



HAL
open science

Élaboration de la disposition AZERTY modernisée

Anna Maria Feit, Mathieu Nancel, Daryl Weir, Gilles Bailly, Maximilian John, Andreas Karrenbauer, Antti Oulasvirta

► **To cite this version:**

Anna Maria Feit, Mathieu Nancel, Daryl Weir, Gilles Bailly, Maximilian John, et al.. Élaboration de la disposition AZERTY modernisée. 2018. hal-01826476

HAL Id: hal-01826476

<https://inria.hal.science/hal-01826476>

Preprint submitted on 29 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

Élaboration de la disposition AZERTY modernisée

Anna Feit¹, Mathieu Nancel^{2,1}, Daryl Weir¹, Gilles Bailly³, Maximilian John⁴, Andreas Karrenbauer⁴, Antti Oulasvirta¹
¹Aalto University ²Inria ³Sorbonne Université, CNRS, ISIR ⁴Max Planck Institute for Informatics

Document de travail utilisé dans la rédaction de l'annexe F "Élaboration de la disposition AZERTY modernisée" de la norme AFNOR Z 71-300 : "Dispositions de clavier bureautique français" [9].

Rédigé entre mars et août 2017.
Mis à jour entre mai et juin 2018.
Soumis le 29 juin 2018.
Publié le XX juillet 2018.

1. Introduction

Choisir la meilleure disposition des caractères sur un clavier consiste à définir l'équilibre optimal entre plusieurs besoins liés à son utilisation, pour différentes catégories d'utilisateurs et d'usages. Décider où placer un caractère est un problème difficile. Par exemple, placer le caractère '[' sur une touche très accessible peut faciliter l'entrée de texte pour un groupe d'utilisateurs (informaticiens), mais priver d'autres utilisateurs de cet emplacement privilégié pour un autre caractère qui leur serait plus utile. Ce type de compromis s'applique pour chaque caractère, chaque emplacement (touche ou combinaison de touches), et chacune des relations entre ces caractères et ces emplacements ; cela le rend impossible à résoudre à la main en un temps raisonnable sans faire de présupposition sur le "bon" positionnement des caractères, ce qui peut entraîner l'omission de solutions potentiellement meilleures.

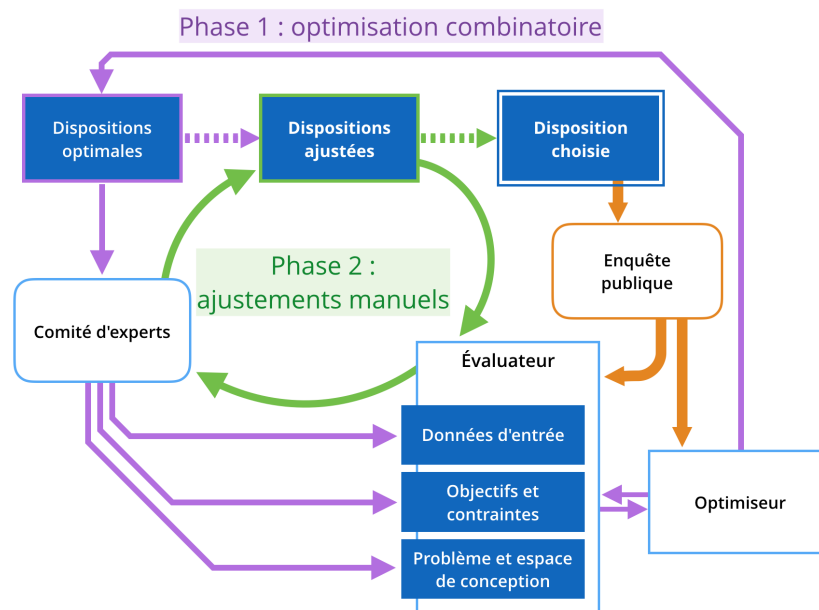


Figure 1 : La disposition AZERTY a été conçue en deux phases : (1) une phase d'optimisation dans laquelle toutes les dispositions possibles sont évaluées automatiquement afin d'en choisir la meilleure pour un jeu de critères donné ; (2) une phase d'ajustements dans laquelle les membres du comité apportent des changements spécifiques, tout en minimisant l'impact négatif de ces modifications sur ces mêmes critères. Leur évaluation est calculée de la même manière dans les deux phases.

La conception de la nouvelle disposition AZERTY s'est faite en deux phases (Figure 1) : une phase d'optimisation combinatoire permettant d'obtenir une disposition optimale pour un jeu de critères donnés pouvant être formulés mathématiquement (vitesse de saisie, similarité avec l'AZERTY "traditionnel", groupement de caractères similaires, et risques de troubles musculo-squelettiques) ; et une phase d'ajustements manuels durant laquelle des modifications locales

ont été effectuées pour satisfaire des critères subjectifs et/ou n'ayant pas pu être formulés dans l'optimiseur. Chacune de ces phases a consisté en un processus itératif, dans lequel les membres du comité sont intervenus à chaque étape :

- pour ajuster les paramètres de l'optimiseur (phase 1) à partir des dispositions optimales produites, jusqu'à obtenir une disposition de travail satisfaisante,
- pour proposer directement des modifications de cette disposition (phase 2), et décider de leur application en tenant compte, entre autres, de leur effet sur les critères d'optimisation.

Suite à un premier cycle entre ces deux phases, une enquête publique a été lancée sur la disposition obtenue. Ses résultats ont motivé des ajustements des paramètres de l'outil d'évaluation et des contraintes de l'optimiseur, qui ont à leur tour lancé un nouveau cycle (phase 1 puis phase 2).

Cette annexe décrit la collecte de données et la formulation des critères d'évaluation utilisés dans les deux phases, ainsi que la conception de l'outil d'optimisation utilisé en phase (1).

L'approche choisie en phase 1 s'appuie sur les dernières avancées de la recherche en optimisation combinatoire – et plus précisément la programmation linéaire en nombre entiers – et en conception de claviers. De par son aspect systématique et flexible, cette approche permet de trouver, parmi un très grand nombre de dispositions possibles, la solution qui répond au mieux à l'équilibre souhaité entre les quatre objectifs établis par le comité. Les propriétés de l'algorithme utilisé garantissent mathématiquement que la disposition produite est une bonne solution, quel que soit le nombre de solutions possibles. Trois problèmes ont été résolus : 1) la définition du problème d'affectation d'un caractère à un emplacement, 2) la collection de données représentatives et le développement de modèles pour les quatre objectifs de conception, et 3) la conception d'un solveur efficace pour le problème adressé.

