

Techniques d'enquête

Un algorithme du recuit simulé pour la stratification et la répartition simultanée de l'échantillon

par Mervyn O'Luing, Steven Prestwich et S. Armagan Tarim

Date de diffusion : le 21 juin 2022



Statistique
Canada

Statistics
Canada

Canada

Comment obtenir d'autres renseignements

Pour toute demande de renseignements au sujet de ce produit ou sur l'ensemble des données et des services de Statistique Canada, visiter notre site Web à www.statcan.gc.ca.

Vous pouvez également communiquer avec nous par :

Courriel à infostats@statcan.gc.ca

Téléphone entre 8 h 30 et 16 h 30 du lundi au vendredi aux numéros suivants :

- | | |
|---|----------------|
| • Service de renseignements statistiques | 1-800-263-1136 |
| • Service national d'appareils de télécommunications pour les malentendants | 1-800-363-7629 |
| • Télécopieur | 1-514-283-9350 |

Normes de service à la clientèle

Statistique Canada s'engage à fournir à ses clients des services rapides, fiables et courtois. À cet égard, notre organisme s'est doté de normes de service à la clientèle que les employés observent. Pour obtenir une copie de ces normes de service, veuillez communiquer avec Statistique Canada au numéro sans frais 1-800-263-1136. Les normes de service sont aussi publiées sur le site www.statcan.gc.ca sous « Contactez-nous » > « Normes de service à la clientèle ».

Note de reconnaissance

Le succès du système statistique du Canada repose sur un partenariat bien établi entre Statistique Canada et la population du Canada, les entreprises, les administrations et les autres organismes. Sans cette collaboration et cette bonne volonté, il serait impossible de produire des statistiques exactes et actuelles.

Publication autorisée par le ministre responsable de Statistique Canada

© Sa Majesté la Reine du chef du Canada, représentée par le ministre de l'Industrie 2022

Tous droits réservés. L'utilisation de la présente publication est assujettie aux modalités de l'[entente de licence ouverte](#) de Statistique Canada.

Une [version HTML](#) est aussi disponible.

This publication is also available in English.

Un algorithme du recuit simulé pour la stratification et la répartition simultanée de l'échantillon

Mervyn O’Luining, Steven Prestwich et S. Armagan Tarim¹

Résumé

La présente étude combine le recuit simulé avec l'évaluation delta pour résoudre le problème de stratification et de répartition simultanée de l'échantillon. Dans ce problème particulier, les strates atomiques sont divisées en strates mutuellement exclusives et collectivement exhaustives. Chaque partition de strates atomiques est une solution possible au problème de stratification, dont la qualité est mesurée par son coût. Le nombre de Bell de solutions possibles est énorme, même pour un nombre modéré de strates atomiques, et une couche supplémentaire de complexité s'ajoute avec le temps d'évaluation de chaque solution. De nombreux problèmes d'optimisation combinatoire à grande échelle ne peuvent être résolus de manière optimale, car la recherche d'une solution optimale exige un temps de calcul prohibitif. Un certain nombre d'algorithmes heuristiques de recherche locale ont été conçus pour résoudre problème, mais ils peuvent rester coincés dans des minima locaux, ce qui empêche toute amélioration ultérieure. Nous ajoutons, à la suite existante d'algorithmes de recherche locale, un algorithme du recuit simulé qui permet de s'échapper des minima locaux et s'appuie sur l'évaluation delta pour exploiter la similarité entre des solutions consécutives, et ainsi réduire le temps d'évaluation. Nous avons comparé l'algorithme du recuit simulé avec deux algorithmes récents. Dans les deux cas, l'algorithme du recuit simulé a permis d'obtenir une solution de qualité comparable en beaucoup moins de temps de calcul.

Mots-clés : Algorithme du recuit simulé; logiciel R; répartition des échantillons; stratification optimale.

1. Introduction

Dans l'échantillonnage aléatoire simple stratifié, une population est divisée en strates mutuellement exclusives et collectivement exhaustives, puis des unités d'échantillonnage sont sélectionnées au hasard dans chacune de ces strates. Les objectifs de la stratification sont abordés dans Cochran (1977). Si les variances intrastrates étaient minimisées, la précision serait améliorée. Il s'ensuit que les petits échantillons qui découlent de chaque strate peuvent être combinés pour donner une petite taille d'échantillon.

À cette fin, nous avons l'intention de construire des strates qui sont homogènes sur le plan interne, mais qui tiennent compte également des mesures aberrantes. Pour y parvenir, nous adoptons une approche qui consiste à rechercher le partitionnement optimal de *strates atomiques* (toutefois, la méthodologie peut également être appliquée aux strates *continues*) créées à partir du produit cartésien de variables de stratification catégorielles, voir Benedetti, Espa et Lafratta (2008); Ballin et Barcaroli (2013, 2020).

Le nombre de Bell, qui représente le nombre de partitions possibles (stratifications) d'un ensemble de strates atomiques, croît très rapidement avec le nombre de strates atomiques (Ballin et Barcaroli, 2013). En fait, il arrive un moment où, même pour un nombre modéré de strates atomiques et les ordinateurs les plus puissants, le problème est insoluble, c'est-à-dire qu'il n'existe aucun algorithme efficace connu pour le résoudre.

1. Mervyn O’Luining, Insight Centre for Data Analytics, Département d'informatique, Université College Cork, Irlande. Courriel : mervyn.oluining@insight-centre.org; Steven Prestwich, Insight Centre for Data Analytics, Département d'informatique, University College Cork, Irlande. Courriel : steven.prestwich@insight-centre.org; S. Armagan Tarim, Cork University Business School, University College Cork, Irlande. Courriel : armagan.tarim@ucc.ie.

De nombreux problèmes d’optimisation combinatoire à grande échelle de ce type ne peuvent être résolus de manière optimale, car la recherche d’une solution optimale exige un temps de calcul prohibitif. Cela oblige à utiliser des *algorithmes d’approximation* ou des algorithmes *heuristiques* qui ne garantissent pas de solutions optimales, mais peuvent fournir des solutions approximatives dans un intervalle de temps acceptable. De cette façon, on échange la qualité de la solution finale contre le temps de calcul (Van Laarhoven et Aarts, 1987). En d’autres termes, des algorithmes heuristiques sont créés pour trouver une solution « suffisamment bonne » en un temps de calcul « suffisamment court » (Sörensen et Glover, 2013).

Un certain nombre d’algorithmes heuristiques ont été créés pour rechercher des solutions optimales ou quasi optimales, pour les scénarios univariés et multivariés de ce problème. Il s’agit notamment de l’algorithme hiérarchique proposé par Benedetti et coll. (2008), de l’algorithme génétique proposé par Ballin et Barcaroli (2013) et de l’algorithme génétique de regroupement proposé par O’Luing, Prestwich et Tarim (2019). Bien qu’efficace, la fonction d’évaluation de ces algorithmes peut être coûteuse en termes de temps d’exécution.

Nous complétons ces travaux par un algorithme du recuit simulé (Kirkpatrick, Gelatt et Vecchi, 1983; Černý, 1985). On a constaté que les algorithmes du recuit simulé fonctionnent bien dans des problèmes comme celui-ci, où il existe de nombreux minima locaux et où la recherche d’une solution globale approximative en un temps de calcul fixe est plus souhaitable que la recherche d’un minimum local précis (Takeang et Aurasopon, 2019). Nous présentons un algorithme du recuit simulé auquel nous avons ajouté l’évaluation delta (voir section 5) pour tirer parti de la similitude entre des solutions consécutives et contribuer à accélérer les temps de calcul.

Nous avons comparé la performance des algorithmes du recuit simulé sur les strates atomiques avec celles de l’algorithme génétique de regroupement du module *SamplingStrata* (Ballin et Barcaroli, 2020). Cet algorithme met en œuvre les opérateurs de regroupement décrits par O’Luing et coll. (2019). Pour y arriver, nous avons utilisé des bases de sondage de tailles différentes contenant ce que nous supposons être des renseignements totalement représentatifs pour les colonnes de variables cibles et auxiliaires.

À la suite de la suggestion d’un réviseur de *Techniques d’enquête*, nous avons ensuite comparé l’algorithme du recuit simulé avec un algorithme génétique traditionnel utilisé par Ballin et Barcaroli (2020) sur des strates continues. Dans les deux séries d’expériences, nous avons utilisé une solution initiale créée par l’algorithme à K moyennes (Hartigan et Wong, 1979) dans un processus en deux étapes (voir la section 2.3 pour obtenir de plus amples précisions).

La section 2 fournit des renseignements généraux sur les strates atomiques, présente l’algorithme du recuit simulé et motive l’ajout de l’évaluation delta comme moyen d’améliorer le temps de calcul. Le recuit simulé en deux étapes est également abordé. La section 3 du présent document décrit la fonction de coût et l’algorithme d’évaluation. La section 4 donne un aperçu de l’algorithme du recuit simulé. La section 5 présente l’algorithme du recuit simulé amélioré avec évaluation delta. La section 6 présente une comparaison des performances de l’algorithme du recuit simulé avec l’algorithme génétique de regroupement au moyen d’une solution initiale et d’hyperparamètres peaufinés. La section 7 fournit

ensuite des précisions sur la comparaison de l'algorithme du recuit simulé avec l'algorithme génétique de Ballin et Barcaroli (2020) sur des strates continues. La section 8 présente les conclusions, et la section 9 propose des travaux supplémentaires. L'annexe contient des renseignements détaillés sur les contraintes de précision, les hyperparamètres et le peaufinage des hyperparamètres pour les deux comparaisons ainsi que les spécifications informatiques.

