

Systèmes complexes

Master 2 SDIA

Correction de l'examen du 08-01-2024

Question 1 (4 pts):

L'algorithme PSO peut être appliqué aux problèmes binaires et de permutation en utilisant des fonctions ou des règles pour transformer la représentation réelle du vecteur de position en une chaîne binaire ou en une permutation. Cette transformation est réalisée avant l'évaluation de fitness ; les équations originales de modification de la position et de la vitesse restent inchangées. (1 pt)

Pour l'espace binaire, on peut, par exemple, varier les valeurs réelles de position dans l'intervalle $[0, 1]$; on fait la transformation comme suit : les valeurs qui sont inférieures à 0.5 sont mappés 0 et les autres valeurs (qui sont ≥ 0.5) sont mappés 1. (1.5 pt)

Exemple illustrative ($n = 8$): pour $X_i = [0.74, 0.22, 0.84, 0.45, 0.31, 0.12, 0.53, 0.95]$ la solution discrète correspondante est : $S = [1\ 0\ 1\ 0\ 0\ 0\ 1\ 1]$

Pour l'espace combinatoire de permutation, une approche populaire consiste à trier les valeurs réelles de position de façon croissante, puis à construire une permutation par les indexes $\{1, 2, \dots, n\}$ dans la représentation réelle des valeurs triées. (1.5 pt)

Exemple illustrative ($n = 8$): pour $X_i = [3.4, 1.72, 0.74, 2.45, 3.07, 2.67, 0.95, 1.14]$ la solution discrète correspondante est : $P = [3\ 7\ 8\ 2\ 4\ 6\ 5\ 1]$

Question 2 (4 pts):

a) Calculer les probabilités de sélectionner chaque individu $i = 1, 2, \dots, 8$ en utilisant la méthode de la roulette de casino (2 pts).

Individu i	Fitness F_i	Probabilité de sélection P_i
1	0.875	0.21875
2	0.25	0.0625
3	0.5	0.125
4	0.375	0.09375
5	0.625	0.15625
6	0.125	0.03125
7	0.75	0.1875
8	0.5	0.125
Somme	4.0	1.0

b) Ecrire l'algorithme de sélection par roulette de casino (2 pts).

```
// retourner l'index dans {1, 2, ... M} de l'individu sélectionné
int roulette_wheel_selection()
{
    int selected_idv;
    double cumulated_prob[20], sum_fitness = 0.0, rho;
    for (int i = 0; i < M; i++) {
        sum_fitness += F[i];
    }
    /* sélectionner un individu selon la probabilité */
    rho = rand01(&seed); // génération d'un nombre aléatoire dans [0, 1]
    cumulated_prob[0] = F[0] / sum_fitness;
    if (rnd <= cumulated_prob [0])
    {
        selected_idv = 0;
    }
    Else
    {
        cumulated_prob[1] = F[1] / sum_fitness + cumulated_prob[0];
        for (int i = 1; i < M; i++)
        {
            if (rho <= cumulated_prob[i])
            {
                selected_idv = i;
                break;
            }
            cumulated_prob[i + 1] = F[i + 1] / sum_fitness + cumulated_prob[i];
        }
    }
    return selected_idv;
}
```

Problème (12 pts)

1. Codage de solution (2 pt)

La représentation de solution de ce problème correspond à un vecteur (X) de n entiers dans $\{1, 2, \dots, K\}$

2	K	3	$K - 1$...	1	$K - 2$
---	-----	---	---------	-----	---	---------

2. Taille de l'espace de recherche (2 pt)

Le problème de classification de n objets en K classes est un problème d'optimisation NP-difficile avec K^n configurations possibles.

3. Génération d'une population initiale (2 pt)

La population initiale (de M individus) est générée de façon aléatoire

```
// la population est donnée par la matrice X de taille m * n
void generate_initial_pop()
{
    for (int i = 0 ; i < M; i++)
    {
        for (int j = 0; j < n; j++)
        {
            X[i][j] = random(K + 1);
            // génération d'un nombre aléatoire entier dans [1..K]
        }
    }
}
```

4. Formulation de la fonction fitness (3 pt)

Soient g_1, g_2, \dots, g_k les centres de gravités de K classes, et g le centre de gravités de tous les (n) objets.

L'inertie intra de chaque classe CL_k , $k = 1, 2, \dots, K$ est donnée par (1 pt):

$$Intra_k = \frac{1}{n_k} \sum_{w \in CL_k} \|w_j - g_{k,j}\|^2,$$

où n_k est le nombre de objets dans la classe CL_k . La somme de inertie intra-classe est donnée par :

$$INTRA = \sum_{k=1}^K Intra_k$$

L'inertie inter-classe est donnée par (1 pt) :

$$INTER = \frac{1}{K} \sum_{k=1}^K \sum_{j=1}^n \|g_{k,j} - g_j\|^2,$$

où n_k est le nombre de objets dans la classe k .

La fonction fitness est donnée par (1 pt) :

$$F(X) = INTRA + INTER$$

5. Un opérateur de croisement et un opérateur de mutation (3 pt)

On peut appliquer un opérateur de croisement à deux point.

Voici un exemple pour $n = 8$ et $K = 4$.

P1:

2	1	1	3	2	4	1	4
---	---	---	---	---	---	---	---

P1:

4	3	1	2	2	3	1	2
---	---	---	---	---	---	---	---

E1:

2	1	1	2	2	3	1	4
---	---	---	---	---	---	---	---

E2:

4	3	1	3	2	4	1	2
---	---	---	---	---	---	---	---

Pour la mutation, on peut appliquer changement aléatoire d'un gène sélectionné aléatoirement.

E :

2	1	1	2	2	3	1	4
---	---	---	---	---	---	---	---

E' :

2	1	1	2	2	2	1	4
---	---	---	---	---	---	---	---