2. Le processus de conception

Notre approche s'appuie sur les méthodes d'optimisation combinatoire pour concevoir une nouvelle disposition AZERTY. Les spécifications de ce processus de conception ont été élaborées avec un comité d'experts (incluant entre autres des experts de la langue française, des professionnels de la production de claviers physiques, et des chercheurs en Interaction Homme-Machine) afin de calibrer de manière itérative les paramètres de l'algorithme d'optimisation (ou "optimiseur"), et de définir les corpus de texte à utiliser afin de représenter au mieux les différents groupes d'utilisateurs et les tâches d'entrée de texte considérées. Le système final optimise l'arrangement spatial d'un ensemble prédéfini de caractères sur un ensemble d'emplacements fixes du clavier (CF Figure 2), à partir d'objectifs normatifs définis par un comité d'experts et d'observations statistiques sur l'utilisation contemporaine de la langue française.

L'initialisation de l'algorithme d'optimisation s'effectue en trois principales étapes :

1. *Formulation du problème et espace de conception* : le problème consiste à trouver, pour chaque caractère considéré, une position parmi les emplacements disponibles sur le clavier physique qui satisfasse au mieux un ensemble d'objectifs. Cette formulation permet l'application de méthodes de recherche opérationnelle existantes pour trouver de très bonnes solutions au problème. Elle permet également de caractériser précisément l'espace des solutions possibles.
2. *Objectifs et contraintes* : le comité de normalisation énonce les caractéristiques souhaitées (objectifs) et nécessaires (contraintes) de la nouvelle disposition AZERTY, qui sont regroupées sous la forme de critères d'évaluation et formulées mathématiquement. Ces objectifs ne considèrent pas seulement la performance de la saisie, mais aussi la similarité avec la précédente disposition AZERTY, le regroupement spatial des caractères similaires, et les risques de troubles musculo-squelettiques.
3. *Données en entrée* : Nous avons collecté un grand nombre de données afin de paramétrer nos critères d'évaluation. Ceci inclut des données concernant la durée du mouvement d'une touche à une autre, des estimations ergonomiques, une mesure de distance entre les emplacements du clavier prenant en compte les modificateurs, et une quantification de la similarité entre caractères. Nous avons également collecté des corpus de textes représentatifs

des usages courants, allant de textes formels à des publications sur médias sociaux. Ceci garantit que le processus d'optimisation est en adéquation avec la majorité des besoins pour l'entrée de texte.

Une fois réalisé (à l'aide du solveur commercial Gurobi¹), l'algorithme d'optimisation a été utilisé en interaction avec le comité d'experts pour explorer des trillions d'arrangements différents de caractères sur le clavier. Le comité a défini plusieurs scénarios, par exemple en contraignant le positionnement de certains caractères à certains emplacements, ou en explorant différentes pondérations des critères d'évaluation pour favoriser par exemple des dispositions plus proches de celle du précédent AZERTY, ou qui permettent une saisie plus rapide. L'optimiseur produit des dispositions optimales selon les critères d'évaluation définis plus haut, qui peuvent alors être évaluées qualitativement par le comité. Ceci a entraîné des ajustements dans les scénarios ou dans les pondérations, ce qui a produit de nouvelles dispositions optimisées, et ainsi de suite (flèches violettes en Figure 1). Finalement, des ajustements manuels ont été explorés par le comité à partir de la solution la plus satisfaisante de l'optimiseur. Ils ont consisté à déplacer ou échanger les emplacements de différents caractères afin d'ajuster localement l'équilibre entre les critères d'évaluation. Le même algorithme d'évaluation des fonctions objectif a permis de surveiller les effets de chacun de ces changements sur chacun des quatre critères (flèches vertes en Figure 1).

La disposition finale n'est donc pas nécessairement celle minimisant tous les critères, ni celle minimisant la fonction objectif globale. Des choix de conception sont apparus plus tard dans le processus, ou n'ont pas pu être implémentés mathématiquement, ou ont été le résultat d'un consensus parmi le groupe d'experts ou les participants à l'enquête publique (flèches oranges en Figure 1) malgré les scores de la fonction objectif.

Le résultat obtenu avec ce processus est une disposition des caractères qui améliore significativement l'utilisabilité des claviers pour une large proportion des utilisateurs, validée à la fois quantitativement à l'aide de la formalisation mathématique des critères d'évaluation, et qualitativement par le comité d'experts.

La suite du document décrit la définition et la configuration du processus d'optimisation en détail.

3. Formulation du problème et espace de conception

Le problème de conception consistant à trouver la meilleure disposition des caractères sur le clavier peut être formulé comme le problème d'arrangement des lettres : *Étant donnés N caractères et M emplacements ($N \leq M$), quel est l'arrangement des caractères sur les emplacements qui minimise le coût de l'interaction pour saisir un caractère C_j après n'importe quel caractère C_i ?*

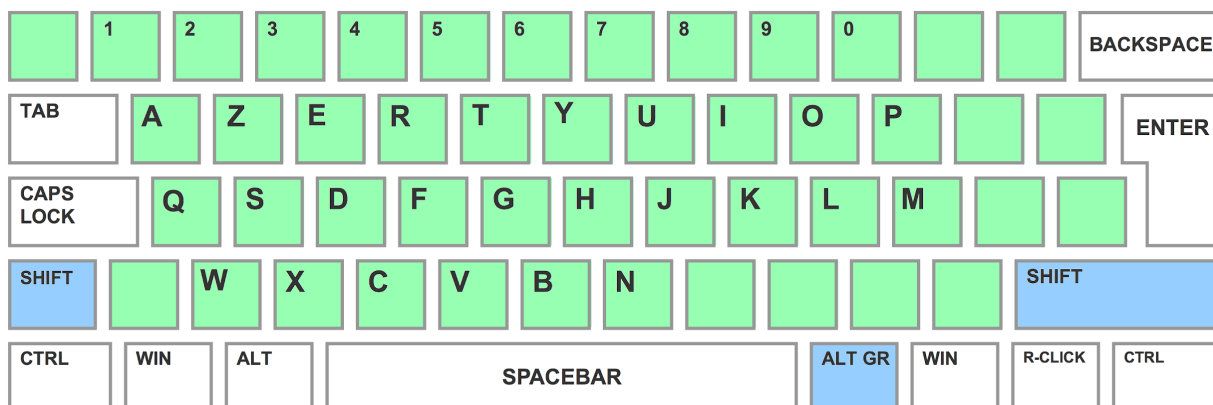


Figure 2 : Emplacements et modifieurs considérés.

¹ <http://www.gurobi.com/>

L'ensemble des N caractères à placer, $C = \{C_1, C_2, \dots, C_N\}$ a été déterminé par le comité d'experts. L'ensemble des emplacements $S = \{S_1, S_2, \dots, S_M\}$ consiste en la zone dite alphanumérique de l'arrangement de clavier harmonisé à 48 touches tel que défini dans le standard ISO/IEC² (voir la zone verte sur la Figure 2). Les emplacements non considérés sont ceux occupés par des caractères fixes, à savoir les positions des lettres A-Z minuscules et majuscules et des chiffres 0-9. Nous avons considéré des scénarios allant jusqu'à 122 caractères spéciaux à placer (à comparer avec les 46 du précédent AZERTY), et jusqu'à 129 emplacements : 22 emplacements sans modificateurs, 12 emplacements utilisant le modifieur Shift, 48 emplacements utilisant Alt Gr, et 48 emplacements utilisant à la fois Alt Gr et Shift (les modificateurs sont indiqués en bleu sur la Figure 2).

