

**Des algorithmes administratifs aux algorithmes
publics :
le cas de Admission Post Bac**

Patrick CÉGIELSKI, coordonnateur
cegielski@u-pec.fr

Avril 2018

Legal Notice

Copyright © 2018 Patrick Cégielski
Université Paris XII - IUT Sénart-Fontainebleau
Route forestière Hurtault
F-77300 Fontainebleau
cegielski@u-pec.fr

Préface

Le XXI^e siècle a vu les algorithmes s'introduire dans notre vie quotidienne. Les premiers à connaître la célébrité ont été créés par les grandes sociétés de services technologiques de l'information (le moteur de recherche de *Google* date de 1998, le réseau social *Facebook* a débuté en 2004), mais rapidement, l'Administration s'est révélée être aussi une grande utilisatrice d'algorithmes. Ainsi dans le domaine de l'éducation en particulier, la plateforme AFFELNET pour l'affectation au collège été ouverte en 2008 et l'application *Admission Post-Bac* (APB en abrégé) puis *ParcoursSup*, qui nous servira d'exemple tout au long de cet ouvrage, a été mise en place en 2009. Qu'est-ce que cela a changé pour nous, l'homme de la rue en tant que citoyen ?

L'Administration règle les liens entre l'État et les citoyens, et certains liens entre les citoyens eux-mêmes, non pas de façon arbitraire et dépendant de l'humeur de l'instant, mais d'une façon codifiée par les *procédures administratives* : lorsqu'on veut faire établir un passeport ou une pièce d'identité, on sait à quel endroit on doit s'adresser et on connaît la liste des pièces qu'il faut apporter. L'Administration se devant d'être « humaine », la loi laisse de la latitude pour que l'autorité compétente puisse l'appliquer au mieux dans chaque cas particulier : devant une même contravention au code de la route, un agent de police peut verbaliser ou non suivant qu'il pense que celle-ci aurait pu avoir des conséquences graves ou non.

Les ordinateurs sont arrivés à point nommé pour faciliter le traitement du grand nombre, toujours croissant, d'instances de procédures administratives : les déclarations d'impôts sur le revenu ne se font quasiment plus sous forme « papier », qu'un grand nombre de fonctionnaires devait ne serait-ce que recopier, mais « en ligne ». Les *algorithmes administratifs* mettent alors en place une grande partie des *procédures administratives*, avec toutes les conséquences qui découlent de cette transformation : la latitude, possible avec les procédures administratives traitées directement par les êtres humains représentant l'autorité compétente, ne peut pas se retrouver dans un *algorithme*. La rigueur et le caractère systématique de celui-ci est, au contraire, la condition *sine qua non* pour l'implémenter sur un ordinateur, qui n'a en aucune manière la capacité de juger.

L'introduction des algorithmes administratifs conduit à un renouvellement des réflexions sur

les procédures administratives et leur philosophie. Cet ouvrage voudrait apporter sa contribution à ce renouvellement, en proposant une analyse par un cas concret d’algorithme administratif, à savoir APB.

Cet ouvrage est le rapport final d’un groupe de travail universitaire, ALGOCIT¹, dans le cadre d’un PEPS² du CNRS³ accordé par la Comue UPE⁴ pour les deux années 2015 et 2016. L’équipe pluridisciplinaire était constituée d’enseignants-chercheurs et d’une stagiaire, tous de l’UPEC⁵ : Mathias BÉJEAN, maître de conférences en Gestion à la Faculté de Sciences Économiques et de Gestion, Patrick CÉGIELSKI, professeur (Informatique⁶) à l’IUT⁷ de Sénart-Fontainebleau, Julien CERVELLE, professeur (Informatique) à la Faculté de Sciences et Technologie, Mikhaël COZIC, maître de conférences en Philosophie à la Faculté de Lettres, Langues et Sciences Humaines, Maïté GUILLEMAIN, maître de conférences en Droit privé à l’IUT de Sénart-Fontainebleau, Roman LE ROUX, maître de conférences en sciences de l’information et de la communication à l’ÉSPÉ⁸, Farida SEMMAK, maître de conférences en Informatique à la Faculté de Sciences Économiques et de Gestion, Julien TESSON, maître de conférences en Informatique à la Faculté de Droit, Arnaud THAUVRON, maître de conférences de Gestion à la Faculté de Sciences Économiques et de Gestion, Sylvie THORON, professeur (Économie) à la Faculté d’Administration et Échanges Internationaux, Marion VALARCHER, étudiante de master en sociologie à l’université Paris-Descartes, en stage à l’UPEC, Pierre VALARCHER, professeur (Informatique) à l’IUT de Sénart-Fontainebleau, responsable du projet, et Noé WAGENER, maître de conférences en Droit public à la Faculté de Droit.

