SRI2A
IA - Représentation et
Résolution de problèmes

Avril 2023 – 1h30
(documents autorisés)

Pour être comptabilisée, toute réponse devra être justifiée.

Exercice 1 (4 pts)

1. Exprimez sous la forme d'une base de formules en logique propositionnelle les connaissances suivantes :
   — Un individu A a l'articulation du genou abîmée et, quand on a l'articulation abîmée, guérir nécessite une prothèse.
   — La pose d'une prothèse nécessite une opération.
   — Il existe deux risques liés à une opération : les risques dus à l'anesthésie et les risques d'infections post-opératoires.
2. Exprimez le problème *Etant données les connaissances précédentes et sachant que l'individu A ne veut prendre aucun risque, A va-t-il guérir ?*
3. Quelle méthode et quel outil pouvez-vous utiliser pour résoudre ce problème ?
4. Donnez la réponse au problème en la justifiant.

Exercice 2 (4 pts)

On considère une rangée de n cases, dont une seule est vide (dans la suite de ce sujet, une case vide sera dénotée par le symbole $\_$). Chaque case non vide comporte soit un jeton marqué d'une croix $(+)$, soit un jeton marqué d'un rond $(\circ)$.
On peut déplacer les jetons selon les règles suivantes :
— Les jetons **marqués d'une croix** ne peuvent être déplacés **que vers la droite** :
   — soit d'une case si la case de droite est vide (mouvement $+ >$),
   — soit de deux cases si la case de droite est marquée par un rond et que la suivante toujours à droite est vide (mouvement $+\circ >$).
— Les jetons **marqués d'un rond** ne peuvent être déplacés **que vers la gauche** :
   — soit d'une case si la case de gauche est vide (mouvement $< \circ$),
   — soit de deux cases si la case de gauche est marquée par une croix et que la suivante toujours à gauche est vide (mouvement $< +\circ$).

Exemple : pour $n = 3$, en partant de $(+, \_, \circ)$ par le mouvement $+ >$, on obtient $(\_, +, \circ)$. Puis par le mouvement $< +\circ$, on obtient $(\circ, +, \_)$. Et enfin par le mouvement $+ >$, on obtient $(\circ, \_, +)$.
Un exemple de problème est le suivant :

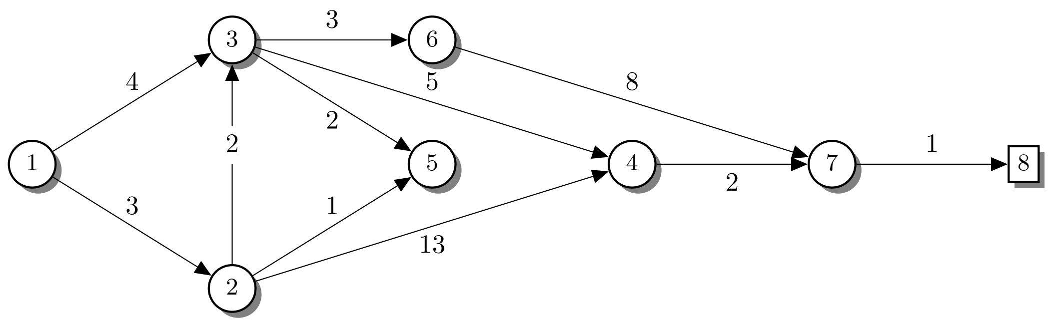
> Soit une rangée de 5 cases avec la configuration initiale $(+, +, \_, \circ, \circ)$.
> On cherche à atteindre la configuration $(\circ, \circ, \_, +, +)$.

Vous répondrez aux questions suivantes :

1. Donner la définition formelle des états et des opérateurs pour un n quelconque.
2. Donner au moins deux exemples de ce type de problème pour lesquels, dans l'espace de recherche défini dans la question précédente, il n'existe pas de solution.

Exercice 3 (4 pts)

On considère l'espace d'états donné par la figure suivante.



Le graphe est orienté, les valeurs sur les arcs donnent le coût de passage d'un état à un autre. L'état initial est 1. On considère qu'on dispose d'une fonction qui reconnaît les états finaux (indiqués par un rectangle sur la figure).

