

Université de Mila
Faculté de Mathématiques et informatique
Département d'informatique

1ère année Master I2A

Année / 2025/2026

Matière : Résolution de problèmes et optimisation combinatoire

Date : 17/05/2026

Durée : 2h00

Examen du deuxième semestre

Exercice 1 : (3 pts)

Donner la ou les bonne(s) réponse(s) pour chacune des questions suivantes :

1. Dans un problème d'optimisation, si l'objectif est de minimiser un ou plusieurs critère(s), alors il est facile de trouver la borne supérieure et le défi major est de trouver la borne inférieure (vrai ou faux) (1 pt) ?
2. Dans un algorithme génétique, si nous utilisons la méthode élitiste dans l'étape de remplacement, alors il est recommandé de (1 pt) :
 - Diminuer la probabilité de croisement. (La mettre : $0.5 \leq pc \leq 0.6$)
 - Augmenter la probabilité de croisement. (La mettre : $0.9 \leq pc \leq 0.95$)
3. Quel est le rôle de l'opérateur de mutation dans un algorithme génétique ? (1 pt)

Problème : (17 pts)

Soit E un ensemble de n entiers (n est un nombre pair) et on considère un entier *cible*. Notre problème consiste à trouver un sous-ensemble S de n/2 entiers dont la somme des entiers dans S est la plus proche possible de l'entier *cible*. Soit une instance simple de ce problème, avec n = 10 et cible = 15.

30	-20	-15	10	8	22	18	2	14	17
----	-----	-----	----	---	----	----	---	----	----

1. Donner 2 solutions réalisables avec une représentation binaire, 2 solutions réalisables avec une représentation réel et 2 solutions irréalisables. (3 pts)

On cherche à résoudre ce problème avec un algorithme génétique en considérant la représentation binaire des solutions. Soit *elements*[] le tableau qui contient les entiers:

2. En modélisant le problème comme un problème de minimisation. Proposer une fonction qui permet d'évaluer et de calculer le fitness d'une solution *Sol* passée en paramètre. (4 pts)
3. Ecrire la fonction qui implémente l'opérateur de croisement à deux points entre deux parents PS1 et PS2 ? (2 pts)
4. Il est clair que la fonction qui implémente le croisement à deux points crée parfois des solutions irréalisables. Ecrire une fonction qui permet de tester la réalisabilité d'une solution *sol* passée en paramètre et si *sol* n'est pas réalisable, alors la fonction modifie *sol* pour la rendre réalisable sinon la fonction retourne *sol* sans modification? (4 pts)
5. Ecrire une fonction qui implémente le croisement uniforme entre deux parents PS1 et PS2 et qui retourne toujours une solution réalisable (sans aucune intervention d'une fonction de correction). (4 pts)

Université de Mila
Faculté de Mathématiques et informatique
Département d'informatique

1ère année Master I2A

Année 2025/2026

Matière : Résolution de problèmes et optimisation combinatoire

Date : 17/05/2026

Durée : 2h00

Corrigé type de l'examen du deuxième semestre

Exercice 1 : (3 pts)

Donner la ou les bonne(s) réponse(s) pour chacune des questions suivantes :

1. Vrai (1 pt)
2. Dans un algorithme génétique, si nous utilisons la méthode élitiste dans l'étape de remplacement, alors il est mieux de (1 pt) :
« Augmenter la probabilité de croisement. (La mettre : $0.9 \leq p_c \leq 0.95$) »
3. Le rôle de l'opérateur de mutation dans un algorithme génétique est d'assurer la diversité dans la population et par conséquent il assure une bonne exploration de l'espace de recherche. (1 pt)

Problème : (17 pts)

Soit l'instance simple du problème, avec $n = 10$ et cible = 15.

30	-20	-15	10	8	22	18	2	14	17
----	-----	-----	----	---	----	----	---	----	----

4. Donner 2 solutions réalisables avec une représentation binaire, 2 solutions avec une représentation réelle et 2 solutions irréalisables :

2 Solutions réalisables avec représentation réelle :

S1 : {30, -20, 8, 22, 14}

S2 : {-20, -15, 10, 2, 17}

2 Solutions réalisables avec représentation binaire :

S3 :

0	0	1	1	1	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---

S4 :

1	0	1	0	1	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---

2 Solutions irréalisables :

S5 : {30, -20, 8, 22, 14, 10}

S6 :

1	0	1	0	1	0	1	1	1	1
---	---	---	---	---	---	---	---	---	---

On cherche à résoudre ce problème avec un algorithme génétique en considérant la représentation binaire des solutions. Soit *elements*[] le tableau qui contient les entiers:

5. En modélisant le problème comme un problème de minimisation. Proposer une fonction qui permet d'évaluer et de calculer le fitness d'une solution *Sol* passée en paramètre. (4 pts)

Fonction Evaluer (Sol[], elements[] : tableau d'entiers, n, Cible : entier) : entier

Variable

somme : entier ;

Debut

Pour i = 1 à n faire

somme ← somme + Sol[i] × elements[i] ;

FinPour

Si(Cible > somme) alors

Retourner (Cible-somme) ;

Sinon

Retourner (somme-Cible) ;

FinSi

Fin

6. Ecrire la fonction qui implémente l'opérateur de croisement à deux points ? (2 pts)

Fonction Croisement2Points (SP1[], SP2[] : tableau d'entiers, n : entier) : Matrice d'entiers

Variable

Point1, Point2 : entier ;

Enfants[][] : matrice d'entiers ;

Début

Point1 ← entierAleatoire(1, n - 2) ;

Point2 ← entierAleatoire(Point1 + 1, n - 1) ;

Pour i = 1 à Point1 faire

Enfants[2][i] ← SP1[i]

Enfants[1][i] ← SP2[i]

FinPour

Pour i = Point1 + 1 à Point2 faire

Enfants[1][i] ← SP1[i]

Enfants[2][i] ← SP2[i]

FinPour

Pour i = Point2 + 1 à n faire

Enfants[2][i] ← SP1[i]

Enfants[1][i] ← SP2[i]

FinPour

Retourner Enfants[][] ;

Fin.

7. Il est clair que la fonction qui implémente le croisement à deux points crée parfois des solutions irréalisables. Ecrire une fonction qui permet de tester la réalisabilité d'une solution ***sol*** passée en paramètre et si ***sol*** n'est pas réalisable, alors la fonction modifie ***sol*** pour la rendre réalisable sinon la fonction retourne ***sol*** sans modification? (4 pts)

Fonction correctionSol (Sol[] : tableau d'entiers, n : entier) : tableau d'entiers

Variable

Cpt, position : entier ;

Debut

Pour i = 1 à n faire

Si(Sol[i] = 1) alors

Cpt ← Cpt + 1 ;

FinSi

FinPour

Si(Cpt > n/2) alors

Tantque (Cpt > n/2) faire

position ← EntierAleatoire(1, n) ;

Si(Sol[position] = 1) Alors

Cpt ← Cpt - 1 ; Sol[Position] ← 0 ;

FinSi

FinTantque

FinSi

Si(Cpt < n/2) alors

Tantque (Cpt < n/2) faire

position ← EntierAleatoire(1, n) ;

Si(Sol[position] = 0) Alors

Cpt ← Cpt + 1 ; Sol[Position] ← 1 ;

FinSi

FinTantque

FinSi

Retourner Sol[] ;

Fin.

8. Ecrire une fonction qui implémente le croisement uniforme entre deux parents PS1 et PS2 et qui retourne toujours une solution réalisable (sans aucune intervention d'une fonction de correction). (4 pts)

Fonction croisementUniforme (Sp1[], Sp2[] : tableau d'entiers, n : entier) : tableau d'entiers

Variable

r : réel; nbrElements[] : tableau d'entiers; nbrOblg : entier;

enfant[] : tableau d'entiers;

Debut

Si (Sp1[n] == 1 ou Sp2[n] == 1) alors

 nbrElements[n] ← 1;

Sinon

 nbrElements[n] ← 0;

FinSi

Pour i = n - 1 à 1 faire

 nbrElements[i] ← nbrElements[i+1];

 Si (Sp1[i] == 1 ou Sp2[i] == 1) alors

 nbrElements[i] ← nbrElements[i] + 1;

 FinSi

FinPour

nbrOblg ← n/2;

pour i = 1 à n faire

 Si nbrOblg > 0 faire

 Si (nbrElements[i] > nbrOblg) alors

 r ← réelAléatoire(0, 1);

 Si (r > 0.5) alors

 enfant[i] ← Sp1[i];

 Sinon

 enfant[i] ← Sp2[i];

 Finsi

 Sinon

 Si (Sp1[i] == 1 ou Sp2[i] == 1) alors

 enfant[i] ← 1;

 Sinon

 enfant[i] ← 0;

 FinSi

 FinSi

 nbrOblg ← nbrOblg - enfant[i];

FinSi

FinPour

Retourner enfant[];

Fin.