2. Renseignements généraux

2.1 Stratification des strates atomiques

Les strates atomiques sont créées à l'aide de colonnes de variables auxiliaires catégoriques, comme le *groupe d'âge*, le *sexe* ou l'*ethnicité* pour une enquête sur les personnes, ou encore, l'*industrie*, le *type d'entreprise* et la *taille de l'effectif* pour les enquêtes-entreprises. Le classement recoupé des intervalles de classe des colonnes de variables auxiliaires forme les strates atomiques.

Les colonnes de variables auxiliaires qui sont corrélées aux colonnes de variables cibles peuvent permettre un gain de précision de l'échantillon ou de *similarité*. Chaque colonne de variable cible, y_g , contient la valeur de la caractéristique de l'enquête qui nous intéresse, par exemple le *revenu total*, pour chaque élément de population de l'échantillon.

Une fois ces éléments créés, nous obtenons des statistiques sommaires, telles que la fréquence, la moyenne et l'écart-type des valeurs observées pertinentes, à partir d'une ou de plusieurs colonnes de variables cibles qui font partie de chaque strate atomique. Les données sommaires sont ensuite agrégées afin de calculer les moyennes et les variances pour chaque strate, qui sont à leur tour utilisées pour calculer la répartition de l'échantillon pour une stratification donnée.

Le partitionnement des strates atomiques qui fournit la répartition d'échantillon du *minimum absolu*, c'est-à-dire le minimum de toutes les répartitions d'échantillon possibles pour l'ensemble des stratifications possibles, est appelé une *stratification optimale*. Il peut y avoir une multitude de partitionnements de ce type. Bien qu'une stratification optimale soit *la* solution au problème, chaque stratification représente une solution de qualité variable (plus le coût est faible [répartition *minimale* ou *optimale* d'échantillon], plus la qualité est élevée). Pour chaque stratification, le coût est estimé par l'algorithme de Bethel-Chromy (Bethel, 1985, 1989; Chromy, 1987). Une description plus détaillée, et une analyse de la méthodologie de cette approche pour la détermination de la stratification et de la répartition de l'échantillon en combinaison, se trouvent dans Ballin et Barcaroli (2013).

2.2 Algorithmes du recuit simulé

Le principe de base de l'algorithme du recuit simulé (ARS) (Kirkpatrick et coll., 1983; Černý, 1985) est qu'il peut accepter des solutions inférieures à la meilleure solution actuelle afin de trouver les minima (ou maxima) absolus. C'est l'un de plusieurs algorithmes de recherche locale stochastique, qui concentrent

leur attention dans un voisinage local d’une solution initiale donnée (Cortez, 2014) et reposent sur différentes techniques stochastiques pour échapper aux minima locaux attrayants (Hoos et Stützle, 2004).

Basé sur le recuit physique en métallurgie, l’ARS est conçu pour simuler le processus de refroidissement contrôlé du métal liquide vers un état solide (Luke, 2013). Ce refroidissement contrôlé utilise le paramètre de température pour calculer la probabilité d’accepter des solutions inférieures (Cortez, 2014). Cette probabilité d’acceptation n’est pas seulement fonction de la température, mais aussi de la différence de coût entre la nouvelle solution et la meilleure solution actuelle. Pour une même différence de coût, une température plus élevée signifie une plus grande probabilité d’accepter des solutions de qualité inférieure.

Pour une température donnée, les solutions sont générées de manière itérative en appliquant une petite perturbation, générée de manière aléatoire, à la meilleure solution actuelle. En général, dans les ARS, une perturbation correspond au petit déplacement d’une particule choisie au hasard (Van Laarhoven et Aarts, 1987). Dans le contexte de notre problème, nous entendons par perturbation le déplacement (ou le repositionnement) de q strates atomiques (généralement $q = 1$) choisies au hasard d’une strate choisie au hasard à une autre.

En réponse à une perturbation, la meilleure solution actuelle passe à une nouvelle solution. Si une perturbation entraîne un coût moindre pour la nouvelle solution, ou si le coût ne change pas, cette solution est toujours sélectionnée comme la meilleure solution actuelle. Si la nouvelle solution entraîne un coût plus élevé, elle est acceptée avec la probabilité d’acceptation mentionnée ci-dessus. Cette condition d’acceptation est appelée le critère de Metropolis (Metropolis, Rosenbluth, Rosenbluth, Teller et Teller, 1953). Ce processus se poursuit jusqu’à la fin de la séquence, au moment où la température est décrétementée et où une nouvelle séquence commence.

Si les perturbations sont mineures, la solution actuelle et la nouvelle solution seront très similaires. En effet, dans notre ARS, nous supposons qu’il n’y a qu’une légère différence entre les solutions consécutives en raison de ces perturbations (voir la section 4.1 pour obtenir de plus amples précisions). C’est pourquoi nous avons ajouté l’évaluation delta, dont nous traiterons plus en détail dans la section 5, pour tirer parti de cette similitude et contribuer à améliorer les temps de calcul.

Par conséquent, et comme il est mentionné dans l’introduction, nous présentons un ARS avec évaluation delta et le comparons avec l’algorithme génétique de regroupement (AGR) lorsque les deux sont combinés avec une solution initiale. Nous le comparons également avec un algorithme génétique utilisé par Ballin et Barcaroli (2020) sur des strates continues. Nous donnons plus de renseignements sur les solutions initiales dans la section 2.3 ci-dessous.

2.3 Recuit simulé en deux étapes

Un processus de recuit simulé en deux étapes, dans lequel une solution initiale est générée par un algorithme heuristique lors de la première étape, a été proposé pour des problèmes tels que le *problème de placement de cellules* (Grover, 1987; Rose, Snelgrove et Vranesic, 1988) ou le *problème de*

partitionnement de graphes (Johnson, Aragon, McGeoch et Schevon, 1989). Lisic, Sang, Zhu et Zimmer (2018) ont combiné une solution initiale, générée par l'algorithme à K moyennes, avec un algorithme du recuit simulé, pour un problème de nature semblable à celui-ci, mais où la répartition de l'échantillon ainsi que le nombre de strates sont fixes, et où l'algorithme recherche l'arrangement optimal des unités d'échantillonnage entre les strates.

L'algorithme du recuit simulé utilisé par Lisic et coll. (2018) commence par une solution initiale (stratification et répartition d'échantillons à chaque strate) et, pour chaque itération, génère une nouvelle solution candidate en déplaçant une strate atomique d'une strate à une autre et en ajustant la répartition de l'échantillon pour cette stratification. Chaque solution candidate est ensuite évaluée pour mesurer le coefficient de variation (CV) des variables cibles et est acceptée, en tant que nouvelle meilleure solution actuelle, si sa fonction objective est inférieure à la solution précédente. Les solutions de qualité inférieure sont également acceptées avec une probabilité, ρ , qui est fonction d'un paramètre de température réglable et du changement de qualité de la solution entre les itérations. La température se refroidit, à une vitesse qui est également réglable, au fur et à mesure que le nombre d'itérations augmente.

À la suite de ces travaux, Ballin et Barcaroli (2020) ont recommandé de combiner une solution initiale, générée par l'algorithme à K moyennes, avec le regroupement et les algorithmes génétiques traditionnels. Ils démontrent que l'algorithme à K moyennes fournit de meilleures solutions de départ par rapport à la solution de départ générée par une approche stochastique. Nous combinons également une solution initiale de type K moyennes avec l'ARS dans les expériences décrites dans les sections 6 et 7.

3. Le problème de stratification et de répartition simultanée de l'échantillon

Notre objectif est de partitionner des L strates atomiques en H sous-populations ou strates non vides. Un partitionnement représente une stratification de la population. Nous cherchons à limiter la répartition de l'échantillon à cette stratification tout en maintenant la mesure de similarité inférieure ou égale à la limite supérieure de précision, ε_g . Cette similarité est mesurée par le CV du total de la population estimée pour chacune des G colonnes de variables cibles, \hat{T}_g . Nous indiquons par n_h l'échantillon attribué à la strate h et le coût de l'enquête pour une stratification donnée est calculé comme suit :

$$C(n_1, \dots, n_H) = \sum_{h=1}^H C_h n_h$$

où C_h est le coût moyen de l'enquête sur une unité dans la strate h et n_h est la répartition de l'échantillon à la strate h . Dans notre analyse, C_h est fixé à 1.

La variance de l'estimateur est donnée par :

$$\text{VAR}(\hat{T}_g) = \sum_{h=1}^H N_h^2 \left(1 - \frac{n_h}{N_h}\right) \frac{S_{h,g}^2}{n_h} \quad (g=1, \dots, G)$$

où N_h est le nombre d’unités dans la strate h et $S_{h,g}^2$ est la variance de la strate h pour chaque colonne de variables cibles g .