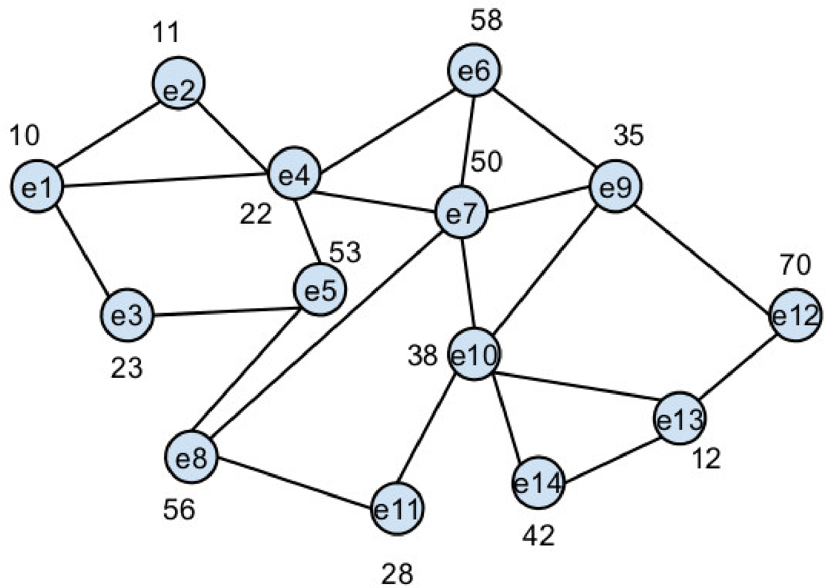
1. Décrivez l'application du A* sur cet espace en utilisant l'heuristique h suivante :

s	1	2	3	4	5	6	7	8
$h(s)$	0	11	8	8	10	9	1	0

   Vous donnerez le chemin solution et son poids.
2. h est-elle admissible ? Qu'en concluez-vous sur la réponse obtenue à la question 1 ?

Exercice 4 (4 pts)

On considère le graphe suivant :



Chaque nœud correspond à une solution. Le voisinage d'une solution e est constitué de l'ensemble des solutions e' qui ont une arête allant de e à e' (le graphe est non-orienté). La valeur de chaque solution est indiquée à côté du nœud. On part d'une solution donnée et on cherche la solution dont la valeur est **la plus grande**.

1. Donnez l'ordre des solutions visitées en utilisant le *steepest hill climbing*, sans reprise, en partant de e_1.
2. Décrivez l'exécution de *tabou* (solutions visitées + liste de tabous à chaque étape) avec une taille de liste égale à 3 en partant de e_1. Arrêtez-vous quand l'algorithme boucle (c'est-à-dire quand on retrouve le même état courant avec la même liste de tabous que lors d'une étape précédente) ou que la liste de tabous atteint la taille de 3.

Exercice 5 (4 pts)

Les étudiants de SRI2A doivent bientôt partir en stage. 3 étudiants (E_1, E_2 et E_3) n'ont pas encore choisi leur stage et il ne reste plus que 5 sujets de stage disponibles sachant que la thématique et la localisation de chaque sujet sont données dans le tableau suivant :

Numéro sujet	thématique	localisation
stage 1	programmation en Java	entreprise sur Toulouse
stage 2	robotique	laboratoire à Londres
stage 3	programmation python	entreprise à Madrid
stage 4	vision	entreprise à Londres
stage 5	automatique	laboratoire à Tokyo

Pour simplifier le problème, on considérera que chaque localisation peut accueillir plusieurs stagiaires à la fois (dans le pire des cas, les 3 étudiants seront au même endroit).
Pour attribuer les sujets aux étudiants, l'équipe pédagogique doit aussi tenir compte des préférences des étudiants :

— E_1 et E_2 veulent faire le stage à l'étranger.
— E_2 déteste l'automatique et la robotique.
— E_3 voudrait faire de la programmation (peu importe le langage).
— E_1 veut travailler sur **exactement** la même thématique de E_3
— E_2 et E_3 ne s'entendent pas du tout et ne veulent pas risquer de se retrouver au même endroit.

1. De quel type de problème s'agit-il ?
2. Donner une représentation de ce problème.
3. Quel algorithme pouvez-vous utiliser pour résoudre ce problème ?
4. Proposer une solution en la justifiant.
5. Que se passe-t-il si on supprime l'hypothèse simplificatrice en considérant désormais qu'il n'y a qu'une seule place disponible dans chaque localisation ?