Dans une première partie (*Les algorithmes administratifs*), nous donnons, en plaçant le tout dans son contexte, la définition de ce qu’est un algorithme administratif. La seconde partie (*L’affectation*) étudie une sous-classe importante de ce type d’algorithmes administratifs qui sont utilisés pour l’affectation de ressources aux candidats à leur utilisation et, comme exemple concret, l’affectation dans un établissement d’enseignement supérieur. Certains citoyens sont très défiant à l’égard des algorithmes administratifs et veulent absolument comprendre comment ils fonctionnent de façon à être sûrs de ne pas être mal traités, voire floués, par ceux-ci. L’État devrait donc faire preuve de grandes qualités de communication pour répondre à cette attente. Or, pour des raisons diverses, il ne le fait pas systématiquement.

Le cas pris en exemple montre donc comment tirer parti des informations disponibles sur un algorithme (administratif) donné pour en obtenir le plus de renseignements possibles, à défaut de disposer des informations complètes. Un algorithme se présente au citoyen par une *interface*, qui permet à celui-ci d’entrer les informations demandées par l’administration et d’obtenir la réponse de celle-ci. La troisième partie (*Que nous apprend l’interface ?*) montre comment l’interface d’APB, et la documentation de celle-ci, constituée essentiellement par le *guide utilisateur*, permettent d’obtenir des éléments sur celui-ci et analyse les premières conséquences de l’algorithmisation de la procédure par rapport aux procédures antérieures. Cependant, l’interface ne dit pas tout sur un algorithme. La quatrième partie (*Que nous apprend l’algorithme ?*) nous apprend précisément ce qu’il faut entendre par algorithme (chapitre sept), à la fois tel que le fait la dis-

1. Pour « Algorithmes et Citoyenneté ».

2. *Projet Exploratoire Premier Soutien*.

3. *Centre National de la Recherche Scientifique*.

4. Communauté d’Universités et d’Établissements Université Paris - Est.

5. *Université Paris 12 dite Université Paris - Est - Créteil*.

6. Les professeurs des universités sont censés ne pas être spécialisés, afin de pouvoir prendre en compte de façon très réactive les domaines émergents. Ils sont cependant rattachés à une section CNU (*Conseil National des Universités*) ; c’est celle-ci que nous indiquons entre parenthèses.

7. *Institut Universitaire de Technologie*.

8. *École Supérieure du Professorat et de l’Éducation*.

cipline appelée *informatique fondamentale* mais également une certaine dérive des développeurs que nous précisons. Elle étudie ensuite la partie de l'algorithme d'APB rendue publique et montre ce qu'on y apprend de plus, par rapport à l'interface (chapitre huit). Enfin, elle suggère quelques propositions d'amélioration de l'interface d'APB/ParcoursSup pour tenir compte d'une conséquence de l'algorithmisation de la procédure d'affectation dans les établissements d'enseignement supérieurs non visible avec l'interface actuelle. L'algorithme n'est pas suffisant et on doit étudier le code de son implémentation, ce qui fait l'objet de la cinquième partie (*Que nous apprend le code ?*). L'étude d'un sous-programme du code d'APB, communiqué par le ministère, permet de découvrir une autre conséquence, plutôt maladroite d'ailleurs, de l'algorithmisation de la procédure d'affectation dans l'enseignement supérieur.

On voit donc que chaque couche de l'algorithme révélée au public apporte des informations différentes. En conséquence, il faut que chacune de ces couches soit rendue publique si on veut instaurer une confiance dans les algorithmes administratifs. La conclusion donne quelques recommandations sur la façon de rendre publics les algorithmes administratifs, d'où le titre de l'ouvrage.

Dans le premier chapitre (*Les algorithmes et les citoyens*) quelques exemples, devenus classiques, sont donnés parmi ceux qui ont pu conduire à une certaine défiance à l'égard des algorithmes qui « gouverneraient notre vie »⁹.

