



ÉCOLE POLYTECHNIQUE DE L'UNIVERSITÉ DE TOURS  
64, Avenue Jean Portalis  
37200 TOURS, FRANCE  
Tél. (33)2-47-36-14-14  
Fax (33)2-47-36-14-22  
[www.polytech.univ-tours.fr](http://www.polytech.univ-tours.fr)



# DI3

## Rapport de projet S4

### Projet tutoré 2: Sac à dos

Auteur(s)

**Thomas Couchoud**

[\[thomas.couchoud@etu.univ-tours.fr\]](mailto:thomas.couchoud@etu.univ-tours.fr)

**Victor Coleau**

[\[victor.coleau@etu.univ-tours.fr\]](mailto:victor.coleau@etu.univ-tours.fr)

Encadrant(s)

**Yannick Kergosien**

[\[yannick.kergosien@univ-tours.fr\]](mailto:yannick.kergosien@univ-tours.fr)

**Polytech Tours**

**Département DI**

# Table des matières

---

<b>1</b>	<b>Structure du projet</b>	<b>2</b>
1.1	Fichiers . . . . .	2
1.2	Structures . . . . .	3
1.2.1	Parser . . . . .	3
<b>2</b>	<b>Parser</b>	<b>4</b>
<b>3</b>	<b>Heuristique</b>	<b>5</b>
<b>4</b>	<b>Metaheuristiques</b>	<b>6</b>
4.1	Local . . . . .	6
4.2	Tabou . . . . .	6
4.3	Genetique . . . . .	6
<b>5</b>	<b>Autres</b>	<b>7</b>
	<b>Conclusion</b>	<b>8</b>
	<b>Annexes</b>	<b>8</b>

# Introduction

---

Le projet que nous avons choisi est celui portant sur le problème du sac à dos (*Knapsack problem*). Celui-ci se généralise très simplement et donne lieu à de nombreux problèmes analogues.

Dans notre cas nous devons remplir un sac à dos d'objets. Chaque objet a une certaine valeur près définie ainsi que des "poids" sur différentes dimensions. On peut imaginer le cas où l'on tente de remplir sa valise pour partir en voyage. Chaque objet a une valeur selon l'importance qu'on lui donne ainsi que des "poids" qui pourraient être la place qu'il occupe, son poids etc. L'idée ici est d'essayer de maximiser la valeur que nous emportons avec nous sachant que notre valise est limitée en poids et taille.

D'un point de vue mathématique, on peut modéliser ceci simplement :

- $X$  Un vecteur définissant quels items sont dans le sac ou pas (ex :  $(0; 0; 1)$  définira un sac avec seulement le 3ème item de pris).
- $W_j$  Le poids maximum que le sac peut contenir sur la dimension  $j$ .
- $w_{i,j}$  Le poids du  $i$ ème item sur la  $j$ ème dimension.
- $v_i$  La valeur du  $i$ ème item.

Les contraintes sont :  $\forall j \in [0...m], \sum_{i=0}^n x_i w_{i,j} \leq W_j$

On appellera la fonction objectif  $z(X)$  la fonction donnant la valeur d'un sac :  $z(X) = \sum_{i=0}^m x_i v_i$

# Chapitre 1

## Structure du projet

---

Le C ayant un langage rassemblant tout les fichiers en un lors de la compilation, il est nécessaire de choisis judicieusement ses noms de fonctions afin d'éviter les duplicatas. Dans notre cas nous avons choisi un formatage simple : [Nom du .c / Nom de la structure]\_[Nom de la fonction]. Nous aurons donc des fonctions du type *population\_create(...)* ou bien *metaheuristicGenetic\_search(...)*.

### 1.1 Fichiers

Concernant l'organisation des fichiers en eux même, chaque type d'entre eux est localisé à un endroit différent. En effet nous avons commencé par mettre tout nos .c et .h dans un même dossier. Cependant le projet grandissant assez vite, il a rapidement été un lieux où l'on se perd. Pour cela nous avons décidé de séparer les .h des .c puisque nous travaillons principalement sur les .c. Cela permet de s'y retrouver plus aisément. Ainsi la structure de notre dossier sources est la suivante :

- src → Le dossier racine contenant nos .c pour le programme
- headers → Le dossier contenant nos headers pour le programme
- unit → Le dossier contenant nos .c pour les tests unitaires
- headers → Le dossier contenant nos headers pour les tests unitaires

Intéressons nous au dossier src. Nous avons décider de créer des fichiers spécifiques pour certaines structures ainsi que leur fonctions. Celles-ci sont :

- Bag → Représentant le contenu du sac pour une solution directe.
- SolutionDirect → Représentant une solution directe.
- SolutionIndirect → Représentant une solution indirecte.
- Solution → Représentant l'union d'une solution directe et indirecte.
- Instance → Représentant une instance.
- Item → Représenant un élément du sac.