Corrigé

Exercice 1 (4 pts)

1. Exprimez sous la forme d'une base de formules en logique propositionnelle les connaissances suivantes :
   — Un individu A a l'articulation du genou abîmée et, quand on a l'articulation abîmée, guérir nécessite une prothèse.
   — La pose d'une prothèse nécessite une opération.
   — Il existe deux risques liés à une opération : les risques dus à l'anesthésie et les risques d'infections post-opératoires.

   Le vocabulaire :
   — *GA articulation du genou abîmée,*
   — *G guéri,*
   — *P prothèse,*
   — *O opération,*
   — *RA risques liés à l'anesthésie,*
   — *RI risques d'infections post-opératoires,*

   Les formules :
   — *Un individu A a l'articulation du genou abîmée et, quand on a l'articulation abîmée, guérir nécessite une prothèse.*
   $$GA \wedge (GA \rightarrow (G \rightarrow P))$$
   — *La pose d'une prothèse nécessite une opération.*
   $$P \rightarrow O$$
   — *Il existe deux risques liés à une opération : les risques dus à l'anesthésie et les risques d'infections post-opératoires.*
   $$O \rightarrow (RA \wedge RI)$$

2. Exprimez le problème *Etant données les connaissances précédentes et sachant que l'individu A ne veut prendre aucun risque, A va-t-il guérir ?*

   On commence par traduire "l'individu A ne veut prendre aucun risque" : $\neg RA \wedge \neg RI$.
   Puis on se pose la question suivante : $BC \cup \{\neg RA \wedge \neg RI\} \models G$?

3. Quelle méthode et quel outil pouvez-vous utiliser pour résoudre ce problème ?

   On peut transformer ce pb en un pb de satisfiabilité et utiliser un solver SAT :
   $$BC \cup \{\neg RA \wedge \neg RI, \neg G\} \text{ est-elle satisfiable ?}$$
   Si oui, alors la réponse au pb initial est NON (A ne guérira pas forcément).
   Si non, alors la réponse au pb initial est OUI (A guérira).

4. Donnez la réponse au problème en la justifiant.

   Il suffit d'exhiber un modèle : on met toutes les variables propositionelles à faux sauf la variable GA. Cela correspond au cas où A a l'articulation abîmée mais ne se fait pas opérer. Cette interprétation est un modèle de $BC \cup \{\neg RA \wedge \neg RI, \neg G\}$ et donc A ne guérira pas.

Exercice 2 (4 pts)

On considère une rangée de n cases, dont une seule est vide (dans la suite de ce sujet, une case vide sera dénotée par le symbole _). Chaque case non vide comporte soit un jeton marqué d'une croix (+), soit un jeton marqué d'un rond (◦).

On peut déplacer les jetons selon les règles suivantes :
— Les jetons **marqués d'une croix** ne peuvent être déplacés **que vers la droite** :
   — soit d'une case si la case de droite est vide (mouvement + >),

    — soit de deux cases si la case de droite est marquée par un rond et que la suivante toujours à droite est vide (mouvement $+\circ >$).
- Les jetons **marqués d'un rond** ne peuvent être déplacés **que vers la gauche** :
    — soit d'une case si la case de gauche est vide (mouvement $< \circ$),
    — soit de deux cases si la case de gauche est marquée par une croix et que la suivante toujours à gauche est vide (mouvement $< +\circ$).

Exemple : pour $n = 3$, en partant de $(+, \_, \circ)$ par le mouvement $+ >$, on obtient $(\_, +, \circ)$. Puis par le mouvement $< +\circ$, on obtient $(\circ, +, \_)$. Et enfin par le mouvement $+ >$, on obtient $(\circ, \_, +)$.

Un exemple de problème est le suivant :

> Soit une rangée de 5 cases avec la configuration initiale $(+, +, \_, \circ, \circ)$.
> On cherche à atteindre la configuration $(\circ, \circ, \_, +, +)$.

Vous répondrez aux questions suivantes :

1. Donner la définition formelle des états et des opérateurs pour un n quelconque.

Un état sera un n-uplet $(v_1, \ldots, v_n)$ donnant la valeur de chacune des cases (ces valeurs v_i étant prises dans l'ensemble $\{\circ, +, \_\}$) avec la contrainte suivante : $\exists! i \in [1, n]$ tel que $v_i = \_$.
Dans le cadre du pb décrit, $n = 5$. L'état initial est $(+, +, \_, \circ, \circ)$ et l'état final est $(\circ, \circ, \_, +, +)$.