Le chapitre deux (*L'Administration et ses procédures*) rappelle l'environnement ayant donné naissance à l'administration publique, la nécessité des procédures et la philosophie générale qui la guide. Il pose le **problème fondamental lié aux algorithmes administratifs**, qui doivent concilier, d'une part, la *tradition juridique*, suivant laquelle le législateur, dans sa grande sagesse, donne un cadre général mais laisse à l'autorité compétente le soin d'apprécier à sa juste valeur chaque cas individuel, et, d'autre part, l'*impératif informatique*, où rien ne peut être laissé à l'appréciation de l'ordinateur, qui n'a aucune capacité pour cela.

Le chapitre trois (*L'affectation des ressources*) introduit le problème général de l'affectation d'une ressource dans une Cité, et plus particulièrement celui de l'affectation des lycéens dans l'enseignement supérieur français.

L'affectation nous renvoie à une thématique de l'économie, appelée *théorie du choix social*, dont le but est de définir les préférences collectives d'un groupe (société) à partir des préférences individuelles de chacun de ses membres. Cette thématique, née des réflexions de CONDORCET sur des procédures de vote, a conduit à un traitement formel, surtout depuis les travaux de KENNETH ARROW en 1952. Le chapitre quatre montre comment l'affectation, que les économistes appellent *appariement*, peut être considérée comme une classe particulière de problèmes d'agrégation des préférences individuelles. Il résume les principaux résultats de la littérature qui s'est développée à partir des travaux de deux théoriciens des jeux, GALE et SHAPLEY en 1962. Ces derniers ont proposé en particulier un « bon » algorithme d'appariement qui satisfait des critères de stabilité et d'optimalité.

Le chapitre cinq (*Le cadre législatif de l'affectation dans le supérieur*) fait le point sur l'état du droit concernant la procédure administrative d'affectation dans l'enseignement supérieur, dans une perspective synchronique.

Nous avons déjà commenté le chapitre six (*L'interface d'APB*), chapitre unique de la troisième partie.

Le chapitre sept (*Qu'est-ce qu'un algorithme ?*) commence par donner une idée informelle de ce dont il s'agit. Il note ensuite une dérive de certains développeurs, qui consiste à considérer l'algorithme comme le squelette de leur programme non encore totalement abouti. Le concept d'algorithme n'a reçu de définition formelle que bien tardivement (en 1984 pour être précis), dont

9. Pour reprendre, en l'édulcorant, la une du *Point* n° 2298 du 22 septembre 2016.

nous donnons une idée. On termine par ce que dit le droit positif¹⁰ (tant français que celui de l'Union européenne, mais il en va de même des droits étrangers) sur les algorithmes.

Le chapitre huit (*Une partie de l'algorithme d'APB*) concerne l'analyse de l'algorithme d'affectation utilisé par APB. Il s'agit apparemment d'un secret bien gardé puisque, malgré une demande récurrente de bien des citoyens et le vote de la loi sur le numérique, obligeant la divulgation des algorithmes administratifs, celui-ci ne l'a toujours pas été intégralement. L'administration concernée a cependant rendu publique une partie de cet algorithme, la seule que nous puissions analyser.

Le chapitre neuf (*Qu'est-ce qu'un programme informatique ?*) explique en quoi consiste un programme informatique.

Comment, lorsqu'un code n'est pas publié, obtenir les renseignements qui ont des chances d'être proches de ce qui est fait ? Le chapitre dix (*La rétro-ingénierie*) nous explique comment faire, en général, dans un tel cas.

Le chapitre onze (*Bases de données*) donne une présentation rapide des bases de données, du langage SQL d'interrogation de celles-ci et du langage PL/SQL utilisé dans la portion de code communiquée par le ministère.

Dans le chapitre douze (*Une partie du cœur d'APB*), nous reproduisons la portion de code utilisé par APB que le Ministère de l'Éducation a fini par envoyer tardivement à une association, « Droit des Lycéens » et nous le commentons largement.

Une enquête sociologique a été menée pour voir comment l'application APB a été perçue par les bacheliers 2016, dont nous donnons les résultats au chapitre treize (*Une enquête sociologique*).

Une des conclusions de l'étude du code communiqué par le ministère est qu'il semble ne pas avoir été conçu selon les règles du Génie logiciel dégagées à la fin des années 1970 pour pallier les défauts des logiciels conçus jusqu'alors, ce qui ne préjuge d'ailleurs en rien *a priori* de la qualité de celui-ci. Le chapitre quatorze (*L'ingénierie des besoins*) nous donne une idée de ces règles, bonnes pratiques qu'il conviendrait de toujours suivre.