Avec 129 emplacements et jusqu'à 122 caractères, il y a plus de 10^{213} configurations possibles, un espace qu'il est inconcevable d'explorer manuellement. En revanche, l'optimisation combinatoire permet d'explorer systématiquement de tels nombres de configurations, et de les évaluer simultanément sur plusieurs critères à la fois et sans a priori. De plus, elle offre des garanties numériques sur la qualité de la disposition obtenue.

Pour répondre à la question de savoir si un caractère doit être placé sur un certain emplacement, il faut considérer la relation de ce caractère avec tous les autres caractères et les emplacements sur lesquels ils sont placés. Par exemple, pour décider où placer le caractère 'é', nous devons considérer la position du caractère 'e' car ils peuvent être entrés l'un à la suite de l'autre. Ceci rend le problème d'arrangement "quadratique" : non seulement chaque caractère et chaque emplacement doivent être considérés, mais également les relations entre toutes les paires de caractères et d'emplacements. Ce type de problème est particulièrement difficile à résoudre. Des travaux de recherche précédents ont utilisé de tels termes quadratiques [2, 5], mais sur des ensembles de caractères et un espace de conception bien plus petits. À notre échelle, trouver une solution globalement optimale faisant intervenir de nombreux termes quadratiques n'est pas possible en un temps raisonnable (ce problème appartient à la classe des problèmes dits NP-complets [2]).

Pour cette raison, nous proposons une simplification du problème qui limite les facteurs quadratiques dans notre formulation. La modification s'appuie sur l'hypothèse que le pourcentage de caractères spéciaux utilisés consécutivement est négligeable, et ainsi que la majorité des caractères spéciaux sont utilisés avant ou après les 26 lettres de l'alphabet ou un espace, $L = \{L_1, L_2, \dots, L_{27}\}$. La position des lettres sur le clavier étant fixe, la minimisation d'un coût d'interaction $I(i, c)$ correspondant à la saisie d'un caractère spécial C_i avant ou après une lettre fixe L_c devient un problème linéaire :

$$\min \sum_{i=1}^n \sum_{c=1}^{27} I(i, c)$$

Objectifs et Contraintes

Le but de cette standardisation est de définir une nouvelle disposition des caractères sur le clavier qui, en comparaison avec le précédent AZERTY, (1) facilite la saisie de texte en Français correct, (2) permet l'accès à davantage de caractères spéciaux fréquemment utilisés dans les langues européennes, la programmation, les mathématiques, etc., et (3) soit intuitive à utiliser et facile à appréhender. Nous avons formulé quatre critères d'évaluation à partir de ces objectifs de conception qui déterminent le coût I de la saisie d'un caractère spécial avant ou après une lettre de l'alphabet. Nous décrivons ces critères et présentons ensuite leur formulation mathématique et le problème d'optimisation complet comme un programme linéaire en nombres entiers.

Nous considérons les 4 critères d'évaluation suivants :

- Performance : Pour faciliter la saisie de texte français correct, notre solution doit permettre de minimiser le temps nécessaire pour saisir des caractères spéciaux fréquemment utilisés avant ou après une lettre fixe (p. ex. é + e).

² "ISO/IEC 9995-2:2009 - Information technology -- Keyboard layouts for" 11 Dez.. 2009, <https://www.iso.org/standard/51644.html>.

- Associations : Pour garantir une disposition cohérente et intuitive, les caractères similaires doivent être situés à proximité les uns des autres, p. ex. '(' et ')'.
Note : Il s'agit là du seul objectif qui considère des relations entre caractères spéciaux. Nous ne considérons qu'un nombre restreint de similarités entre caractères, et pouvons ainsi garantir que de bonnes dispositions peuvent toujours être trouvées en un temps raisonnable.
- Familiarité : Pour faciliter l'adaptation à la nouvelle disposition, les caractères spéciaux les plus utilisés doivent être placés à proximité de leur emplacement dans la disposition AZERTY précédente.
- Ergonomie : Pour faciliter l'entrée de texte français correct, les paires de lettres fréquentes doivent être ergonomiquement faciles à saisir. Exécuter de façon répétée les mêmes mouvements extrêmes peut entraîner des troubles musculo-squelettiques [1,3]. En particulier, saisir des caractères nécessitant des modificateurs (Alt, Shift) peut favoriser des postures inconfortables du poignet et des doigts.

Le problème est défini comme un programme linéaire en nombres entiers à l'aide des variables suivantes :

- $\{x_{ik} \mid i = 1, \dots, N, k = 1, \dots, M\}$ est égal à 1 si le caractère C_i est placé sur l'emplacement S_k , 0 sinon.
- $L = \{L_1, L_2, \dots, L_{27}\}$ est l'ensemble des 26 lettres de l'alphabet et espace, qui ont une position fixe sur le clavier. Pour une lecture simplifiée, nous utiliserons dans la suite du document la même variable pour indiquer le caractère fixe L_c et l'emplacement physique correspondant sur le clavier.

Nous pouvons alors formaliser mathématiquement les quatre critères d'évaluation comme suit :

Performance:

Les séquences caractère-lettre ou lettre-caractère fréquentes devraient être rapides à saisir. Le score de performance est donc défini comme le terme linéaire :

$$P = \sum_{i=1}^n \sum_{k=1}^m \sum_{c=1}^{27} (p_{ci}t_{ck} + p_{ic}t_{kc}) x_{ik}$$

où i est un indice sur les n caractères spéciaux, k est un indice sur les m emplacements disponibles sur le clavier, et c un indice sur les 27 lettres fixes. p_{ic} représente la fréquence de la séquence $C_i \rightarrow L_c$ (et réciproquement p_{ci}), et t_{kc} quantifie le temps pour se déplacer de l'emplacement S_k à l'emplacement L_c (et réciproquement t_{ck}). Il est important de noter que le temps de mouvement n'est pas symétrique, c'est à dire que $t_{kc} \neq t_{ck}$.

La collection de données de performance t et de fréquence p est détaillée plus bas.

Associations:

Les caractères spéciaux devraient être placés à proximité de caractères (lettres ou non) qui leurs sont similaires. Le score d'Associations est défini avec un terme linéaire et un terme quadratique :

$$A = \sum_{i=1}^n \sum_{k=1}^m \sum_{c=1}^{27} s_{ic} d_{kc} x_{ik} + \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^m \sum_{l=1}^m s_{ij} d_{kl} x_{ik} x_{jl}$$

où i, j sont des indices sur les n caractères spéciaux, k, l sont des indices sur les m emplacements disponibles, et c est un indice sur les 27 lettres. s_{ic} et s_{ij} représentent une estimation de la similarité entre le caractère C_i et la lettre L_c , et les caractères C_i et C_j respectivement. d_{kc} quantifie la distance entre l'emplacement de la lettre fixe L_c et l'emplacement S_k , et d_{kl} la distance entre les emplacements S_k et S_l . Noter que le score de similarité et la distance entre deux emplacements sont tous les deux symétriques.