Comme il est mentionné ci-dessus, ε_g est la limite supérieure de précision du CV pour chaque \hat{T}_g :

$$CV(\hat{T}_g) = \frac{\sqrt{\text{VAR}(\hat{T}_g)}}{E(\hat{T}_g)} \leq \varepsilon_g.$$

Le problème peut être résumé de la manière suivante :

$$\begin{aligned} \min \quad & n = \sum_{h=1}^H n_h \\ \text{sous la contrainte} \quad & CV(\hat{T}_g) \leq \varepsilon_g \quad (g = 1, \dots, G). \end{aligned}$$

Pour résoudre le problème de répartition pour une stratification particulière avec l’algorithme de Bethel-Chromy, la contrainte de précision supérieure pour la variable g peut être exprimée comme suit :

$$\begin{aligned} CV(\hat{T}_g)^2 \leq \varepsilon_g^2 & \equiv \sum_{h=1}^H \frac{N_h^2 S_{h,g}^2}{n_h} - N_h S_{h,g}^2 \leq E(\hat{T}_g^2) \varepsilon_g^2 \\ & \equiv \sum_{h=1}^H \frac{N_h^2 S_{h,g}^2}{\left(E(\hat{T}_g^2) \varepsilon_g^2 + \sum_{h=1}^H N_h S_{h,g}^2 \right) n_h} \leq 1. \end{aligned}$$

Ensuite, nous substituons

$$\frac{N_h^2 S_{h,g}^2}{\left(E(\hat{T}_g^2) \varepsilon_g^2 + \sum_{h=1}^H N_h S_{h,g}^2 \right)}$$

à ξ_h, g et remplaçons le résumé du problème par le suivant :

$$\begin{aligned} \min \quad & n = \sum_{h=1}^H n_h \\ & \sum_{h=1}^H \frac{\xi_h, g}{n_h} \leq 1 \quad (g = 1, \dots, G) \end{aligned}$$

où $\frac{1}{n_h} > 0$. Dans l’algorithme de Bethel-Chromy, des multiplicateurs de Lagrange sont utilisés pour dériver une solution pour chaque n_h .

$$\frac{1}{n_h} = \begin{cases} \frac{\sqrt{1}}{\left(\sqrt{\sum_{g=1}^G \alpha_g \xi_h, g} \sum_{h=1}^H \sqrt{\sum_{g=1}^G \alpha_g \xi_h, g} \right)} & \text{si } \sum_{g=1}^G \alpha_g \xi_h, g > 0 \\ +\infty & \text{autrement} \end{cases}$$

où

$$\alpha_g = \frac{\lambda_g}{\sum_{g=1}^G \lambda_g},$$

et λ_g est le multiplicateur de Lagrange (Benedetti et coll., 2008). L'algorithme commence par un paramètre par défaut pour chaque α_g et utilise la descente de gradient pour converger vers une valeur finale pour ceux-ci.

4. Aperçu de l'algorithme du recuit simulé

L'ARS avec évaluation delta est décrit dans l'algorithme 1 ci-dessous. Nous décrivons ensuite les algorithmes heuristiques que nous avons utilisées dans l'ARS. L'évaluation delta est expliquée plus en détail dans la section 5.

Algorithme 1 Algorithme du recuit simulé

Function SIMULATEDANNEALING (S is the starting solution, f is the evaluation function (Bethel-Chromy algorithm), $best$ is the current best solution, $BSFSF$ is the best solution found so far, $maxit$ is the maximum number of sequences, J is the length of sequence, T_{max} is the starting temperature, T_{min} is the minimum temperature, DC is the Decrement Constant, $L_{max\%}$ is a % of L (number of atomic strata), $P(H+1)$ is the probability of a new stratum, $H+1$, being added)

```

T ← Tmax
best ← S
Cost(best) ← f(best)
while i < maxit && T > Tmin do
  if RANDOM(0,1) ≤ 1/J then
    for l=1 to L do
      if RANDOM(0,1) ≤ P(H+1) then
        move atomic stratum l to new stratum H+1
      end if
    end for
    end if
  for j=1 to J do
    if i=1 & j=1 then q = L × Lmax%
    else if i=1 & j>1 then q = ceiling(q × 0.99)
    else if i>1 then q = 1
    end if
    Randomly select h and h'
    next ← PERTURBATION(best)

    Cost(next) ← f(next)
    ΔE ← COST(next) - COST(best)
    if ΔE ≤ 0 then
      best ← next
    else if RANDOM(0,1) < e(-ΔE/T) then
      best ← next
    end if
    if best <= BSFSF then
      BSFSF ← best
    end if
  end for
  T ← T * DC
end while
return BSFSF
end function

```

► en utilisant l'algorithme de Bethel-Chromy

► voir la section 4.3

► 0,99 n'est pas réglable

► Assigner q strates atomiques de h et h'

► en utilisant l'évaluation delta

► Critère de Metropolis

4.1 Perturbation

Considérons la solution suivante représentée par la stratification :

$$\{1, 3\}, \{2\}, \{4, 5, 6\}.$$

Les nombres entiers dans chaque strate représentent les strates atomiques. Dans la perturbation, la nouvelle solution ci-dessous est créée en déplaçant arbitrairement des strates atomiques, dans cet exemple $q = 1$, d’une strate choisie au hasard à une autre.

$$\{1, 3, 2\}, \{\emptyset\}, \{4, 5, 6\}.$$

La première strate gagne une strate atomique supplémentaire $\{2\}$ pour devenir $\{1, 3, 2\}$, tandis que la strate intermédiaire ou deuxième strate a été « vidée » (et est supprimée), et il ne reste que deux strates. Les strates ne sont vidées que lorsque la dernière strate atomique restante a été déplacée vers une autre strate.

Pour clarifier la façon dont cela fonctionne dans l’algorithme : chaque solution est représentée par un vecteur de nombres entiers, les strates atomiques qui ont le même nombre entier sont dans la même strate. Un vecteur séparé des nombres entiers uniques de la solution représente les strates. Par exemple, la première solution $\{1, 3\}, \{2\}, \{4, 5, 6\}$ serait représentée par le vecteur $[1 \ 2 \ 1 \ 3 \ 3 \ 3]$ et les strates seraient représentées par le vecteur $[1 \ 2 \ 3]$. Lorsque la nouvelle solution est créée, la deuxième strate a été supprimée et ne fait plus partie de la solution. Autrement dit, le vecteur de la nouvelle solution est : $[1 \ 1 \ 1 \ 3 \ 3 \ 3]$ et le vecteur des strates est $[1 \ 3]$. Compte tenu de la suppression de la strate 2, et pour plus de clarté, nous renommons la strate 3 en 2 de sorte que cette solution devient : $[1 \ 1 \ 1 \ 2 \ 2 \ 2]$ et les strates sont maintenant représentées par le vecteur $[1 \ 2]$. La strate $[1 \ 2]$ restera dans toutes les solutions ultérieures, sauf si une autre strate est « vidée » ou si une nouvelle strate est ajoutée.

4.2 Évaluation et acceptation

Chaque nouvelle solution est évaluée à l’aide de l’algorithme de Bethel-Chromy, et le critère d’acceptation de Metropolis est appliqué. Si elle est acceptée, la nouvelle solution ne diffère de la solution précédente que par la perturbation mentionnée ci-dessus. Si elle n’est pas acceptée, nous continuons avec la solution précédente, et essayons de nouveau de déplacer q strates atomiques choisies au hasard entre deux strates choisies au hasard.

4.3 Séquences et nouvelles strates

Cela se poursuit pour la longueur réglable de la séquence, J . Cette durée doit être suffisante pour permettre à la séquence d’atteindre l’équilibre. Cependant, il n’existe pas de règle pour déterminer J . Au début de chaque nouvelle séquence, nous avons H strates dans la meilleure solution actuelle. Avec une probabilité fixe de $1/J$, une strate supplémentaire est ajoutée. Si une nouvelle strate doit être ajoutée,

l'ARS parcourt en boucle chaque strate atomique et la déplace vers une nouvelle strate, appelée $H + 1$, car chaque strate est étiquetée séquentiellement de 1 à H (voir section 4.1), avec une probabilité réglable, $P(H + 1)$. L'algorithme fonctionne pour un nombre réglable de séquences, *maxit*.

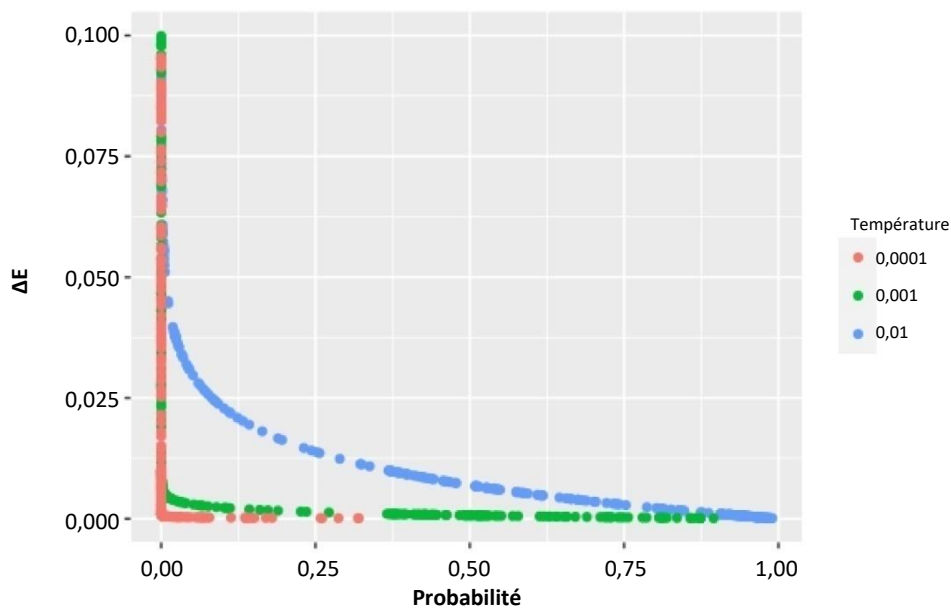
4.4 Température

La température est décrétementée depuis une température de départ, T_{\max} , jusqu'à une température minimale, T_{\min} , ou jusqu'à ce que *maxit* soit atteint. Comme nous partons d'une solution quasi optimale, nous choisissons que T_{\max} ne soit pas supérieure à 0,01 et nous définissons T_{\min} comme étant $1,0 \times 10^{-11}$.

Cela permet de tenir compte de la nature avancée de la recherche et permet à l'algorithme de mettre davantage l'accent sur la recherche de solutions supérieures, avec une probabilité toujours plus faible d'accepter des solutions inférieures. Cependant, une température basse, T , n'équivaut pas toujours à une faible probabilité d'acceptation.

