On pose T un ensemble de n tache periodique independente entre elle t1, t2, … tn. Chaque tache ti e T est caracterisé par {Ci,Di,Ti} où :

Ci est le pire temps de reponse

Ti est la periode

Di est l’écheance ( deadline ) de la tache

La taux d’utilisation de la tache est donnée par Ui = Ci / Ti

Chasue tache à un temps d’arrivé phi i . On considere que le phi i =0 qqs i = 1 .. n .

Periode harminique :

Nous definnisons la periode harmonique par Th tel que :

Th = min {Ti}/2 si exist tj e T tq Tj < 2\*min{Ti}

Th = min{Ti} sinon

Deadline monotonic :

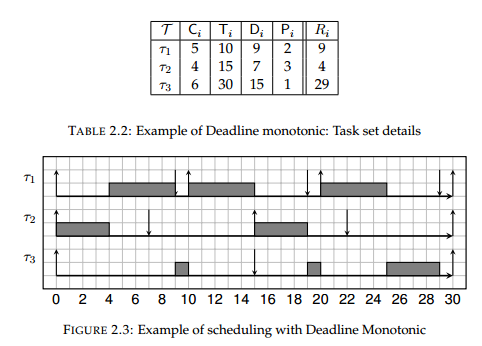
Deadline monotonic est un algorithme d’ordonnancement à priorité fixe par tache, les priorités sont accordées au taches selon leurs deadlines, la plus grande priorité est accordée a la tache de plus petites deadline.

Joseph et al ont proposé un test d’ordonnancabilité basé sur le pire temps de reponse Ri.

Le pire temps de reponse est le momont ou la tache i de priorité p terminera son execution qaund les taches les plus prioritaire sont actifs avec elle en meme temps.

Nous pouvons utilisé l’eqaution recurrente suivante pour le calcul du pire temps de reponse d’une tache :

Exmeple : posons T un ensemble de 3 taches (voir table )



En utilisant l’equation , on obtient le resultat de la coolonne R . on remarque que le pire temps de reponse de la tache t3 est 29 et il est superieur à sa deadlien qui est 15. Donc, la tache n’est pas ordonnancable. La simulation confirme le fait .

Test d’ordonncabilité :

Soit T ={t1,t2,…tn} un ensemble de n taches.

T est ordonnancable si qqs ti e T Ri < Di

Tache d’endormissement :

Posons une tache d’endormissement tsleep ={Csleep,Dsleep,Tsleep} tel que :

Dsleep = Tsleep = Th

Csleep = max ( c ) tel que T u Tsleep est ordonnancable par Deadline monotonic

/ Csleepmin <= C <= Csleepmax avec Csleepmax = (1- somme Ui)\*TSleep