Les notions de distance d entre emplacements et de similarité s entre caractères sont détaillés plus bas.

Familiarité:

Les caractères devraient être situés à proximité de leur position dans la disposition AZERTY précédente, le cas échéant. Le score de familiarité est défini comme le terme linéaire :

$$F = \sum_{i=1}^n \sum_{k=1}^m p_i d_{ki} x_{ik}$$

où p_i représente la fréquence du caractère C_i et d_{ki} la distance entre l'emplacement S_k et l'emplacement du caractère C_i sur la disposition AZERTY précédente (la notation est simplifiée pour faciliter la lecture).

Ergonomie :

La saisie de séquences de caractères fréquentes ne doit pas causer de mouvements ni de postures inconfortables pour le poignet et des doigts. Le score d'ergonomie est défini comme le terme linéaire :

$$P = \sum_{i=1}^n \sum_{k=1}^m \sum_{c=1}^{27} (p_{ci} e_{ck} + p_{ic} e_{kc}) x_{ik}$$

où e_{ck} représente le coût ergonomique de se déplacer de l'emplacement de la lettre L_c à l'emplacement S_k (et réciproquement e_{kc}). Noter que le score d'ergonomie n'est pas symétrique, c'est à dire que $e_{kc} \neq e_{ck}$.

La quantification du coût ergonomique e est détaillée plus bas.

La fonction objectif

Ces quatre critères permettent de formuler la fonction objectif de l'optimiseur comme le programme linéaire en nombres entiers suivant :

$$\min w_P P + w_A A + w_F F + w_E E$$

Avec pour contraintes :

$$\begin{aligned} x_{ik} &= x_{jl} && \forall i, j \in \{1, \dots, n\} \text{ s.t. } C_j \text{ capitalized } C_i \\ & && \forall k, l \in \{1, \dots, m\} \text{ s.t. } S_l \text{ shifted } S_k \\ \sum_{k=1}^m x_{ik} &= 1 && \forall i \in \{1, \dots, n\} \\ \sum_{i=1}^n x_{ik} &\leq 1 && \forall k \in \{1, \dots, m\} \\ x_{ik} &\in \{0,1\} && \forall i \in \{1, \dots, n\}, \forall k \in \{1, \dots, m\} \end{aligned}$$

La fonction à minimiser est la somme pondérée des quatre critères d'évaluation présentés ci-dessus, dont la somme des poids $w_P + w_A + w_F + w_E = 1$. La première contrainte garantit qu'un caractère en majuscule doit être assigné à l'emplacement "Shift"-é de son équivalent en minuscule (Alt+Shift si le caractère minuscule est sur un emplacement Alt). Les seconde et troisième contraintes garantissent l'unicité des assignations : un caractère est assigné à un emplacement unique, et un emplacement ne peut être assigné qu'à un unique caractère au maximum. La dernière contrainte garantit la binarité des variables de décision.

Paramètres des critères d'évaluation et collecte de données

Le succès d'un processus d'optimisation réside en partie dans la qualité des données utilisées dans le calcul de la fonction objectif, qui doivent être suffisamment représentatives des tâches et des utilisateurs modélisés. Nous décrivons ici la collecte et le traitement des données utilisées dans les critères d'évaluation des dispositions de caractères sur le clavier décrits dans la section précédente.

Statistiques de la langue française

La plupart des critères d'évaluation de l'optimiseur nécessitent de connaître la distribution statistique de tout caractère et digramme (séquence de deux caractères) utilisés dans la langue française. Afin de refléter au mieux l'usage moderne du clavier français, le comité d'experts a collecté des corpus de texte dans trois catégories distinctes axées sur l'exactitude orthographique, grammaticale, et l'utilisation correcte des caractères spéciaux (catégorie "Formelle"), les expressions usuelles et l'utilisation de vocabulaire émergent (catégorie "Populaire"), et l'amélioration de l'accès aux caractères utilisés en programmation (catégorie "Code"). Les corpus ont été soit obtenus par collecte automatique depuis Internet, soit fournis par l'ELRA (European Language Resources Association). L'ensemble collecté est le plus grand corpus utilisé pour l'optimisation de dispositions de caractères sur clavier français, avec plus de cinq milliards de caractères au total.

Catégorie	Corpus	Nb caract.	Poids par corpus	Commentaires	Source
Formelle	Wikipédia en Français	4811 M	0,07	Tous les articles en Juin 2014	Collection automatique
	Politiques environnementales (x2)	417 M	0,14	[13, 15]	ELRA
	Législation du travail (x2)	490 M	0,14	[14, 16]	
	Q&R au Journal Officiel de l'Union Européenne	2 M	0,07	[11]	
	Retranscriptions d'émissions de radio	10 M	0,07	[18, 19]	
	Articles de journaux d'information	162 M	0,07	Le Monde [10], Le Monde Diplomatique [12], Est Républicain.	
Populaire	E-mails anonymisés	1 M	0,07	[17]	Collection automatique
	Facebook	2 M	0,07	1000 <i>posts</i> les plus récents de 10 comptes célèbres.	
	Twitter	1 M	0,15	1000 <i>posts</i> les plus récents de 10 comptes célèbres. Exclusion de 75% des URLs pour permettre la généralisation de nos statistiques.	
Code	Python, C++, Java, Javascript, CSS, HTML	9 M	0,15	10 projets par langage, de plus de 1000 LdC chacun, sans le contenu des commentaires.	

La catégorie "Formelle" est constituée de textes en Français, soit écrits par des professionnels, soit corrigés quotidiennement par une communauté dédiée. Elle inclut l'ensemble des pages Wikipédia en Français en Juin 2014, des textes légaux sur la législation environnementale et du travail, des questions écrites adressées au Journal Officiel de

l'Union Européenne et les réponses correspondantes, des retranscriptions d'émissions de radio, et des articles de journaux d'information français reconnus.

La catégorie "Populaire" contient des emails anonymisés, et les mille publications (*posts*) les plus récentes, en Août 2016, de 10 comptes Twitter et 10 comptes Facebook célèbres. Le contenu, la grammaire, l'orthographe et l'utilisation de caractères spéciaux de ces publications n'ont pas été corrigés *a posteriori* afin de refléter un usage réaliste. Nous avons observé que les adresses web (URL) sont très fréquentes dans les tweets, peut-être à cause de la limite des 140 caractères (doublée depuis). Afin de ne pas biaiser notre jeu de données (nous nous intéressons aux usages généraux, pas à Twitter en particulier), nous avons aléatoirement retiré 75% de ces URLs du corpus Twitter.