De petites différences positives dans la qualité de la solution (où la nouvelle solution a une qualité légèrement inférieure à la meilleure solution actuelle), ΔE , se produisent souvent parce que nous commençons par une solution initiale de bonne qualité. La figure 4.1 montre que la probabilité que de telles solutions soient acceptées, $e^{(-\frac{\Delta E}{T})}$, augmente au fur et à mesure que cette différence se réduit pour le même T . Néanmoins, la figure 4.1 montre également que, pour les mêmes changements dans la qualité de la solution, lorsque T diminue, la probabilité diminue également (et il se comporte de plus en plus comme un algorithme d'escalade).

Figure 4.1 Probabilité d'accepter une solution inférieure en fonction de ΔE et T .



5. Amélioration des performances de l’algorithme du recuit simulé à l’aide de l’évaluation delta

Comme il est indiqué plus haut, la seule différence entre les solutions consécutives est que q strates atomiques ont été déplacées d’un groupe à l’autre. Comme pour les autres algorithmes heuristiques, q est également réglable, et pour la première séquence nous avons ajouté l’option de fixer $q > 1$ et de réduire q pour chaque nouvelle solution de la première séquence jusqu’à $q = 1$. La raison en est que, lorsque $q > 1$, la taille accrue de la perturbation peut contribuer à réduire le nombre de strates. Dans ce cas, nous définissons q comme étant un pourcentage réglable de la taille de la solution, ou du nombre de strates atomiques, L , à partitionner. Après la première séquence $q = 1$.

En outre, comme les strates s’excluent mutuellement, ce déplacement de q strates atomiques d’une strate à l’autre n’a aucune incidence sur les strates restantes. Ross, Corne et Fang (1994) présentent une technique appelée évaluation delta, dans laquelle l’évaluation d’une nouvelle solution repose sur des solutions similaires évaluées précédemment, afin d’accélérer de manière importante les expériences d’algorithmes évolutionnaires ou d’échéanciers. Nous utilisons les propriétés similaires de deux solutions consécutives pour appliquer l’évaluation delta à l’ARS. Il s’ensuit donc que, dans la première séquence, q doit être maintenu à un niveau bas et que la réduction à $q = 1$ doit être rapide.

L’algorithme de Bethel-Chromy nécessite l’établissement des moyennes et des variances pour chaque strate afin de calculer la répartition de l’échantillon. Cependant, nous utilisons les informations déjà calculées pour les strates restantes $H - 2$, et nous nous contentons d’effectuer les calculs pour les deux strates concernées par la perturbation. Ainsi, le calcul des moyennes et des variances des strates de H est réduit à un simple sous-ensemble de ce qui serait autrement nécessaire.

Rappelons maintenant que l’algorithme de Bethel-Chromy commence par une valeur par défaut pour chaque α_g et utilise la descente de gradient pour trouver une valeur finale pour chaque α_g . Cette recherche se poursuit jusqu’à ce que l’algorithme atteigne un seuil de taille de pas minimum, ou bien dépasse un nombre maximal d’itérations. Ce seuil minimal est caractérisé par ϵ , qui est fixé à $1,0 \times 10^{-11}$ dans Ballin et Barcaroli (2020), et le nombre maximal d’itérations est de 200. Nous supposons que cette recherche sera considérablement réduite si nous utilisons les valeurs α_g de l’évaluation de la solution actuelle comme point de départ pour la solution suivante.

Les deux mises en œuvre ci-dessus de l’évaluation delta entraînent une réduction notable des temps de calcul, comme le montrent les expériences décrites ci-dessous.

6. Comparaison des performances des deux algorithmes

6.1 Plan d’évaluation

Dans cette section, nous présentons la comparaison des performances de l’algorithme génétique de regroupement avec l’algorithme du recuit simulé. Nous avons utilisé un certain nombre d’ensembles de

données de tailles différentes dans ces expériences. Chaque ensemble de données comporte un certain nombre de régions (appelées ici « domaines »). Une stratification optimale et une répartition minimale de l'échantillon ont été sélectionnées pour chaque domaine.

La somme des échantillons pour tous les domaines donne la taille totale de l'échantillon. La taille de l'échantillon ou le coût de la solution définit la qualité de la solution. Pour obtenir de plus amples précisions sur les domaines, voir Ballin et Barcaroli (2013). Le but de ces expériences était d'examiner si l'ARS pouvait atteindre une qualité de solution comparable à celle de l'AGR en moins de temps de calcul par solution, ce qui se traduit par des économies de temps d'exécution.

Cependant, nous avons également comparé les temps d'exécution totaux, car cela est une conséquence de la nécessité d'entraîner les hyperparamètres pour les deux algorithmes. Plus de renseignements sont fournis dans l'annexe.

Nous présentons les résultats de ces expériences dans la section 6.4 où, aux fins de comparaison, nous exprimons les résultats de l'ARS en tant que ratio de ceux de l'AGR.

6.2 Comparaison du nombre de solutions générées

Après la première itération, l'AGR conserve les solutions d'élite, E , de l'itération précédente. Elles sont calculées par le produit du taux d'élitisme (la proportion de la population de chromosomes qui sont des solutions élites), E_R , et de la taille de la population de chromosomes (le nombre de solutions candidates à chaque itération), N_p . Comme E ont déjà été évaluées, elles ne sont pas évaluées de nouveau.

Pour cette raison, nous avons comparé les temps d'évaluation des solutions évaluées dans l'AGR avec tous ceux de l'ARS. Pour l'AGR, le nombre total de solutions évaluées, N_{solAGR} , est fonction du nombre de domaines, D , de la taille de la population de chromosomes, des solutions non élites (calculées par le produit de $1 - E_R$ et N_p) et du nombre d'itérations, I . Pour obtenir de plus amples précisions sur la mise en œuvre de l'AGR (par exemple, les solutions d'élite, le taux d'élitisme, la population de chromosomes), nous renvoyons le lecteur à (Falkenauer, 1998)

$$N_{\text{solAGR}} = \left(D \times \left(N_p + \left(N_p \times (1 - E_R) \times (I - 1) \right) \right) \right).$$

Dans le cas de l'algorithme du recuit simulé, le nombre maximal de solutions, N_{solARS} , est égal au nombre de domaines, D , multiplié par le nombre de séquences, $maxit$, et par la longueur de la séquence, J . Rappelons que l'ARS s'arrête également si la température minimale a été atteinte; c'est pourquoi nous nous référons au *nombre maximal* de solutions plutôt qu'au *total*. Cependant, aux fins de comparaison, étant donné que la température n'est décrétementée qu'à la fin de chaque séquence et que nous avons un petit nombre de séquences dans les expériences ci-dessous, nous supposons que le nombre total de solutions a été généré

$$N_{\text{solARS}} = D \cdot maxit \cdot J.$$

6.3 Ensembles de données, variables cibles et auxiliaires

Le tableau 6.1 fournit un résumé par ensemble de données des variables cibles et auxiliaires.

Tableau 6.1
Résumé par ensemble de données des variables cibles et auxiliaires

Ensemble de données	Variables cibles	Description	Variables auxiliaires	Description
Municipalités suisses	Surfacebois	Zone boisée	POPTOT	Population totale
	Airbat	Zone comportant des bâtiments	Hapoly	Zone de la municipalité
American Community Survey, 2015	HINCP	Revenu du ménage, 12 derniers mois	BLD	Unités dans la structure
	VALP	Valeur de la propriété	TEN	Tenure
	SMOCP	Coûts mensuels sélectionnés du propriétaire	WKEXREL	Expérience professionnelle du chef de famille et du conjoint
	INSP	Montant annuel de l’assurance contre l’incendie, les risques ou les inondations	WORKSTAT	Situation professionnelle du membre responsable du ménage ou du conjoint dans les ménages familiaux
HFL			Combustible de chauffage domestique	
YBL			Date de construction initiale de la structure	
Recensement des États-Unis, 2000	HHINCOME	Revenu total du ménage	PROPINSR	Coût annuel de l’assurance immobilière
			COSTFUEL	Coût annuel du combustible de chauffage domestique
			COSTELEC	Coût annuel de l’électricité
			VALUEH	Valeur de la maison
Prêts Kiva	term_in_months	Durée pour laquelle le prêt a été déboursé	sector	Catégories de haut niveau, par exemple les aliments
	lender_count	Nombre total de prêteurs	currency	Devise du prêt
	loan	Montant en \$ US	activity	Catégorie plus granulaire, par exemple fruits et légumes
			region	Nom de la région dans le pays
			partner_id	Identifiant de l’organisation partenaire
Statistiques sur le commerce des produits de base de l’Organisation des Nations Unies	trade_usd	Valeur de la transaction en \$ US	commodity	Type de produit de base, par exemple « Chevaux, vivants, sauf élevage de race pure »
			flow	Si la marchandise était une importation, une exportation, une réimportation ou une réexportation
			category	Catégorie de produit, par exemple la soie ou les engrais

Les variables cibles et auxiliaires pour l’ensemble de données des municipalités suisses ont été sélectionnées sur la base de l’expérience décrite dans Ballin et Barcaroli (2020). En conséquence, *POPTOT* et *HApoly* ont été convertis en variables catégorielles à l’aide de l’algorithme de classification à K moyennes. Cependant, nous avons utilisé plus de domaines et d’itérations dans notre expérience. De plus amples renseignements sur cet ensemble de données sont fournis par Barcaroli (2014).

