BE1 : AlgoGN

Problème 1 : On considère un jeu à deux joueurs dont les règles sont les suivantes :
- en même temps le joueur 1 et le joueur 2 annonce "pair" ou "impair"
- si les deux joueurs ont dit "pair" OU si les deux joueurs ont dit "impair", le joueur 1 gagne 1 euro et le joueur 2 perd 1 euro
- si l'un a dit "pair" et l'autre impair, le joueur 2 gagne 1 euro et le joueur 1 perd 1 euro

Une stratégie pour le joueur 1 est de jouer avec probabilité p_1^1 "pair" et p_2^1 "impair"
Une stratégie pour le joueur 2 est de jouer avec probabilité p_1^2 "pair" et p_2^2 "impair"

Le but est de trouver p_1^1, p_2^1, p_1^2 et p_2^2 qui maximise l'espérance de gain pour chacun des joueurs.

1) Calculer l'espérance de gain pour le joueur 1 lorsque :
- le joueur 1 joue avec probabilité p_1^1 "pair" et p_2^1 "impair"
- le joueur 2 joue avec probabilité p_1^2 "pair" et p_2^2 "impair"

Calculer l'espérance de gain pour le joueur 1 lorsque :
- le joueur 1 joue avec probabilité p_1^1 "pair" et p_2^1 "impair"
- contre une population de $s = 1..N_2$ où le joueur $\{2, s\}$ joue avec probabilité $p_1^{2,s}$ "pair" et $p_2^{2,s}$ "impair"

2) On considère l'algorithme génétique suivant :
α) Créer une population 1 (respectivement 2) de taille N_1 (resp. N_2) ayant des stratégies choisit au hasard
β) Trouver $rang_1^{max,0}$ (resp. $rang_2^{max,0}$) l'individu de la population 1 (resp. 2) ayant la meilleure espérance de gain face à la population 2 (resp. 1).

γ) "Au temps k"
- Trouver $rang_1^{max,k}$ (resp. $rang_1^{min,k}$) l'individu de la population 1 ayant la meilleure (resp. la pire) espérance de gain face à l'individu $rang_2^{max,k-1}$ de la population 2.
- Trouver $rang_2^{max,k}$ (resp. $rang_2^{min,k}$) l'individu de la population 2 ayant la meilleure (resp. la pire) espérance de gain face à l'individu $rang_1^{max,k-1}$ de la population 1.
- Dans les deux populations, on élimine le pire individu et on le remplace par une mutation du meilleur.
- On incrémente k de 1 et on recommence γ tant que $k < temps_{max}$

Implanter cet algorithme sur ce problème, et étudier sa convergence pour différentes valeurs de $N_1 = N_2$ et $temps_{max}$. Qu'observe t'on (voir les graphes de convergence) ? Peut on l'expliquer ?

3) Faire la même chose pour le jeu de pierre papier ciseaux. Quel est la meilleure façon de jouer ?

Problème 2 : Jeu de Poker Le jeu comporte deux joueurs. Chacun tire un nombre entre 0 et 1 de manière aléatoire (suivant une loi uniforme) de manière indépendante.

Les stratégies possibles du joueur 1 sont $\{parier,\ ne\ pas\ parier\}$.
Les stratégies possibles du joueur 2 sont $\{appeler,\ se\ coucher\}$.

α) Si le joueur 1 parie et le joueur 2 se couche ALORS le joueur 1 gagne 1 euro et le joueur 2 perds 1 euro.

β) Si le joueur 1 parie et le joueur 2 appelle ALORS :
- si $X > Y$ alors le joueur 1 gagne $1 + B$ euros et le joueur 2 perds $1 + B$ euros (B est le montant du pari).

- si $X < Y$ alors le joueur 2 gagne $1 + B$ euros et le joueur 1 perds $1 + B$ euros (B est le montant du pari).

- si $X = Y$ alors le jeu est nul (ni gain, ni perte).

γ_{Borel}) Si le joueur 1 ne parie pas ALORS le joueur 2 gagne 1 euro et le joueur 1 perds 1 euro
Pour simplifier le problème :

- on considère qu'une stratégie pour le joueur 1 est une variable aléatoire $S1(X)$ qui renvoie *parier* avec probabilité $p(X)$ et *ne pas parier* avec probabilité $1 - p(X)$

- on considère qu'une stratégie pour le joueur 2 est une variable aléatoire $S2(Y)$ qui renvoie *appeler* avec probabilité $q(Y)$ et *se coucher* avec probabilité $1 - q(Y)$

p et q sont des fonctions en escaliers (soit des vecteurs en matlab), i.e.

$$p(X) = p_i, \quad si \;\; X \in [ih, (i+1)h]$$

$$q(Y) = q_i, \quad si \;\; Y \in [ih, (i+1)h]$$

avec $h = 1/10$ par exemple.

1) En notant que

$$E(Gain_{Joueur_2})(p,q) = -1 + 2\int_0^1 p(x)dx - (2+B)\int_0^1\int_0^1 p(x)q(y)dxdy + 2(1+B)\int_0^1 (\int_0^x p(x)q(y)dy)dx$$

Trouver le vecteur de meilleure stratégie pour le joueur 1 (et le joueur 2).

2) Même question pour le jeu avec la condition,

γ_{VN}) Si le joueur 1 ne parie pas ALORS :
- si $X > Y$ alors le joueur 1 gagne 1 euro et le joueur 2 perds 1 euro.

- si $X < Y$ alors le joueur 2 gagne 1 euro et le joueur 1 perds 1 euro.

- si $X = Y$ alors le jeu est nul (ni gain, ni perte).

Problème 3 : Un texte contenant des mots français (non accentués et de longueur de 5 lettres au plus) a été codé en permutant les 26 lettres de l'alphabet. On se donne cinq dictionnaires de mots français de 1 à 5 lettres respectivement.

Le script Matlab : "LectureFichierTexte.m" permet de charger en mémoire les dictionnaires de 1 à 5 lettre en code ASCII (c'est à dire $a \mapsto 97$... $z \mapsto 122$ et "espace=32") et le texte codé dans la $TexteCodeEnASCII$ de taille (nombre de mots $\times$ 5).

Par soucis de simplicité on crée 5 matrices contenant les mots de 1 à 5 lettres du texte codé.

On pourra utiliser/modifier la fonction $distanceTextePourUnePermutation(P)$ qui a une permutation des lettre (c'est à dire P est une permutation du vecteur [97 : 122] (représentant le code ASCII)) renvoie la distance du texte codé dont on a permuté les lettres aux dictionnaires des mots français existant.

On pourra utiliser/modifier la fonction $AffichageTexte(P, Texte)$ permet d'afficher un texte (texte codé) ayant subit une permutation de lettre P.

On pourra utiliser/modifier la fonction $[P] = PermuationAleatoire(W)$ qui échange deux éléments de W

pour créer la nouvelle permutation P.

1) Essayer de retrouver la phrase avant codage.