La catégorie "Code" consiste en des fichiers de texte dans quatre langages de programmation (C++, Java, Python, Javascript) et deux langages de présentation (HTML, CSS). Ces fichiers ont été collectés depuis la plateforme Github à raison de 10 projets de plus de 1000 lignes de code (LdC) par langage. Le contenu des commentaires de chaque fichier a été retiré du corpus, mais pas les délimiteurs de ces commentaires.

Nous avons ensuite calculé les fréquences de chaque caractère et digramme. Les tailles des différents corpus variant drastiquement (de 1 million à 4,8 milliards de caractères), nous avons d'abord calculé ces fréquences pour chaque corpus séparément. La distribution finale (*p*) utilisé dans nos critères d'évaluation est la somme pondérée de ces fréquences pour chaque caractère et digramme. Les pondérations ont été discutés et validés par le comité d'experts.

Temps de mouvement

Le critère de "Performance" favorise la rapidité de saisie des caractères spéciaux avant ou après des lettres normales (non accentuées) ou un espace. Cela nécessite de pouvoir estimer la durée d'un mouvement entre tout emplacement de lettre et tout emplacement destiné à accueillir des caractères spéciaux, dans les deux sens. La conception de certaines dispositions de claviers s'appuie sur des présuppositions quant à la méthode de typage des utilisateurs, par exemple le *touch-typing* à 10 doigts, mais des travaux récents ont montré qu'il existe de grandes disparités d'usage des doigts entre utilisateurs [7, 8], ce qui réduit la validité de ce genre de modèle utilisé "par défaut". Notre approche a donc été d'utiliser des données réelles, recueillies à partir d'utilisateurs nombreux et variés, afin d'obtenir des distributions d'usages plus réalistes.

Il existe de tels jeux de données pour des séquences de lettres simples, mais ils n'incluent pas les emplacements avec modifieurs (Alt, Shift, Alt+Shift). Nous avons donc conduit une étude externalisée à grande échelle (*crowdsourcing*) afin d'obtenir ces données de durées. Nous nous sommes intéressés à tous les mouvements entre les emplacements de lettres fixes et les emplacements destinés aux caractères spéciaux, dans les deux sens, sur des claviers classiques à 48 touches (voir Figure 2). Plus précisément :

- Les **emplacements de lettres** correspondent aux touches alphabétiques classiques du clavier AZERTY, avec ou sans le modifieur Shift (a-z en minuscules et majuscules, sans accents) et à la barre d'espace,
- Les **emplacements pour caractères spéciaux** correspondent aux 22 touches du clavier ne contenant pas de lettres, sans modifieurs ou avec le modifieur Shift, et aux 48 caractères du clavier avec les modifieurs Alt³ et Alt+Shift.

L'ensemble correspond à 7560 séquences possibles d'emplacements. Afin d'obtenir des durées généralisables au-delà de nos participants individuels, nous avons recueilli des données d'au moins trois participants différents pour chacune de ces séquences. En raison du très grand nombre de tests à effectuer, nous avons recruté un large panel international de participants via deux canaux : (1) plus de 630 participants ont été recrutés via la plateforme participative Crowdfunder⁴ ;

³ Il est à noter que "Alt" dans cette expérience correspond à la touche Alt, Alt Gr ou Option à *droite* de la barre d'espace seulement, comme initialement requis par le comité. Les participants ont reçu pour instruction de n'utiliser que cette touche pour Alt. L'usage d'une touche Alt gauche était ignorée par le système, et pouvait ainsi être comptabilisée comme erreur si l'utilisateur la combinait avec une autre touche.

⁴ <https://www.crowdfunder.com/>

(2) environ 270 participants ont été recrutés bénévolement via une annonce sur une page Internet dédiée à l'évaluation de la saisie de texte. Afin de pouvoir recruter le plus de participants possible, les instructions étaient présentées en Anglais.

Avant de commencer l'étude, notre programme détecte la disposition des caractères sur le clavier utilisé par le participant. Le participant doit ensuite effectuer un pré-test dans lequel il doit retranscrire 10 phrases en Anglais, afin d'établir une vitesse de saisie de référence. L'étude à proprement parler consiste en une série de tâches de saisie de 10 caractères. Au début d'une tâche, un stimulus est présenté au participant consistant en deux séquences prononçables (pseudo-syllabes) de quatre lettres en minuscules ou espace, séparées par un caractère spécial. Ce caractère spécial est représenté sous la forme d'un certain nombre de modificateurs et du caractère correspondant à l'emplacement non modifié sur la touche correspondante du clavier du participant (Figure 3). La tâche du participant consiste à saisir ces lettres et caractère, dans l'ordre, sans erreur, et le plus rapidement possible.

Il est important de noter que le stimulus dépend du clavier du participant, et que les données recueillies sont liées aux *emplacements sur le clavier*, par aux caractères présentés : par exemple, si la séquence "zeki [Alt]+[M] tuwe" est présentée à un utilisateur de clavier AZERTY, la séquence "weki [Alt]+[;] tuze" sera présentée à un utilisateur de clavier QWERTY. Les informations recueillies dans les deux cas seront utilisées dans le calcul de la fonction objectif de Performance comme si elles avaient été saisies par un clavier AZERTY.

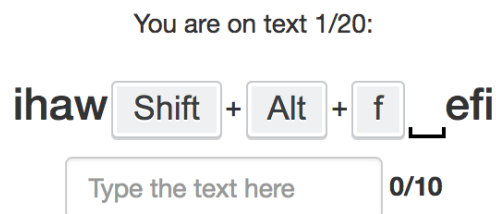


Figure 3 : Exemple de tâche de saisie. Dans cet exemple nous avons recueilli des données de performance pour deux transitions : de l'emplacement de la lettre 'w' à l'emplacement correspondant à [Shift+Alt+f] sur le clavier du participant, et de cet emplacement à la barre d'espace (représentée par le crochet horizontal).

Les pseudo-syllabes utilisées dans les tâches ont été pré-générées avant le début de l'étude, et les séquences correspondant à des mots existants (en Anglais) ou imprononçables ont été retirées à la main. La présence de pseudo-syllabes réalistes permet de garantir que le participant a utilisé sa stratégie habituelle de saisie avant et après le caractère spécial central, tout en minimisant les effets de familiarité avec des mots existants. Les séquences obtenues n'appartenant pas à une langue spécifique, les durées de saisie obtenues sont indépendantes de la langue du participant (performances "humaines"), ce qui a été confirmé à l'aide de tests statistiques à la fin de l'étude.

Pour chaque séquence, le participant doit effectuer 10 saisies sans erreur ; à chaque erreur la saisie est ignorée, et le participant doit la recommencer jusqu'à l'effectuer sans erreur. L'utilisation de la touche 'Retour arrière' (□) ou des touches directionnelles est aussi considérée comme une erreur. Après chaque saisie sans erreur, le programme enregistre (1) le temps écoulé entre la saisie de la dernière lettre de la première pseudo-syllabe et la saisie complète du caractère central, et (2) le temps écoulé entre la saisie du caractère central et la première lettre de la deuxième pseudo-syllabe. Ainsi, chaque tâche fournit des données de durées pour deux séquences d'emplacements. Le nombre de stimuli par participant varie selon qu'il est rémunéré (Crowdfunder, 20 stimuli) ou non (10 stimuli). Pour chaque séquence possible, nous avons recueilli les données d'au moins 3 participants, la plupart des séquences ayant été saisie par 4 ou 5 participants.

