

## Notes sur le TP

### 1. Méthodes

Il y'a 4 méthodes de comparées : `bit_random`, `bit_greedy`, `bit_annealing`, `bit_annealing_improved`. Pour faire tourner une méthode, il faut se placer dans le dossier *epsilon* et saisir la commande : `python3 snp.py -m nom_algo` avec `nom_algo` parmi les 4 cités.

Par défaut, les algorithmes tournent une fois sur 300 itérations. On peut en revanche modifier le nombre de runs en ajoutant `-runs nb_runs`

L'algorithme d'intérêt est le *bit\_annealing\_improved* composée d'une partie (10%) de random, et d'une seconde partie de recuit simulé. Attention, l'évolution de la température se fait de manière décroissante et linéaire entre 1 et 0.1

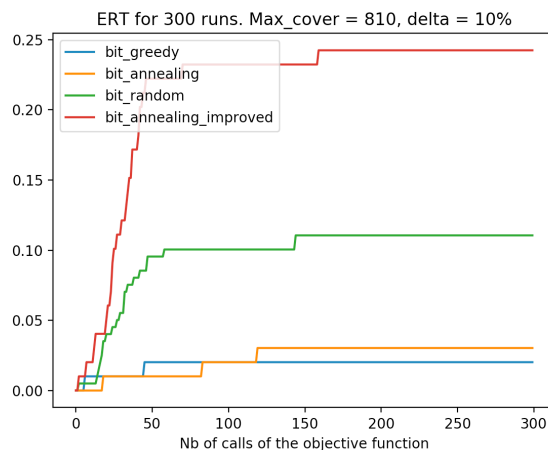
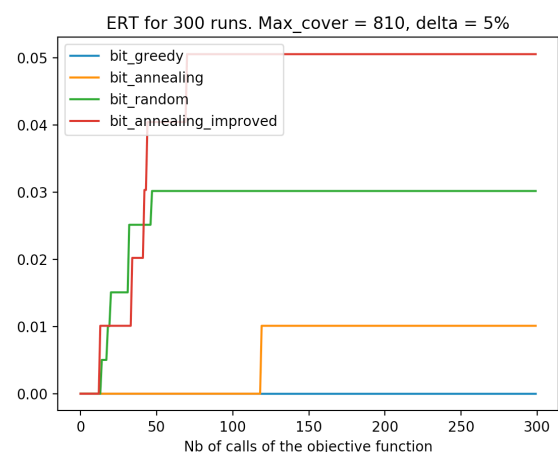
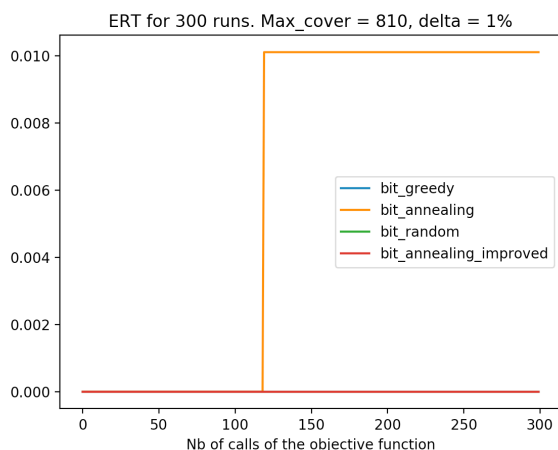
### 2. ERT

Pour lancer l'ERT, il faut se placer dans le dossier *epsilon* et saisir la commande `python3 create_graphs.py`.

Les simulations effectuées ont été faites avec 4 capteurs, sur un domaine de 30x30, 100 runs, avec 300 itérations maximales, un steady-delta de 50 et une couverture maximale obtenue de 810.

### 3. Discussion

Nous pouvons commenter les trois graphes ci dessous correspondant à  $\delta = 1\%, 5\%, 10\%$ :



- Pour  $\delta = 1\%$ , on remarque que la plus grande couverture a été obtenue pour la méthode *bit\_annealing*. Celle-ci est donc la meilleure d'un point de vue résultat de la valeur finale. En revanche, on note que la valeur max n'a été obtenue qu'une fois ( $0.010 = 1\%$  de 100 runs) et ne permet pas de conclure statistiquement.
- Pour  $\delta = 5\%$ , *bit\_annealing* est déjà moins performant que *bit\_random*. On réalise donc qu'il s'agit d'un problème facile. La méthode *bit\_annealing\_improved* semble quant à elle mieux que *bit\_random*. En revanche, les pourcentages d'atteinte de la valeur max restent très faibles ( $<5\%$ )
- Pour  $\delta = 10\%$ , *bit\_annealing\_improved* confirme l'intuition. On observe également que sa vitesse est particulièrement rapide. En effet, après exactement 30 itérations à se comporter comme un random, c'est essentiellement dans les 20 itérations suivantes que l'algorithme atteint le seuil supérieur (lorsqu'il l'atteint). Le plateau qui suit est synonyme d'une tendance forte à se concentrer dans des minima locaux (étant donné que seulement 25% des runs atteignent le seuil)

Enfin, en utilisant comme limite supérieure la couverture maximale obtenue pour *bit\_annealing\_improved*, on observe sur la figure de gauche que la probabilité d'atteindre 90% de la valeur max avoisine les 60% au bout de 60 itérations. On peut également reprendre l'observation faite au dernier point du paragraphe précédent. La figure de droite représente la position et couverture de la meilleure solution

