; n=16 ; smax=12 ; smin=12 ; itCS=2 ;  
poids=1 ; itCP=0

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



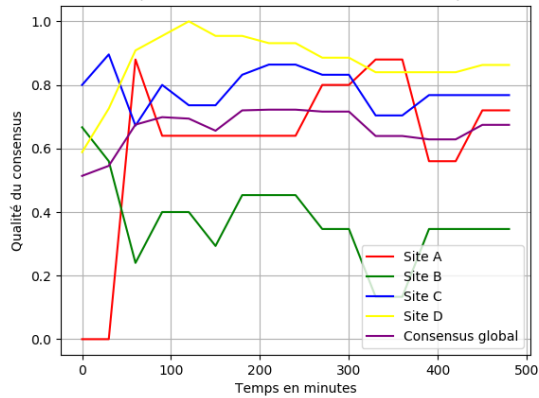
qA=1 ; qB=3 ; qC=5 ; qD=7 ; n=16 ; smax=12 ; smin=12 ; itCS=2 ;  
poids=1 ; itCP=3

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



qA=1 ; qB=3 ; qC=5 ; qD=7 ; n=16 ; smax=12 ; smin=12 ; itCS=0 ;  
poids=1 ; itCP=0

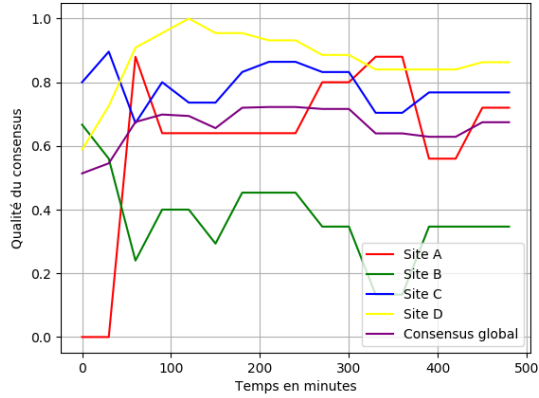
Evolution de la qualité du consensus en fonction du temps sur 50 runs :





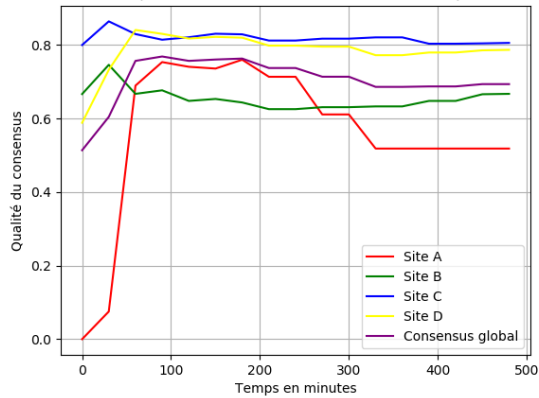
qA=1 ; qB=3 ; qC=5 ; qD=7 ; n=16 ; smax=12 ; smin=12 ; itCS=0 ;  
poids=1 ; itCP=3

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



qA=1 ; qB=3 ; qC=5 ; qD=7 ; n=16 ; smax=12 ; smin=12 ; itCS=2 ;  
poids=1 ; itCP=0

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



---

**Algorithm 1:** Algorithme "minimal"

---

```
n : nombre initial d'agents explorateurs ;
nbreAgents : nombre total d'agents ;
smax : seuil maximal de changements d'opinions ;
nbreRessources : nombre de ressources ;
avisCourant[nbreRessources] : liste des nombres d'opinions choisies ;
indiceC : indice de la ressource courante choisie ;
while  $n \neq \text{nbreAgents}$  do
  foreach agent disséminateur do
    while  $\text{changementsOpinions} \neq \text{smax}$  do
      changementsOpinions += 1 ;
      avisCourant[indiceC] += 1 ;
    end
    n += 1 ;
  ;
end
```

---

---

**Algorithm 2:** Algorithme "amélioré"

---

```
n : nombre initial d'agents explorateurs ;
nbreAgents : nombre total d'agents ;
smax : seuil maximal de changements d'opinions ;
smin : seuil minimal de changements d'opinions ;
itCS : itération critique pour le seuil dynamique ;
poids : poids à ajouter ;
itCP : itération critique pour le poids à ajouter ;
nbreRessources : nombre de ressources ;
avisCourant[nbreRessources] : liste des nombres d'opinions choisies ;
indiceC : indice de la ressource courante choisie ;
qualiteC : qualité de la ressource courante choisie ;
while  $n \neq \text{nbreAgents}$  do
  foreach agent disséminateur do
    while  $\text{changementsOpinions} \neq \text{smax}$  do
      changementsOpinions += 1 ;
      avisCourant[indiceC] += 1 ;
      if  $\text{changementsOpinions} == \text{itCP}$  then
        | avisCourant[indiceC] += poids;
      end
      if  $\text{changementsOpinions} == \text{itCS}$  then
        | smax = smin / qualiteC ;
        | if  $\text{smax} < \text{changementsOpinions}$  then
          | | changementsOpinions = smax ;
        | end
      end
    end
    n += 1 ;
  ;
end
```

---