Pour les autres expériences, nous avons sélectionné des variables cibles et auxiliaires que nous avons jugées susceptibles d'intéresser les concepteurs d'enquêtes. Des renseignements supplémentaires sur les métadonnées de l'American Community Survey, 2015 (U.S. Census Bureau, 2016), du recensement des États-Unis de 2000 (Ruggles, Genadek, Goeken, Grover et Sobek, 2017), de Prêts Kiva (Kiva, 2018), et des données statistiques de l'Organisation des Nations Unies sur le commerce des produits de base (Nations Unies, 2017) figurent dans O'Luing et coll. (2019).

Un autre résumé par ensemble de données du nombre d'enregistrements et de strates atomiques ainsi qu'une description de la variable de domaine sont fournis dans le tableau 6.2 ci-dessous.

Tableau 6.2
Résumé par ensemble de données du nombre d'enregistrements et de strates atomiques et description de la variable de domaine

Ensemble de données	Nombre d'enregistrements	Nombre de strates atomiques, L	Variable de domaine
Municipalités suisses	2 896	579	REG
American Community Survey, 2015	619 747	123 007	ST (les 51 États)
Recensement des États-Unis, 2000	627 611	517 632	Région
Prêts Kiva	614 361	84 897	Code de pays
Statistiques sur le commerce des produits de base de l'Organisation des Nations Unies	352 078	351 916	Pays ou région

6.4 Résultats

Comme il est mentionné précédemment, nous avons utilisé dans chaque expérience une solution initiale créée par l'algorithme *KmeansSolution* (Ballin et Barcaroli, 2020). Nous avons ensuite comparé les performances des algorithmes en ce qui a trait au temps de calcul moyen (en secondes) par solution et à la qualité des solutions. Le tableau 6.3 présente la taille de l'échantillon, les temps d'exécution et les temps d'exécution totaux pour l'ARS et l'AGR.

Tableau 6.3
Résumé par ensemble de données de la taille de l'échantillon et du temps d'évaluation pour l'algorithme génétique de regroupement et l'algorithme du recuit simulé

Ensemble de données	AGR			ARS		
	Taille de l'échantillon	Temps d'exécution (secondes)	Temps d'exécution total (secondes)	Taille de l'échantillon	Temps d'exécution (secondes)	Temps d'exécution total (secondes)
Municipalités suisses	128,69	753,82	10 434,30	125,17	248,91	8 808,63
American Community Survey, 2015	10 136,50	13 146,25	182 152,46	10 279,44	517,76	6 822,42
Recensement des États-Unis, 2000	228,81	2 367,36	36 298,35	224,75	741,75	8 996,85
Prêts Kiva	6 756,19	15 669,11	288 946,79	6 646,67	664,30	7 549,87
Statistiques sur le commerce des produits de base de l'Organisation des Nations Unies	3 216,68	6 535,97	88 459,22	3 120,07	1 169,26	12 161,80

Le temps d’exécution total est la somme des temps d’exécution de 20 évaluations des algorithmes AGR et ARS (par la fonction MBO (optimisation basée sur le modèle) du progiciel R *mlrMBO* (Bischla, Richter, Bossek, Hornb, Thomas et Langb, 2017)) en utilisant 20 ensembles d’hyperparamètres sélectionnés (c’est-à-dire un ensemble pour chaque évaluation). Les renseignements détaillés sur les contraintes de précision et les hyperparamètres pour chaque expérience se trouvent dans l’annexe. Le tableau 6.4 exprime les résultats de l’ARS sous la forme d’un rapport avec ceux de l’AGR.

Tableau 6.4
Comparaison de la taille des échantillons, des temps d’exécution et des temps d’exécution totaux pour l’algorithme génétique de regroupement et l’algorithme du recuit simulé

Ensemble de données	Taille de l’échantillon	Temps d’exécution (secondes)	Temps d’exécution total (secondes)
Municipalités suisses	0,97	0,33	0,84
American Community Survey, 2015	1,01	0,04	0,04
Recensement des États-Unis, 2000	0,98	0,31	0,25
Prêts Kiva	0,98	0,04	0,03
Statistiques sur le commerce des produits de base de l’Organisation des Nations Unies	0,97	0,18	0,14

Comme nous pouvons le constater, la taille des échantillons est similaire, mais l’ARS présente des temps d’exécution et des temps d’exécution totaux nettement plus courts. Lorsque ces expériences sont exécutées en parallèle, dans les cas où il y a un grand nombre de domaines, il se peut qu’il n’y ait pas assez de cœurs pour traiter tous les domaines en une seule exécution. En effet, il peut falloir plusieurs exécutions parallèles pour accomplir la tâche, ce qui aura une incidence sur le temps d’évaluation moyen. Les spécifications de l’ordinateur sont indiquées dans le tableau A.2. Le tableau 6.5 indique le nombre de solutions évaluées par chaque algorithme pour obtenir les résultats présentés au tableau 6.3. Il permet également de comparer le temps d’exécution moyen (en secondes) par solution.

Tableau 6.5
Comparaison du nombre de solutions et du ratio du temps d’exécution (par seconde) entre l’algorithme génétique de regroupement et l’algorithme du recuit simulé

Ensemble de données	Nombre de solutions évaluées		Temps d’exécution moyen par solution (secondes)		
	AGR	ARS	AGR	ARS	Proportion
Municipalités suisses	840 140	210 000	0,0009	0,0012	1,3210
American Community Survey, 2015	2 550 510	459 000	0,0052	0,0011	0,2188
Recensement des États-Unis, 2000	10 872	36 000	0,2177	0,0206	0,0946
Prêts Kiva	2 190 730	730 000	0,0072	0,0009	0,1272
Statistiques sur le commerce des produits de base de l’Organisation des Nations Unies	2 395 026	1 539 000	0,0027	0,0008	0,2784

Les résultats ci-dessus indiquent que l’AGR a permis d’évaluer plus de solutions pour trouver une solution de qualité comparable à celle de l’ARS dans tous les cas, sauf pour l’expérience du *recensement des États-Unis de 2000*. Cependant, nous pouvons également constater que l’ARS prend moins de temps

pour évaluer chaque solution dans tous les cas, sauf pour l'expérience des *municipalités suisses*. Le temps d'exécution moyen pour chaque expérience peut être considéré dans le contexte de la taille de l'ensemble de données, de la parallélisation et des ensembles particuliers d'hyperparamètres utilisés pour l'AGR et l'ARS. À cela s'ajoute la mémorisation de l'algorithme d'évaluation pour l'AGR, et des gains obtenus par évaluation delta par l'ARS.

Les gains sont plus perceptibles pour les grands ensembles de données, en raison de la taille de la solution et du nombre de strates atomiques dans chaque strate. À mesure que la taille des strates augmente, le déplacement de q d'une strate à l'autre (où q est petit) aura une incidence moindre sur la qualité de la solution et, par conséquent, l'évaluation delta sera plus rapide.

7. Comparaison avec la méthode continue dans *SamplingStrata*

Nous avons également comparé l'ARS avec l'algorithme génétique traditionnel que Ballin et Barcaroli (2020) ont appliqué pour partitionner des strates continues. Nous avons utilisé les variables cibles décrites dans le tableau 6.1 ci-dessus comme variables cibles et auxiliaires continues (par souci de clarté, nous les décrivons de nouveau dans le tableau 7.1 ci-dessous) ainsi que les contraintes de précision décrites dans le tableau A.1 (l'annexe). Dans la pratique, la variable cible ne sera pas exactement égale à la variable auxiliaire, bien qu'il soit courant que la variable auxiliaire soit une version imparfaite (par exemple une variable obsolète ou apparentée) disponible dans la base de sondage. Nous invitons le lecteur à en tenir compte lorsqu'il examinera les résultats des comparaisons ci-dessous. Il convient également de mentionner que les solutions initiales ont été créées pour les deux algorithmes en utilisant la méthode à K moyennes. Des renseignements détaillés sur la formation des hyperparamètres pour ces expériences figurent également dans l'annexe.

Tableau 7.1

Résumé par ensemble de données de la description des variables cibles et auxiliaires pour la méthode continue

Ensemble de données	Variables cibles	Variables auxiliaires	Description
Municipalités suisses	Surfacebois	Surfacebois	Zone boisée
	Airbat	Airbat	Zone comportant des bâtiments
American Community Survey, 2015	HINCP	HINCP	Revenu du ménage (12 derniers mois)
	VALP	VALP	Valeur de la propriété
	SMOCP	SMOCP	Coûts mensuels sélectionnés du propriétaire
	INSP	INSP	Montant annuel de l'assurance contre l'incendie, les risques ou les inondations
Recensement des États-Unis, 2000	HHINCOME	HHINCOME	Revenu total du ménage
Prêts Kiva	term_in_months	term_in_months	Durée pour laquelle le prêt a été déboursé
	lender_count	lender_count	Nombre total de prêteurs
	loan	loan	Montant en \$ US
Statistiques sur le commerce des produits de base de l'Organisation des Nations Unies	trade_usd	trade_usd	Valeur de la transaction en \$ US

Les tailles d’échantillon atteintes sont comparées dans le tableau 7.2 ci-dessous où la taille de l’échantillon pour l’ARS est exprimée comme un ratio de l’AGT. Après le peaufinage des hyperparamètres (voir section A.6), les tailles d’échantillon résultantes sont comparables.

Tableau 7.2
Comparaison de la taille des échantillons pour l’algorithme génétique traditionnel et l’algorithme du recuit simulé sur la méthode continue

Ensemble de données	AGT	ARS	Ratio
Municipalités suisses	128,69	120,00	0,93
American Community Survey, 2015	4 197,68	3 915,48	0,93
Recensement des États-Unis, 2000	192,71	179,89	0,93
Prêts Kiva	3 062,33	3 017,79	0,99
Statistiques sur le commerce des produits de base de l’Organisation des Nations Unies	3 619,42	3 258,52	0,90

Le tableau 7.3 vise à comparer les temps d’exécution pour l’ensemble des hyperparamètres qui ont permis de trouver les tailles d’échantillon pour chaque algorithme dans le tableau 7.2 ci-dessus, ainsi que les temps d’exécution totaux pris pour entraîner cet ensemble d’hyperparamètres.