Les mesures recueillies ont été traitées comme suit :

- La moyenne des durées des 5 meilleures saisies de chaque participant sur chaque séquence est calculée, afin de filtrer les essais durant lesquels le participant était distrait ou en train d'apprendre la séquence de caractères.
- Cette moyenne est multipliée par le ratio entre la vitesse de saisie du participant, mesurée lors du pré-test, et la vitesse moyenne de saisie de tous les participants. Cela a pour effet de réduire les durées mesurées des

participants plus lents que la moyenne, et inversement, et ainsi d'éviter aux participants très lents ou très rapides de biaiser les données.

- Après inspection de la distribution des valeurs ainsi obtenues, les données des utilisateurs dont les durées moyennes sont exceptionnellement hautes (> 1200 ms) ou basses (< 50 ms) sont exclues.
- Enfin, pour chaque séquence, la moyenne des données de chaque utilisateur restant l'ayant effectuée est utilisée comme valeur t dans le calcul du critère de Performance.

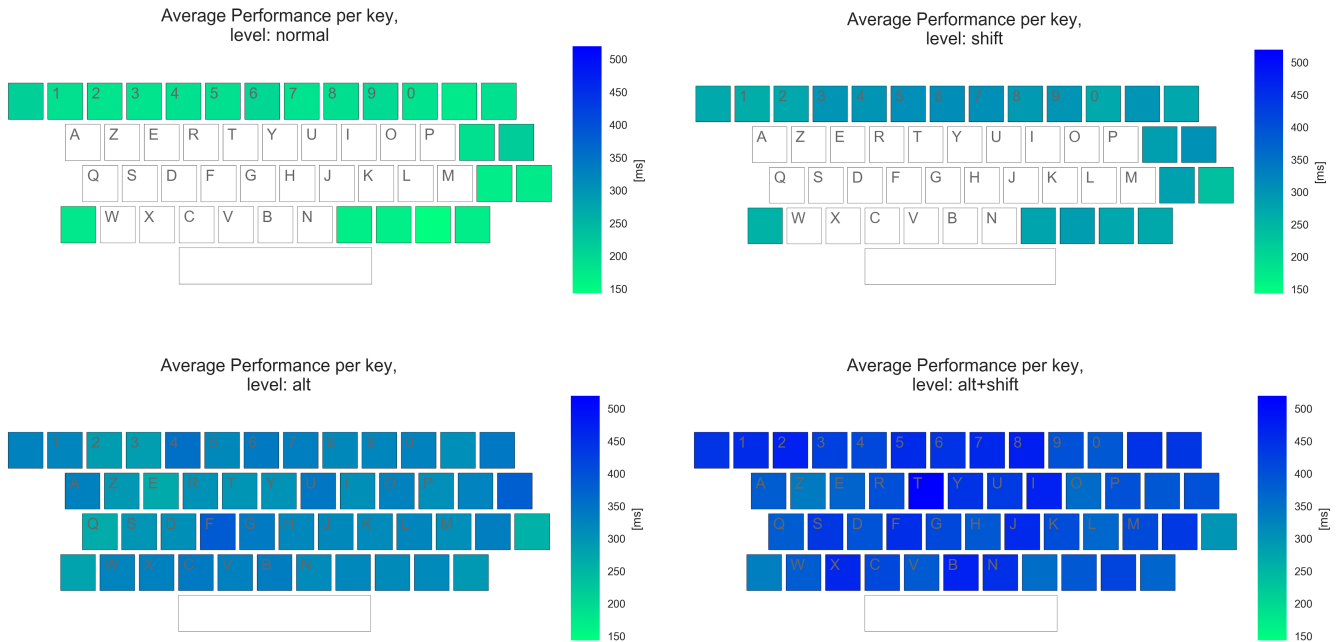


Figure 4 : Durées moyennes de saisie depuis et vers chacun des emplacements considérés, en millisecondes.

La Figure 4 représente les performances obtenues pour chaque emplacement. Il est important de noter que sont montrées ici les moyennes des durées depuis et vers chacun des emplacements représentés (les couleurs encodent des durées en millisecondes) ; chaque séquence, d'une lettre vers un caractère spécial ou inversement, correspond à une valeur différente dans les calculs de l'optimiseur.

Similarités entre caractères

Le but du critère "Associations" est de favoriser la proximité physique entre des caractères similaires. Nous nous attendons à ce que cela facilite la découverte et l'apprentissage de la nouvelle disposition : une fois que la nouvelle localisation d'un caractère est apprise, les caractères similaires seront trouvés plus rapidement par l'utilisateur s'ils sont situés à proximité.

Nous avons défini plusieurs critères ad-hoc de similarité, et avons assigné un poids ($0 < w \leq 1$) à chacun :

- Majuscules ($w=1$), quand un certain caractère est la version en majuscule d'un autre, p. ex. œ et Œ.
- Proximité sémantique ($w=0,9$), quand des caractères ont des significations proches. Par exemple, les caractères \times \cdot $*$ peuvent représenter une multiplication, et $/$ \div une division.
- Inclusion ($w=0,9$), quand un caractère est inclus dans un autre : a^a, o œ, c ç, r ®, etc.
- Complétion ($w=0,9$), quand des caractères se complètent : (), [], "", «», ≅, etc.
- Usage ($w=0,7$), quand des caractères sont déjà associés dans les usages ou la pratique : n ~, c , , etc.
- Alphabet ($w=0,6$), quand un caractère spécial est l'équivalent d'une lettre française dans un alphabet différent : ß et s, ı et i, etc.
- Groupements ($w=0,6$), quand un groupe de caractères peut être associé à un usage particulier, p. ex. les symboles monétaires, les signes de ponctuation, les tirets, etc.

- Similarité visuelle ($w=0,5$), quand des caractères se ressemblent visuellement, p. ex. 8 et ∞, ° et °, etc.
- Similarité phonétique ($w=0,4$), quand des caractères se prononcent de manière semblable, totalement ou en partie, p. ex 3 et j, e et € (euro), r et √ (racine), etc.

Quand deux caractères sont liés par plus d'un de ces critères, nous assignons à cette association le poids le plus élevé. Par exemple, « et < se ressemblent (visuel : 0,5), sont tous les deux des guillemets (groupement : 0,6) et ont des significations très proches, à savoir commencer un dialogue ou une citation (sémantique : 0,9). Le poids de l'association entre « et < sera donc 0,9.