Le chapitre quinze (*Méthodologie de construction d'un programme*) continue cette étude. Nous venons de voir qu'une association a diffusé un code sur son site Web, ce qui a suscité des commentaires de la part d'un certain nombre de contributeurs. Ce type de travail collaboratif est l'une des caractéristiques de la communauté des *logiciels libres*.

Enfin, pour conclure, nous suggérons quelques mesures qui pourraient être prises pour éviter toute dérive des algorithmes administratifs et, par là même, restaurer la confiance de ceux qui s'en défient.

10. Le *droit positif*, du latin *positum* « posé », désigne le droit tel qu'il existe réellement, par opposition, par exemple, au « droit naturel ».

Table des matières

| | |
|---|------------|
| Préface | iii |
| I Les algorithmes administratifs | 1 |
| 1 Les algorithmes et les citoyens | 5 |
| 1.1 Informatisation de la société | 5 |
| 1.2 Heurs et malheurs des ordinateurs dans la vie quotidienne | 7 |
| 1.3 Ordinateurs et vie du citoyen | 9 |
| 1.4 Notre étude de cas | 12 |
| 1.5 Bibliographie | 14 |
| 2 L'Administration et ses procédures | 17 |
| 2.1 L'administration des Cités | 18 |
| 2.2 Les procédures administratives | 21 |
| 2.3 Les algorithmes administratifs | 22 |
| 2.4 Bibliographie | 23 |
| II L'affectation | 25 |
| 3 L'affectation des ressources | 27 |
| 3.1 L'affectation des ressources de la Cité | 27 |
| 3.2 Typologie des procédures administratives d'affectation | 29 |
| 3.3 Le cas de l'affectation dans l'enseignement supérieur | 31 |
| 3.3.1 Affectation dans une université | 31 |
| 3.3.2 Le choix de la filière | 34 |
| 3.4 Bibliographie | 36 |
| 4 L'appariement | 37 |
| 4.1 La théorie du choix social | 38 |
| 4.1.1 Des paradoxes du vote à la théorie du choix social | 38 |
| 4.1.2 Le choix social non dictatorial : évidence des principes et impossibilité théorique | 40 |
| 4.1.3 La théorie du choix social et la question de la sincérité des individus | 41 |
| 4.2 Les modèles d'appariement | 42 |
| 4.2.1 Une décision centralisée | 42 |
| 4.2.2 Les bonnes propriétés d'un modèle d'appariement | 42 |
| 4.2.3 L'algorithme d'acceptation différée | 44 |
| 4.2.4 De l'appariement matrimonial à l'affectation scolaire | 45 |

| | | |
|------------|--|------------|
| 4.2.5 | Classements incomplets, indifférences et tirage au sort : le cas des formations non séle | |
| 4.3 | Le cas des pseudo-classements | 48 |
| 4.4 | Conclusion | 48 |
| 4.5 | Bibliographie | 48 |
| III | Que nous apprend l'interface ? | 51 |
| 5 | L'interface d'APB | 53 |
| 5.1 | La procédure d'admission post-bac | 54 |
| 5.2 | L'interface APB pour le candidat | 55 |
| 5.3 | L'interface APB pour le responsable de filière | 58 |
| 5.3.1 | Le paramétrage des licences par les universités | 61 |
| 5.3.2 | Orientation active | 62 |
| 5.3.3 | Classement dans le cas des filières sélectives | 63 |
| 5.4 | Premières incidences de l'algorithmisation de l'affectation | 65 |
| IV | Que nous apprend l'algorithme ? | 69 |
| 6 | Qu'est-ce qu'un algorithme ? | 71 |
| 6.1 | La notion informelle d'algorithme | 72 |
| 6.2 | Algorithmes et programmes informatiques | 78 |
| 6.3 | Définition formelle des algorithmes | 79 |
| 6.3.1 | Historique | 79 |
| 6.3.2 | Définition des ASM | 80 |
| 6.3.3 | Propriété fondamentale des ASM | 83 |
| 6.4 | Les algorithmes d'un point de vue juridique | 84 |
| 6.5 | Bibliographie | 89 |
| 7 | Une partie de l'algorithme d'APB | 91 |
| 7.1 | Le rôle d'une association de lycéens | 92 |
| 7.2 | Une version de l'algorithme de classement | 95 |
| 7.2.1 | Le document | 95 |
| 7.2.2 | Édulcoration d'un document semi-public | 97 |
| 7.3 | Leçons de l'algorithme d'APB | 99 |
| 7.3.1 | Ce que nous apprend l'algorithme de plus que le guide du candidat | 99 |
| 7.3.2 | Quelques propositions d'amélioration | 100 |
| 7.4 | Instabilité de l'appariement | 101 |
| 7.4.1 | APB est instable | 101 |
| 7.4.2 | Rétablir la stabilité : le cas de la PACES en 2016 | 102 |
| V | Que nous apprend le code ? | 103 |
| 8 | Qu'est-ce qu'un programme informatique ? | 105 |
| 8.1 | Introduction aux langages de programmation | 106 |
| 8.1.1 | Qu'est-ce qu'un ordinateur ? | 106 |
| 8.1.2 | Le principe d'un langage de programmation | 106 |
| 8.1.3 | Exemples | 108 |
| 8.1.4 | La sémantique dénotationnelle | 108 |

