Examen Programmation par contraintes IA302

5 novembre 2019

Abstract

Examen sous forme de mini projets. Le sujet comporte 3 exercices. Le rendu se fera sous forme d'un email (alexandre@ensta.fr) avec en pièce jointe une archive (nom_prenom.zip ou .tar) contenant vos programmes ainsi qu'un fichier "reponses.txt" contenant les réponses aux questions.

1 Problème 1 : Enigma

La machine Enigma est à la source de nombreuses avancées en mathématiques et en informatique. On se propose ici de casser une version simplifiée d'Enigma en utilisant la programmation par contraintes. Enigma, dans une version simple voir Figure 1, se compose de trois rotors et d'un réflecteur. Chaque rotor est une table de transposition (qui change une lettre en une autre) et qui tourne à un rythme particulier. Le premier rotor dit "rapide" tourne d'un cran (donc un décalage dans la table de transposition) à chaque fois qu'une lettre est transposée. Le deuxième dit "moyen" tourne d'un cran lorsque le rapide a fait un tour complet (c'est à dire 26 crans). Enfin le troisième dit "lent" tourne d'un cran lorsque le moyen a fait un tour. Ainsi l'encodage d'une même lettre n'est pas le même à chaque fois, cela rend difficile le cassage du code Enigma. Le réflecteur est également une table de transposition, mais elle est statique et symmétrique. Lorsqu'une lettre est tapée sur le clavier, elle traverse les trois rotors en avant puis elle est transposée par le réflecteur et elle repasse dans les trois rotors au retour. Casser le code Enigma, revient à trouver la position initiale des trois rotors, c'est la clé (trois entiers entre 0 et 25 représentant le décalage des rotors par rapport à la position nulle).

Dans cet exercice, vous allez casser ce code ou plutôt une petite partie, mais le principe est le même.

Pour cela un fichier en Python enigma.py vous est fourni. Il contient les fonctions d'encodage et les paramètres de notre Enigma. Il simule le fait que nous ayons en notre possession une machine Enigma pour essayer de casser le code.

1.1 Question 1 : un seul rotor

Dans un premier temps, les espions adverses n'utilisent que le premier rotor d'Enigma, le"rapide" (sans réflecteur). Nos services secrets ont intercepté un message "BSHCIECWNHN" et également une information importante : on sait que ce message est en anglais et qu'il commence par une formule de politesse comme "HELLO" ou "HI".

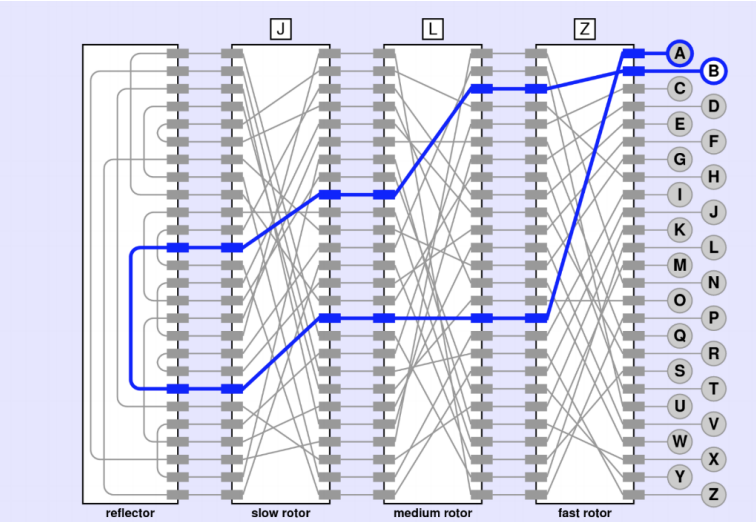

Figure 1: Machine Enigma

1. Ecrivez, dans le fichier texte, un CSP décrivant le problème consistant à trouver la position initiale du premier rotor (le début de la clé) en fonction des informations obtenues par nos services secrets.

2. Ecrivez un algorithme simple pour résoudre ce CSP dans le fichier fourni et donc trouver la clé.

3. Décryptez le message intercepté avec la clé obtenue.

1.2 Question 2 : deux rotors

Nos ennemis se sont rendu compte que le code était facile à casser avec un seul rotor, ils ajoutent donc un rotor “moyen” (sans réflecteur). Nos services ont intercepté un nouveau message “YELUKSEWOANLGWIFECOHXAHVZ“. Ils savent aussi que ce message est en anglais et commence par ”THE“, ”THEY“ ou ”THIS“.

1. Ecrivez, dans le fichier texte, un CSP décrivant le problème consistant à trouver la position initiale des premiers rotors en fonction des informations obtenues par nos services secrets.

2. Ecrivez un algorithme simple pour résoudre ce CSP dans le fichier Python fourni et donc trouver la clé (il peut y avoir plusieurs solutions).

3. Décryptez le message intercepté avec la clé obtenue (trouvez celle parmi les solutions qui décrypte correctement le message).

2 Problème 2 : mots croisés

Nous souhaitons placer six mots de trois lettres de la langue française (des mots de trois lettres vous sont données dans le fichier mots.txt) dans une grille 3x3 comme ci dessous :

Table 1: Mots croisés

	C1	C2	C3
L1	A		
L2	R	U	E
L3	C		

De plus, seule la première lettre de L1 et de C1 doit être commune (la lettre en L2/C1 est différente de celle en L1/C2, idem pour L3/C1 et L1/C3).

1. Ecrivez, dans le fichier texte, un CSP décrivant le problème.
2. En utilisant l'outil Constraints vu en TP implémentez et résolvez ce CSP (le fichier xml créé par l'outil lors d'une sauvegarde est à rendre).

3 Problème 3 : preuve de non existence

Prouvez que le polynome suivant n'accepte pas de zéro dans l'intervalle $[-1.95, 0]$

$$p(x) = x^6 + 4.2x^3 - 3.72x^2 + 5.12x - 0.5$$

1. Ecrivez, dans le fichier texte, un CSP décrivant le problème.
2. En utilisant Ibex, résolvez ce CSP et montrez qu'il n'accepte pas de solutions.