A coté de cela nous avons plusieurs fichiers étant juste un regroupement de fonctions selon leur utilisation :

- Parser → Regroupant les différentes fonctions afin de lire un fichier.
- Heuristic → Regroupant les fonctions liées à la résolution grâce à une heuristique.
- Scheduler → Regroupant les fonctions liées aux différents algorithmes pour l'heuristique.
- MetaheuristiqueLocal → Regroupant les fonctions liées aux différents algorithmes pour la metaheuristique locale.
- MetaheuristiqueTabou → Regroupant les fonctions liées aux différents algorithmes pour la metaheuristique tabou et contenant la structure Tabou.

- MetaheuristiqueGenetic → Regroupant les fonctions liées aux différents algorithmes pour la metaheuristique génétique et contenant la structure Population.

## 1.2 Structures

### 1.2.1 Parser

Cette structure permet la lecture pas à pas d'un fichier. Celle-ci contient le chemin du fichier concerné, l'offset actuel de lecture, le nombre d'instances à lire et le nombre d'instances lues.

Les fonctions suivantes lui sont associées :

- parser\_create → Permet de créer cette structure à partir du chemin d'un fichier.
- parser\_destroy → Pour détruire la structure.
- parser\_getNextInstance → Renvoi la prochaine instance du fichier ou NULL si l'on a atteint la fin.

## Chapitre 2

# Parser

---

L'une des premières parties que nous devons réaliser est le parser. Lors de cette dernière, un choix important a dû se faire : lisons nous toutes les instances d'un fichier en un seul coup ou lisons les une par une ?

Nous avons premièrement décidé de les lire toutes à la suite. En effet ce choix était celui de la simplicité. Nous avons voulu commencer simple afin de pouvoir s'avancer sur les autres tâches à faire. Puis rapidement nous avons implémenté la seconde méthode. Celle-ci nous a paru plus adéquate pour nos utilisations car elle permet d'éviter une utilisation importante de la mémoire pour pas grand chose. Certes nous avons dû créer une structure Parser qui sert principalement à conserver les informations de la dernière lecture, mais ce choix nous a paru être le meilleur.

Ainsi vous trouverez dans notre code les deux méthodes de lecture :

- `parser_readAllFile` → qui renvoi toutes les instances d'un fichier
- `parser_create` → Permet l'utilisation de `parser_getNextInstance` qui renvoi uniquement la prochaine Instance.

## Chapitre 3

# Heuristique

---

Dans le cadre des heuristiques, nous avons du implémenter nos propre critère de selection. Nous allons ici vous en présenter deux.

Le premier se base sur l'algorithme de la dimension critique mais prend cette fois-ci en compte toutes les dimensions. Pour cela nous calculons pour l'item à l'index  $i$  un ratio qui est  $r_i = \sum_{j=0}^m \frac{w_j}{W_j}$ . Ce ratio sert par la suite à calculer un score temporaire afin d'appliquer l'heuristique  $score_i = \frac{v_i}{r_i}$ . De cette manière plus l'item remplira le sac, plus le diviseur sera importante et par conséquent aura un score faible.

Le s

# Chapitre 4

## Metaheuristiques

---

4.1 Local

4.2 Tabou

4.3 Genetique



## Chapitre 5

### **Autres**

---

# Conclusion

---

# Projet tutoré 2: Sac à dos

---

## Rapport de projet S4

**Résumé :** Projet ayant pour objectif la réalisation d'un algorithme cherchant des solutions au problème du sac à dos multidimensionnel.

**Mots clé :** sac à dos, algorithme, C, heuristique, metaheuristique, parser, directe, indirecte

**Abstract :** Project which objective is to find solutions for the multidimensional Knapsack problem.

**Keywords :** backpack, Knapsack, algorithm, C, heuristic, metaheuristic, parser, direct, indirect

Auteur(s)

**Thomas Couchoud**

[[thomas.couchoud@etu.univ-tours.fr](mailto:thomas.couchoud@etu.univ-tours.fr)]

**Victor Coleau**

[[victor.coleau@etu.univ-tours.fr](mailto:victor.coleau@etu.univ-tours.fr)]

Encadrant(s)

**Yannick Kergosien**

[[yannick.kergosien@univ-tours.fr](mailto:yannick.kergosien@univ-tours.fr)]

**Polytech Tours**

**Département DI**

Ce document a été formaté selon le format EPUProjetPeiP.cls (N. Monmarché)

École Polytechnique de l'Université de Tours  
64 Avenue Jean Portalis, 37200 Tours, France  
<http://www.polytech.univ-tours.fr>