Tableau 7.3
Comparaison des ratios des temps d’exécution et des temps d’exécution totaux pour l’algorithme génétique traditionnel et l’algorithme du recuit simulé selon la méthode continue

Ensemble de données	AGT		ARS		Ratio	
	Temps d’exécution (secondes)	Temps d’exécution total (secondes)	Temps d’exécution (secondes)	Temps d’exécution total (secondes)	Temps d’exécution (secondes)	Temps d’exécution total (secondes)
Municipalités suisses	753,82	10 434,30	213,44	1 905,82	0,28	0,18
American Community Survey, 2015	22 016,95	227 635,51	13 351,19	169 115,92	0,61	0,74
Recensement des États-Unis, 2000	3 361,90	46 801,78	51,94	1 147,36	0,02	0,02
Prêts Kiva	3 232,78	48 746,61	300,16	4 149,06	0,09	0,09
Statistiques sur le commerce des produits de base de l’Organisation des Nations Unies	29 045,23	326 931,63	73,18	1 287,38	0,003	0,004

Ces résultats indiquent un temps d’exécution nettement plus court pour l’ARS en ce qui concerne la qualité de la solution atteinte. L’efficacité de calcul gagnée par l’évaluation delta dans l’apprentissage des hyperparamètres recommandés est également évidente dans les temps d’exécution totaux. Pour l’expérience *American Community Survey, 2015*, beaucoup plus de solutions ont été générées par l’ARS que par l’AGT en raison des hyperparamètres donnés, ce qui a une incidence sur les temps d’exécution et le temps d’exécution total (voir également le tableau 7.4). Le tableau 7.4 vise à comparer le nombre de solutions générées par l’algorithme génétique traditionnel et l’algorithme du recuit simulé.

Tableau 7.4
Comparaison du nombre de solutions générées par l'algorithme génétique traditionnel et l'algorithme du recuit simulé selon la méthode continue

Ensemble de données	Nombre de solutions évaluées	
	AGT	ARS
Municipalités suisses	840 140	175 000
American Community Survey, 2015	918 102	5 100 000
Recensement des États-Unis, 2000	43 272	18 000
Prêts Kiva	146 730	292 000
Statistiques sur le commerce des produits de base de l'Organisation des Nations Unies	20 521 026	85 500

Dans tous les cas, à l'exception de *Prêts Kiva* et de *American Community Survey, 2015*, l'ARS a généré moins de solutions. Le faible nombre de solutions générées par les deux algorithmes pour l'expérience du *recensement des États-Unis de 2000* peut indiquer que la solution à K moyennes initiale était proche du minimum absolu. Les résultats de l'enquête *American Community Survey, 2015* indiquent que l'ARS a généré beaucoup plus de solutions pour arriver à une taille d'échantillon comparable à celle de l'AGT. Comme nous déplaçons, principalement, $q = 1$ strates atomiques d'une strate à l'autre, de tels changements ont eu, dans ce cas, une incidence limitée sur la qualité de la solution d'une solution à l'autre. Cependant, les gains obtenus par l'évaluation delta signifiaient qu'un plus grand nombre de solutions étaient évaluées par seconde, ce qui conduisait à une recherche plus complète et à l'obtention d'une taille d'échantillon plus faible.

Pour ces expériences, l'AGT a pris plus de temps pour trouver une taille d'échantillon comparable dans tous les cas. Comme le soulignent O'Luing et coll. (2019), les algorithmes génétiques traditionnels ne sont pas aussi efficaces pour les problèmes de regroupement que l'algorithme génétique de regroupement, car les solutions ont tendance à comporter beaucoup de redondance. Nous proposons donc que l'AGR soit également appliqué aux strates continues. Sur la base de l'analyse ci-dessus, et des performances des ARS en matière de recherche locale de manière générale, ainsi que des gains d'efficacité supplémentaires attribuables à l'évaluation delta, nous proposons également que l'ARS soit considéré comme une solution de rechange à l'algorithme génétique traditionnel.

8. Conclusions

Nous avons comparé l'ARS avec l'AGR dans le cas de strates atomiques et avec l'AGT dans le cas de strates continues (Ballin et Barcaroli, 2020). L'algorithme à K moyennes a fourni de bons points de départ dans tous les cas. Après le peaufinage des hyperparamètres, tous les algorithmes obtiennent des résultats de qualité similaire.

Cependant, les temps d'exécution pour les hyperparamètres recommandés sont plus courts pour l'ARS que pour l'AGR en ce qui concerne les strates atomiques et pour l'algorithme génétique traditionnel en ce qui concerne les strates continues. L'évaluation delta présente également l'avantage de réduire les temps d'apprentissage nécessaires pour trouver les hyperparamètres appropriés pour l'ARS.

L’AGR pourrait tirer avantage d’une extension en un algorithme mémétique en s’appuyant sur la recherche locale pour améliorer rapidement un chromosome avant de l’ajouter à la population de chromosomes de l’AGR.

L’ARS, en s’appuyant sur la recherche locale (avec une acceptation probabiliste des solutions inférieures), est bien adapté à la navigation hors des minima locaux, et la mise en œuvre de l’évaluation delta permet une recherche plus complète du voisinage local que ce qui serait possible dans le même temps de calcul.

9. Autres travaux

La perturbation utilisée par l’ARS déplace aléatoirement les strates atomiques q , où principalement $q = 1$, d’une strate à l’autre. Ce processus stochastique est standard dans les algorithmes du recuit simulé par défaut. Cependant, comme nous utilisons une solution de départ où il existe déjà une similarité au sein des strates, ce processus aléatoire pourrait facilement déplacer une strate atomique ($q = 1$) vers une strate où elle est moins adaptée que la strate dans laquelle elle se trouvait. Cela indique la présence d’une certaine redondance dans la recherche du minimum absolu.

Lisic et coll. (2018) supposent que l’introduction d’une pondération non uniforme dans la sélection des strates atomiques pourrait grandement améliorer les performances de leur méthode du recuit simulé (proposée) en échangeant les strates atomiques proches des bornes de strates plus fréquemment que les strates atomiques plus importantes. Nous sommes d’accord sur le fait que, pour cet algorithme, il serait plus avantageux qu’il y ait une plus grande probabilité qu’une strate atomique dissemblable aux autres strates atomiques soit sélectionnée. Nous pourrions alors rechercher une strate plus appropriée pour y déplacer cette strate atomique.

Pour y parvenir, nous pourrions d’abord sélectionner aléatoirement une strate, puis mesurer la distance euclidienne de chaque strate atomique par rapport au médoïde de la strate, en pondérant la probabilité de sélection des strates atomiques en fonction de leur distance par rapport au médoïde. À ce stade, une strate atomique est sélectionnée en utilisant ces probabilités pondérées.

L’étape suivante consisterait à utiliser un algorithme des K plus proches voisins pour trouver le médoïde de strate le plus proche de cette strate atomique et le déplacer vers cette strate. Cet algorithme simple d’apprentissage automatique s’appuie sur des mesures de distance pour classer les objets en fonction de leurs K plus proches voisins. Dans ce cas, $k = 1$, l’algorithme est donc en pratique un classificateur du plus proche voisin.

Ce degré de complexité supplémentaire de l’algorithme peut annuler les gains obtenus par l’utilisation de l’évaluation delta, en particulier lorsque la taille du problème augmente, ce qui réduit le nombre de solutions évaluées dans le même temps d’exécution. Il pourrait être plus efficace d’utiliser les médianes des colonnes comme un équivalent des médoïdes. Cela pourrait aider l’algorithme à trouver des solutions de meilleure qualité.

Cependant, les suggestions ci-dessus peuvent n'être efficaces qu'à un stade avancé de la recherche, lorsque les strates atomiques de chaque strate sont déjà assez similaires.

Remerciements

Nous tenons à remercier l'équipe de rédaction et les réviseurs de *Techniques d'enquête* pour leurs suggestions constructives lors du processus de révision de cette soumission à la revue, en particulier la suggestion de comparer l'algorithme du recuit simulé avec l'algorithme génétique traditionnel de Ballin et Barcaroli (2020) dans le cas de strates continues. Le présent document est basé sur des travaux financés par l'*Insight Centre for Data Analytics* et la *Science Foundation Ireland* dans le cadre de la subvention n° 12/RC/2289-P2, cofinancée par le Fonds européen de développement régional. Cette publication est également le fruit d'une recherche financée en partie par une subvention de la *Science Foundation Ireland* sous le numéro 16/RC/3918, cofinancée par le Fonds européen de développement régional.

Annexe

Précisions du contexte des comparaisons dans les sections 6 et 7

A.1 Contraintes de précision

Les niveaux de précision supérieurs cibles pour ces expériences, c'est-à-dire les coefficients de variation (CV), pour chacune des cinq expériences sont fournis dans le tableau A.1 ci-dessous.

Tableau A.1

Résumé par ensemble de données des limites supérieures des coefficients de variation

Ensemble de données	CV
Municipalités suisses	0,1
American Community Survey, 2015	0,05
Recensement des États-Unis, 2000	0,05
Prêts Kiva	0,05
Statistiques sur le commerce des produits de base de l'Organisation des Nations Unies	0,05

Nous avons choisi un niveau de précision supérieur de 0,1 pour l'ensemble de données des *municipalités suisses*, conformément au niveau fixé pour l'expérience de Ballin et Barcaroli (2020). Nous avons utilisé un niveau de précision supérieur de 0,05 pour les expériences restantes, étant donné que les niveaux de CV supérieurs généralement fixés par les instituts nationaux de statistique ont tendance à se situer entre 0,01 et 0,1, et que, pour cette raison, les résultats pour les CV situés au milieu de cette fourchette sont intéressants.