Les opérateurs vont correspondre aux mouvements possibles. Il y en a 4 et ils ne sont pas toujours applicables (ils ont donc des préconditions). Soit $e = (v_1, \ldots, v_n)$:
- *si $\exists i, 1 \le i < n$ tel que $v_i = +$ et $v_{i+1} = \_$ alors $+ > (e) = e'$ avec e' un état tel que $e' = (v'_1, \ldots, v'_n)$ avec $\forall j \in [1, i-1] \cup [i+2, n], v'_j = v_j$, $v'_i = \_$ et $v'_{i+1} = +$;*
- *si $\exists i, 1 \le i < n-1$ tel que $v_i = +$, $v_{i+1} = \circ$ et $v_{i+2} = \_$ alors $+\circ > (e) = e'$ avec e' un état tel que $e' = (v'_1, \ldots, v'_n)$ avec $\forall j \in [1, i-1] \cup [i+3, n], v'_j = v_j$, $v'_i = \circ$, $v'_{i+1} = \_$ et $v'_{i+2} = +$;*
- *si $\exists i, 1 < i \le n$ tel que $v_i = \circ$ et $v_{i-1} = \_$ alors $< \circ(e) = e'$ avec e' un état tel que $e' = (v'_1, \ldots, v'_n)$ avec $\forall j \in [1, i-2] \cup [i+1, n], v'_j = v_j$, $v'_i = \_$ et $v'_{i-1} = \circ$;*
- *si $\exists i, 2 < i \le n$ tel que $v_i = \circ$, $v_{i-1} = +$ et $v_{i-2} = \_$ alors $< + \circ (e) = e'$ avec e' un état tel que $e' = (v'_1, \ldots, v'_n)$ avec $\forall j \in [1, i-3] \cup [i+1, n], v'_j = v_j$, $v'_i = +$, $v'_{i-1} = \_$ et $v'_{i-2} = \circ$.*

Remarque : lors de l'application d'un mouvement on ne peut obtenir au plus qu'un seul autre état e' puisqu'il n'existe qu'une seule case vide dans l'état e.

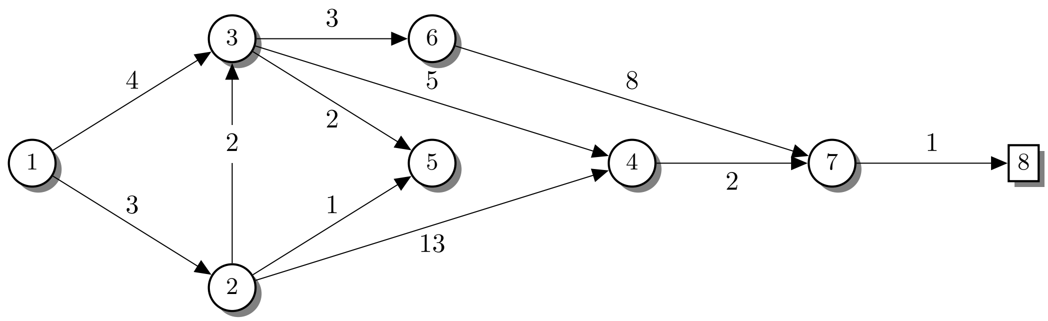
2. Donner au moins deux exemples de ce type de problème pour lesquels, dans l'espace de recherche défini dans la question précédente, il n'existe pas de solution.

Il suffit que la case vide soit mal placée et que l'état final soit différent de l'état initial. Dans ce cas, aucun mouvement ne peut s'appliquer et on ne pourra donc pas trouver une solution. Exemple avec $n = 5$: configuration initiale $(\_, +, +, \circ, \circ)$ et configuration finale $(\_, \circ, \circ, +, +)$. Ou bien $(+, +, \circ, \circ, \_)$ et configuration finale $(\circ, \circ, +, +, \_)$

Autre cas de pb insoluble : il existe au moins un type de valeur pour lequel son nb d'occurrences dans l'état initial diffère de celui dans l'état final. En effet, aucun des mouvements décrits ne permet d'introduire une nouvelle valeur. Exemple pour $n = 5$: configuration initiale $(+, +, \_, \circ, \circ)$ et configuration finale $(\circ, +, \_, +, +)$

Exercice 3 (4 pts)

On considère l'espace d'états donné par la figure suivante.