| | | |
|------------|--|------------|
| 8.1.5 | La sémantique opérationnelle | 109 |
| 8.1.6 | Opérationnalité d'une expression | 110 |
| 8.1.7 | Déclaration des variables | 111 |
| 8.1.8 | Opérationnalité de l'affectation : $x := e$; | 111 |
| 8.1.9 | La conditionnelle : IF e THEN b_1 ELSE b_2 END IF; | 111 |
| 8.1.10 | L'itération bornée : FOR id IN $e_1..e_2$ LOOP b END LOOP; | 112 |
| 8.2 | Bibliographie | 112 |
| 9 | La rétro-ingénierie | 113 |
| 10 | Bases de données | 115 |
| 10.1 | Les bases de données relationnelles | 115 |
| 10.2 | SQL | 116 |
| 10.2.1 | Création d'une base de données | 117 |
| 10.2.2 | Transactions : insertion et mise à jour | 118 |
| 10.2.3 | Projection | 119 |
| 10.3 | PL/SQL | 125 |
| 11 | Une partie du code d'APB | 131 |
| 11.1 | Un sous-programme | 132 |
| 11.2 | Analyse du sous-programme | 143 |
| 11.2.1 | Le contexte | 143 |
| 11.2.2 | Commentaires | 147 |
| 11.2.3 | Quelques interrogations | 166 |
| 11.3 | Les informations déduites du code | 168 |
| 11.4 | Une suggestion | 169 |
| 11.5 | Bibliographie | 170 |
| VI | Qu'en pensent les administrés ? | 171 |
| 12 | Une enquête sociologique | 173 |
| 12.1 | Construction et diffusion d'un questionnaire sur les utilisations et les représentations autour d'APB174 | |
| 12.2 | Transparence et imposition de problématique | 176 |
| 12.3 | APB? Peut mieux faire! | 179 |
| 12.4 | Familiarité informatique et stratégies | 182 |
| VII | Comment améliorer ? | 199 |
| 13 | L'ingénierie des besoins | 201 |
| 13.1 | Contexte | 202 |
| 13.1.1 | Les concepts de base (ou la terminologie) | 202 |
| 13.1.2 | Le logiciel et ses caractéristiques | 202 |
| 13.1.3 | La crise du logiciel et les problèmes de l'ingénierie du logiciel | 203 |
| 13.2 | Les activités de développement d'un logiciel | 204 |
| 13.2.1 | Étape d'« analyse des besoins » | 204 |
| 13.2.2 | Étape de « conception » | 204 |
| 13.2.3 | Étape de « programmation et tests » | 205 |
| 13.2.4 | Étape de déploiement | 205 |
| 13.2.5 | Étape de maintenance | 205 |

| | | |
|-----------|--|------------|
| 13.3 | L'ingénierie des besoins | 205 |
| 13.3.1 | Besoins et exigences | 205 |
| 13.3.2 | Les catégories d'exigences | 206 |
| 13.3.3 | Processus d'ingénierie des exigences | 206 |
| 13.4 | Les modèles de spécification : l'application APB | 207 |
| 13.4.1 | Une approche objet : UML appliquée au cas APB | 207 |
| 13.5 | Conclusion | 213 |
| 13.6 | Bibliographie | 214 |
| 14 | Conclusion | 215 |
| 14.1 | Recommandations sur les algorithmes administratifs | 215 |
| 14.2 | Conclusions sur APB | 217 |
| | Index | 219 |