A.2 Plate-forme de traitement

Le tableau A.2 ci-dessous fournit des précisions sur la plate-forme de traitement utilisée pour ces expériences.

Tableau A.2
Spécifications de la plate-forme de traitement

Spécification	Précisions	Notes
Processeur	Processeur AMD Ryzen 9 3950X à 16 cœurs, 3 493 MHz	
Cœurs	16 cœurs	
Processeurs logiques	32 processeurs logiques	32 cœurs dans R
Modèle de système	X570 GAMING X	
Type de système	PC avec processeur x64	
Mémoire physique installée (RAM)	16 Go	
Mémoire virtuelle totale	35,7 Go	
Nom du système d’exploitation	Microsoft Windows 10 Pro	

Dans tous les cas, la version 4.0 ou ultérieure de R a été utilisée. Nous avons utilisé les progiciels *foreach* (Microsoft Corporation et Weston, 2020a) et *doParallel* (Microsoft Corporation et Weston, 2020b) pour exécuter les expériences en parallèle. Le nombre de cœurs utilisés dans les expériences était de 31 (32 moins 1), ce qui signifie que dans les trois expériences comportant plus de 31 domaines (*American Community Survey 2015*, *Prêts Kiva*, *Statistiques sur le commerce des produits de base de l’Organisation des Nations Unies*), l’algorithme *foreach* a continué à parcourir en boucle les cœurs disponibles jusqu’à ce qu’une solution ait été trouvée pour tous les domaines.

A.3 Hyperparamètres pour l’algorithme génétique de regroupement et l’algorithme du recuit simulé

Les tableaux A.3 et A.4 ci-dessous indiquent le nombre de domaines dans chaque expérience, ainsi que le nombre d’itérations et la taille de la population de chromosomes pour l’algorithme génétique de regroupement et le nombre de séquences, la longueur de la séquence et la température de départ pour l’algorithme du recuit simulé. La section A.4 fournit des précisions sur le peaufinage des hyperparamètres. Pour obtenir de plus amples renseignements sur les hyperparamètres de l’AGR, le lecteur peut consulter les travaux de Ballin et Barcaroli (2013) et de O’Luing et coll. (2019), et pour en savoir plus sur les hyperparamètres de l’ARS, les sections 2.2 et 4.

Tableau A.3
Résumé par ensemble de données des hyperparamètres de l’algorithme génétique de regroupement pour chaque domaine

Ensemble de données	Domaines	Nombre d’itérations, I	Taille de la population de chromosomes N_p	Chance de mutation	Taux d’élitisme, E_R	Ajout, facteur de strate
Municipalités suisses	7	4 000	50	0,0053360	0,4	0,0037620
American Community Survey, 2015	51	5 000	20	0,0008134	0,5	0,0610529
Recensement des États-Unis, 2000	9	100	20	0,0000007	0,4	0,0000472
Prêts Kiva	73	3 000	20	0,0007221	0,5	0,0685005
Statistiques sur le commerce des produits de base de l’Organisation des Nations Unies	171	1 000	20	0,0004493	0,3	0,0866266

Tableau A.4**Résumé par ensemble de données des hyperparamètres de l'algorithme du recuit simulé pour chaque domaine**

Ensemble de données	Domaines	Nombre de séquences, <i>maxit</i>	Longueur de la séquence, <i>J</i>	Température, <i>T</i>	Constante de décrement, DC	% de L pour valeur q maximale, $L_{\max\%}$		Probabilité de nouvelle strate, $P(H + 1)$
Municipalités suisses	7	10	3 000	0,0000720	0,5083686	0,0183356	0,0997907	
American Community Survey, 2015	51	3	3 000	0,0002347	0,6873029	0,0076477	0,0291729	
Recensement des États-Unis, 2000	9	2	2 000	0,0006706	0,5457192	0,0189395	0,0806919	
Prêts Kiva	73	5	2 000	0,0009935	0,7806557	0,0143925	0,0317491	
Statistiques sur le commerce des produits de base de l'Organisation des Nations Unies	171	3	3 000	0,0007902	0,5072737	0,0234728	0,0013775	

A.4 Peaufinage des hyperparamètres pour l'algorithme génétique de regroupement et l'algorithme du recuit simulé

Afin d'affiner les paramètres initiaux ou les *hyperparamètres*, nous avons utilisé l'optimisation séquentielle basée sur un modèle (Hutter, Hoos et Leyton-Brown, 2010). Nous avons d'abord généré un plan initial d'hyperparamètres à partir des fourchettes de valeurs décrites ci-dessous, dans le tableau A.5 pour l'AGR et dans le tableau A.6 pour l'ARS, en utilisant la méthode de plan en hypercube latin (McKay, Beckman et Conover, 2000).

Tableau A.5**Fourchettes pour le peaufinage des hyperparamètres de l'algorithme génétique de regroupement**

Type de valeur	Itérations			Taille de la population			Chances de mutation		Taux d'élitisme, E_r			Ajout, facteur de strate	
	Valeur discrète			Valeur discrète			Valeur numérique		Valeur discrète			Valeur numérique	
Fourchette de valeurs	Valeur inf.	Valeur sup.	Incréments	Valeur inf.	Valeur sup.	Incréments	Valeur inf.	Valeur sup.	Valeur inf.	Valeur sup.	Incréments	Valeur inf.	Valeur sup.
Municipalités suisses	500	5 000	500	10	50	10	0	0,10	0,1	0,5	0,1	0	0,1
American Community Survey, 2015	1 000	5 000	1 000	10	20	10	0	0,001	0,1	0,5	0,1	0	0,1
Prêts Kiva	1 000	3 000	1 000	10	20	10	0	0,001	0,1	0,5	0,1	0	0,1
Statistiques sur le commerce des produits de base de l'Organisation des Nations Unies	500	1 000	500	10	20	10	0	0,001	0,1	0,5	0,1	0	0,1
Recensement des États-Unis, 2000	50	100	50	10	20	10	0	0,000001	0,1	0,5	0,1	0	0,0001

Tableau A.6
Fourchette pour le peaufinage des hyperparamètres de l’algorithme du recuit simulé

Type de valeur	Nombre de séquences, $maxit$			Longueur de la séquence, J			Température, T		Constante de décrémentation, CD		% de L pour valeur q maximale, $L_{max\%}$		Probabilité de nouvelle strate, $P(H + 1)$	
	Valeur discrète			Valeur discrète			Valeur numérique		Valeur numérique		Valeur numérique		Valeur numérique	
Fourchette de valeurs	Valeur inf.	Valeur sup.	Incréments	Valeur inf.	Valeur sup.	Incréments	Valeur inf.	Valeur sup.	Valeur inf.	Valeur sup.	Valeur inf.	Valeur sup.	Valeur inf.	Valeur sup.
Municipalités suisses	10	50	10	1 000	3 000	1 000	0	0,001	0,5	1	0,0001	0,025	0	0,1
American Community Survey, 2015	1	3	1	1 000	3 000	1 000	0	0,001	0,5	1	0,0001	0,025	0	0,1
Prêts Kiva	1	5	1	1 000	2 000	1 000	0	0,001	0,5	1	0,0001	0,025	0	0,1
Statistiques sur le commerce des produits de base de l’Organisation des Nations Unies	1	3	1	1 000	3 000	1 000	0	0,001	0,5	1	0,0001	0,025	0	0,1
Recensement des États Unis, 2000	1	2	1	1 000	2 000	1 000	0	0,001	0,5	1	0,0001	0,025	0	0,1

Comme certaines fourchettes de valeurs des hyperparamètres étaient discrètes, nous avons utilisé une forêt aléatoire avec des arbres de régression pour mettre au point un modèle d’apprenant de substitution. Ensuite, une limite de confiance s’appuyant sur une valeur lambda, λ , pour contrôler le compromis entre l’exploitation et l’exploration a été utilisée comme fonction d’acquisition. L’approche de recherche focalisée (Bischla et coll., 2017) a été utilisée pour optimiser la fonction d’acquisition qui, à son tour, a été utilisée pour proposer les hyperparamètres qui ont été évalués à l’aide de la fonction de substitution (qui est une solution de rechange moins coûteuse à l’utilisation des algorithmes AGR ou ARS). Les hyperparamètres les plus prometteurs ont ensuite été évalués par l’AGR ou l’ARS, et les hyperparamètres et les coûts des solutions ont été ajoutés à la conception initiale. Le processus a ensuite été répété pour un nombre déterminé d’itérations, et les hyperparamètres et les résultats de solution les plus performants ont été sélectionnés. Nous avons mis cela en œuvre en utilisant la fonction *MBO* avec les paramètres indiqués dans le tableau A.7. Ces paramètres sont distincts de ceux qui sont peaufinés, lesquels sont décrits dans les tableaux A.5 et A.6 ci-dessus.

Tableau A.7
Paramètres utilisés dans la fonction MBO

Paramètres MBO	Valeur
Taille initiale de la conception (méthode de conception Hypercube latin)	10
Nombre d’itérations	10
Nombre d’arbres	500
Lambda, λ	5
Points de recherche ciblés	1 000

Comme le montre la portée limitée des paramètres de la fonction *MBO*, il ne s'agissait pas d'un peaufinage exhaustif des hyperparamètres pour l'AGR et l'ARS. Le but de ces expériences était d'examiner si l'ARS pouvait atteindre une qualité de solution comparable à celle de l'AGR en moins de temps de calcul par solution, se traduisant ainsi par des économies de temps d'exécution. Cependant, nous avons également comparé les temps d'exécution totaux, car cela est une conséquence de la nécessité d'entraîner les hyperparamètres pour les deux algorithmes.