Le graphe est orienté, les valeurs sur les arcs donnent le coût de passage d'un état à un autre. L'état initial est 1. On considère qu'on dispose d'une fonction qui reconnaît les états finaux (indiqués par un rectangle sur la figure).

1. Décrivez l'application du A* sur cet espace en utilisant l'heuristique h suivante :

s	1	2	3	4	5	6	7	8
$h(s)$	0	11	8	8	10	9	1	0

Vous donnerez le chemin solution et son poids.

Dans le tableau ci-après, on mettra pour chaque état généré un triplet contenant le numéro de l'état, le numéro de son père, et l'expression correspondant à la fonction f (donc le coût + l'heuristique).

AVoir	**Vus**	s
$(1, -, 0+0)$	$\varnothing$	
$\varnothing$	$(1, -, 0+0)$	$(1, -, 0+0)$
$(2, 1, 3+11)$ $(3, 1, 4+8)$	$(1, -, 0+0)$	$(1, -, 0+0)$
$(2, 1, 3+11)$	$(1, -, 0+0)$ $(3, 1, 4+8)$	$(3, 1, 4+8)$
$(2, 1, 3+11)$ $(4, 3, 9+8)$ $(5, 3, 6+10)$ $(6, 3, 7+9)$	$(1, -, 0+0)$ $(3, 1, 4+8)$	$(3, 1, 4+8)$
$(4, 3, 9+8)$ $(5, 3, 6+10)$ $(6, 3, 7+9)$	$(1, -, 0+0)$ $(3, 1, 4+8)$ $(2, 1, 3+11)$	$(2, 1, 3+11)$
$(4, 3, 9+8)$ $(5, 2, 4+10)$ $(6, 3, 7+9)$	$(1, -, 0+0)$ $(3, 1, 4+8)$ $(2, 1, 3+11)$	$(2, 1, 3+11)$
$(4, 3, 9+8)$ $(6, 3, 7+9)$	$(1, -, 0+0)$ $(3, 1, 4+8)$ $(2, 1, 3+11)$ $(5, 2, 4+10)$	$(5, 2, 4+10)$
$(4, 3, 9+8)$ $(6, 3, 7+9)$	$(1, -, 0+0)$ $(3, 1, 4+8)$ $(2, 1, 3+11)$ $(5, 2, 4+10)$	$(5, 2, 4+10)$
$(4, 3, 9+8)$	$(1, -, 0+0)$ $(3, 1, 4+8)$ $(2, 1, 3+11)$ $(5, 2, 4+10)$ $(6, 3, 7+9)$	$(6, 3, 7+9)$
$(4, 3, 9+8)$ $(7, 6, 15+1)$	$(1, -, 0+0)$ $(3, 1, 4+8)$ $(2, 1, 3+11)$ $(5, 2, 4+10)$ $(6, 3, 7+9)$	$(6, 3, 7+9)$
$(4, 3, 9+8)$	$(1, -, 0+0)$ $(3, 1, 4+8)$ $(2, 1, 3+11)$ $(5, 2, 4+10)$ $(6, 3, 7+9)$ $(7, 6, 15+1)$	$(7, 6, 15+1)$
$(4, 3, 9+8)$ $(8, 7, 16+0)$	$(1, -, 0+0)$ $(3, 1, 4+8)$ $(2, 1, 3+11)$ $(5, 2, 4+10)$ $(6, 3, 7+9)$ $(7, 6, 15+1)$	$(7, 6, 15+1)$
$(4, 3, 9+8)$	$(1, -, 0+0)$ $(3, 1, 4+8)$ $(2, 1, 3+11)$ $(5, 2, 4+10)$ $(6, 3, 7+9)$ $(7, 6, 15+1)$ $(8, 7, 16+0)$	$(8, 7, 16+0)$

Et c'est fini car 8 est un état solution.

