**Strategie trx PI**

**Stratégie + 8 :**

Tout d abord lors du lancement de programme ( dans AFP) nous enregistrons dans un vecteur

**vector < pair<int,double> > MaxCostITrx**

pour chaque **trxID**, on enregistre sa valeur d’interférence la plus grande dans le réseau. Ce vecteur est fixe.

Dans la procedure de choix de trx du joueur PI (**PIChooseTrx(int StrategyTrx,vector<TRXs> LTrxNTreateds,TRXs &trx,AFP\* afpProblem,Individual\* &Sol**)) , nous appliquons cettte stratégie.

Le principe est comme suit, Parmi les trx non traité **LTrxNTreateds,** nous choisisons, le trx qui a le plus grand coût, on fait appelle à l’in formation dans **MaxCostITrx**

**Stratégie + 9 :**

Tout d abord lors du lancement de programme ( dans AFP) nous enregistrons dans un vecteur

**vector < pair<int,double> > MaxCostITrx**

pour chaque **trxID**, on enregistre sa valeur d’interférence la plus grande dans le réseau. Ce vecteur est fixe.

Dans la procédure de choix de trx du joueur PI (**PIChooseTrx(int StrategyTrx,vector<TRXs> LTrxNTreateds,TRXs &trx,AFP\* afpProblem,Individual\* &Sol**)) , nous appliquons cettte stratégie.

Le principe est comme suit, Parmi les trx non traité **LTrxNTreateds,** nous choisisons, le trx qui a le plus **petit** coût, on fait appelle à l’in formation dans **MaxCostITrx**

**Stratégie + 10 :** avec réaffectation : dans chaque ajout , nous enregistrons dans PlanCC le cout ajouté , nous evaluons le nouveau cout avec les autres freq dans sa liste , si c amélioré on valide la réafectation et on mit ajour( freq +) dans sa liste de freq,