Définir des similarités entre tous les caractères ainsi que l'importance de ces similarités peut rapidement devenir un processus subjectif, c'est pourquoi nous ne l'avons pas entrepris de manière exhaustive. En conséquence, "Associations" est essentiellement un critère indicatif qui favorise la proximité des caractères les plus manifestement semblables. La liste des associations à considérer a été souvent modifiée par le comité durant la phase 1, en fonction des résultats observés.

Distance entre les emplacements

La distance entre deux emplacements sur le clavier (d pour "Familiarité" et "Associations") prend en compte la distance physique entre les touches ainsi que la différence de modificateurs nécessaires à leur saisie.

Il est présupposé que que les utilisateurs naviguent visuellement dans une disposition de clavier en termes de rangées et de colonnes, plutôt que dans l'espace euclidien. Ainsi, la distance entre deux touches du clavier physique est calculée selon la méthode dite de Manhattan : la somme des différences de rangées et de colonnes entre ces deux touches.

À cette distance s'ajoute un score représentant la différence de modificateurs entre les deux emplacements considérés, qui dépend du critère évalué. Pour le calcul de "Familiarité", il est présupposé que les utilisateurs explorent les différents niveaux de modificateurs dans l'ordre décroissant de leur fréquence d'usage : pas de modificateur, puis Shift, Alt, et enfin Alt+Shift. La distance entre les touches est alors incrémentée de 1 pour chaque niveau d'écart ; par exemple, une différence de Shift à Alt+Shift incrémente le score de 2. Pour "Associations", seul le nombre de modificateurs est considéré ; par exemple, une différence de Shift à Alt+Shift incrémente le score de 1.

Le résultat est normalisé entre 0 et 1, 1 correspondant à la distance maximale entre deux emplacements.

Ergonomie

Notre approche vise à minimiser les chances de troubles musculo-squelettiques tels que le syndrome du canal carpien, causé par l'extension et la compression du nerf médian dues à des postures extrêmes du poignet et des doigts ainsi qu'à des mouvements répétitifs. L'adduction et l'extension du poignet, de même que l'étirement et la flexion des doigts, font partie des mouvements les plus à risque [1, 3].

Le critère Ergonomie modélise le coût ergonomique de chaque séquence d'emplacements sur le clavier en prenant en compte séparément la touche et les modificateurs.

Concernant la touche, l'optimiseur tente de minimiser :

1. l'adduction du poignet causée par des mouvements vers les touches les plus latérales du clavier,
2. l'extension du poignet causée par des mouvements vers la rangée la plus haute du clavier.

Concernant les modificateurs, l'optimiseur tente de minimiser :

3. la flexion du poignet pour atteindre les modificateurs,
4. l'extension des doigts pour atteindre une touche éloignée des modificateurs,
5. l'extension des doigts lorsque deux modificateurs (Alt+Shift) sont nécessaires.

Il est important de noter que le risque ergonomique de chacun de ces cas dépend de la stratégie de saisie de l'utilisateur. Par exemple, l'adduction du poignet est nécessaire pour atteindre les touches les plus latérales du clavier avec le système dit de *touch-typing*. Cependant, il est également possible de déplacer la main entière afin d'accéder à ces touches avec ou pas de rotation latérale du poignet, ce qui produit un mouvement plus lent mais ergonomiquement moins risqué. Dans l'impossibilité de considérer toutes les stratégies de saisie existantes à la fois, le calcul des coûts ergonomiques est basé sur une position des mains centrale "standard", qui caractérise la méthode de *touch-typing* classique ainsi que de nombreuses autres méthodes.

Le coût ergonomique e est calculé, pour chaque séquence de deux emplacements, sous la forme d'un indice (entier) de sévérité entre 1 et 4. Cet indice pénalise les déplacements vers les zones extrêmes gauche et droite du clavier (1 point), les mouvements vers la rangée du haut (1 point), et les mouvements nécessitant l'utilisation d'un (1 point) ou deux (2 points) modificateurs depuis un emplacement ne nécessitant pas de modificateur. Ce score est ensuite normalisé entre 0 et 1, 1 correspondant au coût ergonomique maximal (Figure 5).

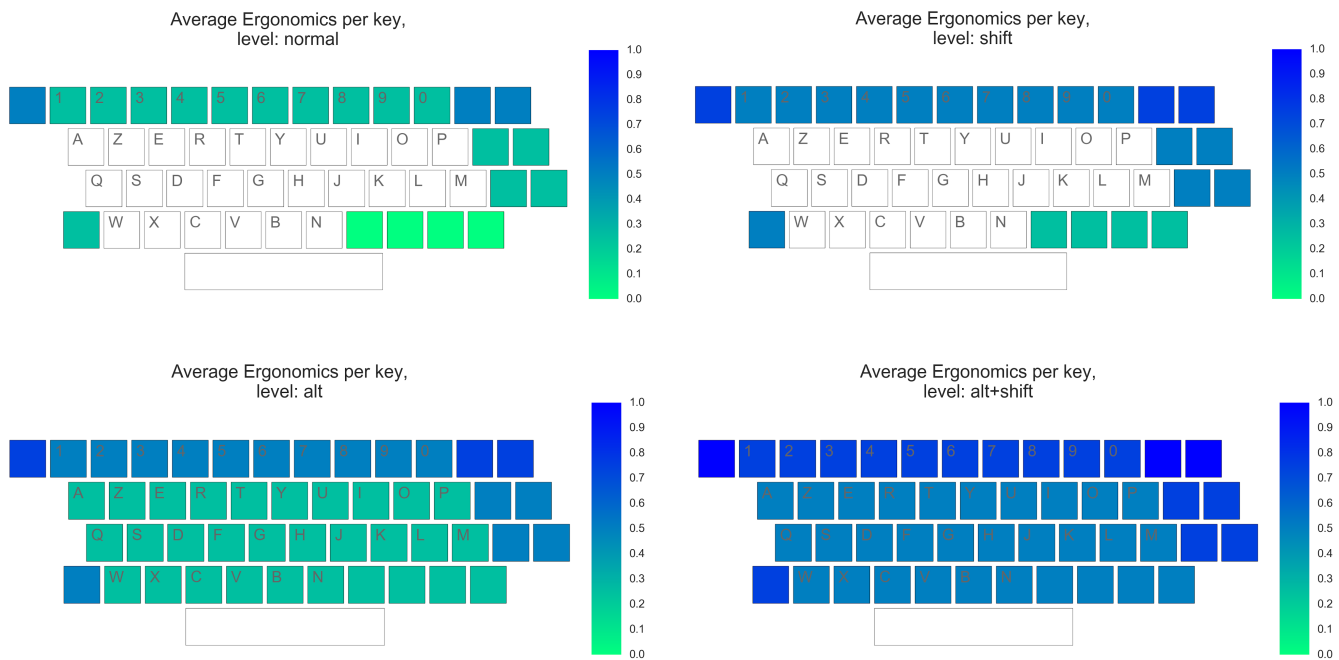


Figure 5 : Coûts ergonomiques normalisés, utilisés par la fonction objectif d'Ergonomie.