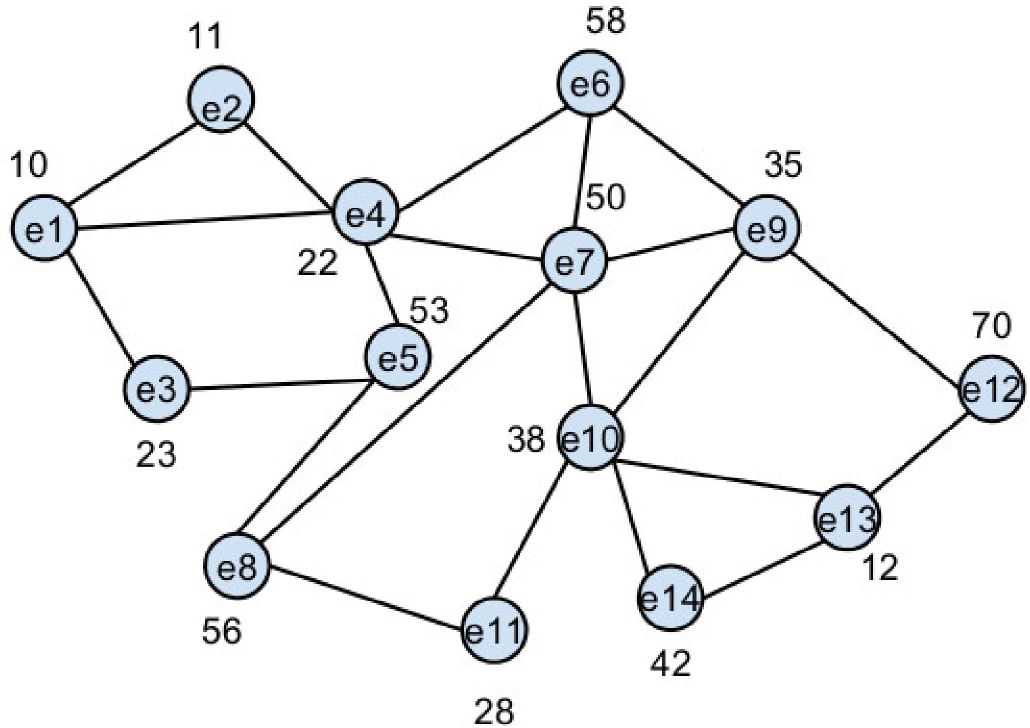
Le chemin solution est donc : $1-3-6-7-8$ *de poids 16.*

2. h est-elle admissible ? Qu'en concluez-vous sur la réponse obtenue à la question 1 ?

h n'est pas admissible car elle est non montone. Par exemple pour le sommet 4, on a $h(4) = 8 \not\leq h(7) + 2 = 3$.
On n'a donc pas de garantie d'avoir trouvé le meilleur chemin. Et effectivement il en existe un autre qui a un meilleur score : $1 - 3 - 4 - 7 - 8$ de poids 12.

Exercice 4 (4 pts)

On considère le graphe suivant :



Chaque nœud correspond à une solution. Le voisinage d'une solution e est constitué de l'ensemble des solutions e' qui ont une arête allant de e à e' (le graphe est non-orienté). La valeur de chaque solution est indiquée à côté du nœud. On part d'une solution donnée et on cherche la solution dont la valeur est **la plus grande**.

1. Donnez l'ordre des solutions visitées en utilisant le *steepest hill climbing*, sans reprise, en partant de e_1.

   On part de e_1 qui vaut 10. On passe à e_3 qui vaut 23, puis e_5 (53), puis e_8 (56). Et arrêt car les fils de e_8 sont tous de coût inférieur.
   La meilleure valeur trouvée est e_8 (56).

2. Décrivez l'exécution de *tabou* (solutions visitées + liste de tabous à chaque étape) avec une taille de liste égale à 3 en partant de e_1. Arrêtez-vous quand l'algorithme boucle (c'est-à-dire quand on retrouve le même état courant avec la même liste de tabous que lors d'une étape précédente) ou que la liste de tabous atteint la taille de 3.

On part de e_1 (10) avec la liste des tabous $T = \varnothing$.
Puis on passe par e_3 (23), puis e_5 (53), puis e_8 (56) sans toucher à T.
Puis on repart sur e_5 (53) avec $T = \{e_8\}$.
Puis on repart sur e_3 (23) avec $T = \{e_8, e_5\}$.
Puis on repart sur e_1 (10) avec $T = \{e_8, e_5, e_3\}$.
La liste des tabous a atteint la taille de 3. On s'arrête et on n'a pas trouvé de meilleure valeur qu'à la question précédente.
La meilleure valeur trouvée est e_8 (56).