Il est important de noter que cette figure représente le coût ergonomique *moyen* d'un déplacement entre l'emplacement représenté et chacune des lettres (sans modificateur) dans les deux sens, le coût ergonomique n'étant pas symétrique.

4. Résultats de l'enquête publique et nouvelle itération

Le processus en deux phases décrit en Introduction a mené à l'élaboration de la disposition présentée lors de l'enquête publique de 2017. Suite au dépouillement des commentaires obtenus (plus de 3700), des changements ont été apportés au processus d'optimisation afin de suivre au mieux les souhaits majoritaires exprimés :

- Réduire le jeu de caractères à placer sur le clavier, notamment en transposant les caractères non-diacritiques plus rares dans des modes spécifiques accessibles via des touches mortes dédiées : symboles monétaires non courants (touche morte ₤), lettres grecques (touche morte μ), caractères utilisés dans la saisie des langues européennes (touche morte Eu).
- Ajuster la fréquence d'apparition des caractères ‘ ’ » « “ ” pour refléter leur usage correct en Français, plutôt que les guillemets droits ' " plus fréquents dans les corpus. Nous avons identifié dans nos corpus Formel et Populaire :

- les instances du guillemet droit simple ' utilisé en tant qu'apostrophe ' et en tant que guillemet ouvrant ' et fermant ' ;
- les instances du guillemet droit double " en tant que guillemets ouvrants «“ et fermants »”.

Nous avons ensuite transféré une partie des fréquences des guillemets droits ' " vers les fréquences des symboles français ' « » “ ” en fonction de leurs usages identifiés.

- Ajuster le poids assigné à la fonction objectif Associations afin de mieux refléter les attentes des utilisateurs.
- Ajuster les poids assignés aux corpus de textes afin de mieux refléter les usages souhaités.
- Fixer la position de @ et # en E00.
- Ajouter une contrainte à l'optimiseur selon laquelle chaque paire de caractères ouvrant-fermant doivent impérativement être placés côte à côte, sur la même rangée, ouvrant à gauche, et accessibles à l'aide des mêmes modifieurs.

Seul le dernier point a nécessité de modifier l'optimiseur lui-même (ajout d'une contrainte). Les autres points n'ont nécessité qu'une modification de ses données d'entrée – ce qui a relancé le processus décrit en Figure 1.

Le placement contigu des paires de caractères ouvrant-fermant est assuré par la contrainte suivante :

$$x_{ik} - x_{jl} = 0 \quad \forall (i, j) \in C, \forall (k, l) \in P$$

$$x_{ik} = 0 \quad \forall (i, j) \in C, \forall k \in R$$

où C est l'ensemble des paires de caractères ouvrant-fermant considérées, P l'ensemble des paires d'emplacements contigus disponibles ayant le même modifieur, et R l'ensemble des emplacements le plus à droite de chaque rangée, avec et sans modifieurs. La première formule s'assure que les paires de caractères sont effectivement côte à côte et dans le bon ordre, la seconde que le premier caractère de chaque paire (le caractère ouvrant) n'est pas placé sur l'emplacement le plus à droite de chaque rangée.

Une fois ces changements effectués, de nouvelles dispositions ont été optimisées pour différentes combinaisons de poids associés aux fonctions objectif (phase 1). Comme précédemment, une fois qu'une disposition optimale a été sélectionnée, les membres du comité ont proposé des ajustements locaux (phase 2) tout en minimisant les conséquences négatives de ces modifications sur les quatre fonctions objectif, pour obtenir la disposition AZERTY finale.

5. Références

1. T. K. Amell and S. Kumar. 2000. Cumulative trauma disorders and keyboarding work. *International journal of industrial ergonomics* 25, 1: 69–78.
2. R. E. Burkard and J. Offermann. 1977. Entwurf von Schreibmaschinentastaturen mittels quadratischer Zuordnungsprobleme. *Zeitschrift für Operations Research* 21, 4: B121–B132.
3. Mircea Fagarasanu and Shrawan Kumar. 2003. Carpal tunnel syndrome due to keyboarding and mouse tasks: a review. *International journal of industrial ergonomics* 31, 2: 119–136.
4. Maximilian John and Andreas Karrenbauer. 2016. A Novel SDP Relaxation for the Quadratic Assignment Problem Using Cut Pseudo Bases. In *Lecture Notes in Computer Science*. 414–425.
5. Andreas Karrenbauer and Antti Oulasvirta. 2014. Improvements to keyboard optimization with integer programming. In *ACM UIST '14*. <https://doi.org/10.1145/2642918.2647382>
6. L. Kaufman and F. Broeckx. 1978. An algorithm for the quadratic assignment problem using Bender's decomposition. *European journal of operational research* 2, 3: 207–211.
7. M. Feit, D. Weir, and A. Oulasvirta. 2016. How We Type: Movement Strategies and Performance in Everyday Typing. In *ACM CHI '16*. DOI: <https://doi.org/10.1145/2858036.2858233>

8. Vivel Dhakal, Anna Maria Feit, Per Ola Kristensson, and Antti Oulasvirta. 2017. Observations on Typing from 136 Million Keystrokes. In ACM CHI '18. DOI: <https://doi.org/10.1145/3173574.3174220>
 9. Dispositions de clavier bureautique français. AFNOR Z 71-300, Association Française de Normalisation (AFNOR), La Plaine Saint Denis, France, 2018.
-
10. Text corpus of "Le Monde" – 1 Month of 2005, 2006 and 2007, ISLRN: 421-401-527-366-2, ELRA ID: ELRA-W0015,
 11. JOC MULTEXT Corpus, ISLRN: 900-482-746-635-0, ELRA ID: ELRA-W0017,
 12. "Le Monde Diplomatique" Text corpus in French - archives 1980-1998, ISLRN: 232-619-161-765-9, ELRA ID: ELRA-W0036-01,
 13. PANACEA English-French and English-Greek parallel corpus acquired for Environment domain, ISLRN: 870-946-931-293-7, ELRA ID: ELRA-W0057,
 14. PANACEA English-French and English-Greek parallel corpus acquired for Labour Legislation domain, ISLRN: 428-891-110-719-1, ELRA ID: ELRA-W0058,
 15. PANACEA Environment French monolingual corpus, ISLRN: 400-316-779-360-9, ELRA ID: ELRA-W0065,
 16. PANACEA Labour French monolingual corpus, ISLRN: 349-917-944-285-0, ELRA ID: ELRA-W0066
 17. EASy Evaluation Package – Emails subset, ISLRN: 238-723-334-894-5, ELRA ID: ELRA-E0034
 18. ESTER Corpus – Transcriptions subset, ISLRN: 055-636-352-982-9, ELRA ID: ELRA-S0241
 19. ESTER 2 Corpus – Transcriptions subset, ISLRN: 123-207-221-143-8, ELRA ID: ELRA-S0338.