Exercice 5 (4 pts)

Les étudiants de SRI2A doivent bientôt partir en stage. 3 étudiants (E_1, E_2 et E_3) n'ont pas encore choisi leur stage et il ne reste plus que 5 sujets de stage disponibles sachant que la thématique et la localisation de chaque sujet sont données dans le tableau suivant :

Numéro sujet	thématique	localisation
stage 1	programmation en Java	entreprise sur Toulouse
stage 2	robotique	laboratoire à Londres
stage 3	programmation python	entreprise à Madrid
stage 4	vision	entreprise à Londres
stage 5	automatique	laboratoire à Tokyo

Pour simplifier le problème, on considérera que chaque localisation peut accueillir plusieurs stagiaires à la fois (dans le pire des cas, les 3 étudiants seront au même endroit).
Pour attribuer les sujets aux étudiants, l'équipe pédagogique doit aussi tenir compte des préférences des étudiants :
- E_1 et E_2 veulent faire le stage à l'étranger.
- E_2 déteste l'automatique et la robotique.
- E_3 voudrait faire de la programmation (peu importe le langage).
- E_1 veut travailler sur **exactement** la même thématique de E_3
- E_2 et E_3 ne s'entendent pas du tout et ne veulent pas risquer de se retrouver au même endroit.

1. De quel type de problème s'agit-il ?

   C'est un pb de satisfaction de contraintes.

2. Donner une représentation de ce problème.

   Variables *: une pour chaque étudiant E_1, E_2, E_3*

   Domaine des variables *: le même pour toutes les variables, la liste des stages $\{1, 2, 3, 4, 5\}$*

   Contraintes *: (pour prendre en compte les préférences des étudiants)*
   Les trois premières préférences permettent de réduire le domaine des variables :
   $D_1 = D_2 = \{2, 3, 4, 5\}$ *puis* $D_2 = \{3, 4\}$ *puis* $D_3 = \{1, 3\}$
   La quatrième préférence correspond à la contrainte suivante donnant les tuples **autorisés** *:*
   $C_{13} = \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5)\}$
   Et la dernière préférence donne la contrainte suivante donnant les tuples **interdits** *:*
   $C_{23} = \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (2, 4), (4, 2)\}$

3. Quel algorithme pouvez-vous utiliser pour résoudre ce problème ?

   On peut utiliser le backtrack avec ou sans ordonnancement

4. Proposer une solution en la justifiant.

On va utiliser le backtrack avec ordonnancement en respectant l'ordre suivant (obtenu en fonction du nb de contraintes ds lesquelles les variables sont impliquées, de la taille des domaines puis de l'ordre alphabétique) : E_2, puis E_3 puis E_1.
Pour E_2 : on choisit la valeur 3.
Pour E_3 : on choisit la valeur 1
Pour E_1 : échec car 1 n'est plus autorisée pour E_1 et 2 viole la contrainte C_{13}. Donc **backtrack**.
Pour E_3 : on choisit la valeur 3. La contrainte C_{23} est alors violée. Donc échec et **backtrack**.
Pour E_2 : on choisit la valeur 4.
Pour E_3 : on choisit la valeur 1
Pour E_1 : échec car 1 n'est plus autorisée pour E_1 et 2 viole la contrainte C_{13}. Donc **backtrack**.
Pour E_3 : on choisit la valeur 3.
Pour E_1 : on choisit la valeur 3. Et c'est fini.
Assignation finale : $\{(E_1, 3), (E_2, 4), (E_3, 3)\}$.

Une autre résolution en utilisant la propagation de valeur serait la suivante :
A cause de C_{13}, le domaine de E_1 devient $\{2, 3, 4, 5\} \cap \{1, 3\} = \{3\}$. Donc E_1 est la variable la plus contrainte. On l'affecte donc avec 3.
Par propagation et C_{13}, le domaine de E_3 devient $\{3\}$ et donc E_3 est affectée avec 3.
Par propagation et C_{23}, le domaine de E_2 devient $\{4\}$ et donc E_2 est affectée avec 4.
Et on retrouve l'assignation finale : $\{(E_1, 3), (E_2, 4), (E_3, 3)\}$.

5. Que se passe-t-il si on supprime l'hypothèse simplificatrice en considérant désormais qu'il n'y a qu'une seule place disponible dans chaque localisation ?

Cela rajoute une contrainte donnant les tuples **autorisés** *et définie en intension de la manière suivante :*

$$C_{123} = \{(i, j, k) \text{ tel que } i, j, k \in \{1, 2, 3, 4, 5\} \text{ et } i \neq j \neq k \neq i\}$$

On se retrouve alors avec un problème insoluble à cause de la contrainte C_{13} qui est en contraction avec C_{123}.