Les tableaux décrivant les hyperparamètres, dans chacune des 20 itérations de peaufinage, pour chaque expérience peuvent être obtenus sur demande auprès des auteurs. Les 10 premiers ensembles d'hyperparamètres ont été générés de manière aléatoire à partir des fourchettes présentées dans les tableaux A.5 et A.6. Les fourchettes sélectionnées ont été définies grâce à une connaissance pratique des algorithmes et des données. Les 10 autres ensembles reflètent les tentatives de la fonction *MBO* d'apprendre les hyperparamètres qui conduisent le mieux chaque algorithme vers la solution optimale en utilisant les solutions précédentes comme guide.

A.5 Hyperparamètres pour l'algorithme génétique traditionnel et l'algorithme du recuit simulé

Les tableaux A.8 et A.9 présentent les hyperparamètres de l'algorithme génétique traditionnel et de l'algorithme du recuit simulé. L'option d'ajout de facteur de strate n'est pas disponible pour l'algorithme génétique traditionnel et, par conséquent, n'est pas incluse dans le tableau A.8. De plus amples précisions sur le peaufinage des hyperparamètres sont fournies dans la section A.6.

Tableau A.8
Hyperparamètres pour l'algorithme génétique traditionnel

Ensemble de données	Itérations	Taille de la population	Chances de mutation	Taux d'élitisme, E_R
Municipalités suisses	4 000	50	0,0053360	0,4
American Community Survey, 2015	1 000	20	0,0009952	0,1
Recensement des États-Unis, 2000	400	20	0,0002317	0,4
Prêts Kiva	200	20	0,0817285	0,5
Statistiques sur le commerce des produits de base de l'Organisation des Nations Unies	5 000	30	0,0005599	0,2

Tableau A.9
Hyperparamètres pour l'algorithme du recuit simulé

Ensemble de données	Nombre de séquences, $maxit$	Longueur de la séquence, J	Température, T	Constante de décrement, CD	% pour valeur q maximale, $L_{max\%}$	Probabilité de nouvelle strate, $P(H + 1)$
Municipalités suisses	5	5 000	0,02311057	0,9427609	0,3736443	0,0229361
American Community Survey, 2015	50	2 000	0,00000005	0,9528952	0,0001021	0,0000008
Recensement des États-Unis, 2000	1	2 000	0,00002000	0,9665631	0,0221147	0,0160408
Prêts Kiva	2	2 000	0,00053839	0,8660943	0,0014281	0,0216320
Statistiques sur le commerce des produits de base de l'Organisation des Nations Unies	2	250	0,00067481	0,9309940	0,0203113	0,0149499

A.6 Peaufinage des hyperparamètres pour l’algorithme génétique traditionnel et l’algorithme du recuit simulé

Nous avons affiné les hyperparamètres pour l’AGT et l’ARS en utilisant la même méthodologie que celle décrite dans la section A.4. Les tableaux présentant les hyperparamètres, dans chacune des 20 itérations de peaufinage, pour chaque expérience peuvent être obtenus sur demande auprès des auteurs. Les 10 premiers ensembles ont été générés de manière aléatoire en utilisant la connaissance pratique des algorithmes et des données pour définir des limites supérieures et inférieures pour chaque hyperparamètre. Dans les 10 autres ensembles, la fonction MBO tente d’optimiser les hyperparamètres en utilisant les solutions précédentes comme guide.

Bibliographie

- Ballin, M., et Barcaroli, G. (2013). [Détermination conjointe de la stratification et de la répartition optimales de l’échantillon en utilisant un algorithme génétique](#). *Techniques d’enquête*, 39, 2, 405-432. Article accessible à l’adresse <https://www150.statcan.gc.ca/n1/fr/pub/12-001-x/2013002/article/11884-fra.pdf>.
- Ballin, M., et Barcaroli, G. (2020). [Optimization of sampling strata with the SamplingStrata package](#). Consulté le 3 mai 2021. <https://barcaroli.github.io/SamplingStrata/articles/SamplingStrata.html>.
- Barcaroli, G. (2014). SamplingStrata: An R package for the optimization of stratified sampling. *Journal of Statistical Software*, 61, 4, 1-24.
- Benedetti, R., Espa, G. et Lafratta, G. (2008). [Une approche arborescente de la formation de strates dans les enquêtes-entreprises polyvalentes](#). *Techniques d’enquête*, 34, 2, 217-226. Article accessible à l’adresse <https://www150.statcan.gc.ca/n1/fr/pub/12-001-x/2008002/article/10760-fra.pdf>.
- Bethel, J.W. (1985). An optimum allocation algorithm for multivariate surveys. *Proceedings of the Section on Survey Research Methods*, American Statistical Association, 209-212.
- Bethel, J. (1989). [Répartition de l’échantillon dans les enquêtes à plusieurs variables](#). *Techniques d’enquête*, 15, 1, 49-60. Article accessible à l’adresse <https://www150.statcan.gc.ca/n1/fr/pub/12-001-x/1989001/article/14578-fra.pdf>.
- Bischla, B., Richter, J., Bossek, J., Horn, D., Thomas, J. et Lang, M. (2017). mlrmo: A modular framework for model-based optimization of expensive black-box functions. *Gradient Boosting in Automatic Machine Learning: Feature Selection and Hyperparameter Optimization*, 36.

- Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45,1, 41-51.
- Chromy, J.R. (1987). Design optimization with multiple objectives. *Proceedings of the Survey Research Methods Section*.
- Cochran, W.G.(1977). *Sampling Techniques*. 3rd Edition. New York: John Wiley & Sons, Inc.
- Cortez, P. (2014). *Modern Optimization with R*. Springer.
- Falkenauer, E. (1998). *Genetic Algorithms and Grouping Problems*. New York: John Wiley & Sons, Inc.
- Grover, L.K. (1987). Standard cell placement using simulated sintering. Dans *Proceedings of the 24th ACM/IEEE Design Automation Conference*, 56-59.
- Hartigan, J.A., et Wong, M.A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 28, 1, 100-108.
- Hoos, H.H., et Stützle, T. (2004). *Stochastic Local Search: Foundations and Applications*. Elsevier.
- Hutter, F., Hoos, H.H. et Leyton-Brown, K. (2010). Sequential model-based optimization for general algorithm configuration (version étendue). Rapport technique TR-2010–10, University of British Columbia, Computer Science.
- Johnson, D.S., Aragon, C.R., McGeoch, L.A. et Schevon, C. (1989). Optimization by simulated annealing: An experimental evaluation; part I, graph partitioning. *Operations Research*, 37, 6, 865-892.
- Kirkpatrick, S., Gelatt, C.D. et Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220, 4598, 671-680.
- Kiva (2018, Mar). [Data Science for Good: Kiva Crowdfunding](https://www.kaggle.com/kiva/data-science-for-good-kiva-crowdfunding). <https://www.kaggle.com/kiva/data-science-for-good-kiva-crowdfunding>.
- Lisic, J., Sang, H., Zhu, Z. et Zimmer, S. (2018). Optimal stratification and allocation for the june agricultural survey. *Journal of Official Statistics*, 34, 1, 121-148.
- Luke, S. (2013). [Essentials of Metaheuristics \(2 Ed.\)](http://cs.gmu.edu/~sean/book/metaheuristics/). Lulu. Disponible gratuitement sur <http://cs.gmu.edu/~sean/book/metaheuristics/>.

McKay, M.D., Beckman, R.J. et Conover, W.J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42, 1, 55-61.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. et Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21, 6, 1087-1092.

Microsoft Corporation and Weston, S. (2020a). *foreach: Provides Foreach Looping Construct*. R package version 1.5.1.

Microsoft Corporation and Weston, S. (2020b). *doParallel: Foreach Parallel Adaptor for the ‘Parallel’ Package*. R package version 1.0.16.

O’Luing, M., Prestwich, S. et Tarim, S.A. (2019). [Algorithme génétique de regroupement pour la stratification et la répartition simultanée de l’échantillon dans les plans de sondage](https://www150.statcan.gc.ca/n1/fr/pub/12-001-x/2019003/article/00007-fra.pdf). *Techniques d’enquête*, 45, 3, 549-568. Article accessible à l’adresse <https://www150.statcan.gc.ca/n1/fr/pub/12-001-x/2019003/article/00007-fra.pdf>.

Rose, J.S., Snelgrove, W.M. et Vranesic, Z.G. (1988). Parallel standard cell placement algorithms with quality equivalent to simulated annealing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 7, 3, 387-396.

Ross, P., Corne, D. et Fang, H.-L. (1994). Improving evolutionary timetabling with delta evaluation and directed mutation. Dans *International Conference on Parallel Problem Solving from Nature*, 556-565. Springer.

Ruggles, S., Genadek, K., Goeken, R., Grover, J. et Sobek, M. (2017). Integrated public use microdata series: Version 7.0 [dataset]. Minneapolis: University of Minnesota.

Sörensen, K., et Glover, F. (2013). Metaheuristics. *Encyclopedia of Operations Research and Management Science*, 62, 960-970.

Takeang, C., et Aurasopon, A. (2019). Multiple of hybrid lambda iteration and simulated annealing algorithm to solve economic dispatch problem with ramp rate limit and prohibited operating zones. *Journal of Electrical Engineering & Technology*, 14, 1, 111-120.

United Nations (2017, Nov). [Global Commodity Trade Statistics](https://www.kaggle.com/unitednations/global-commodity-trade-statistics). <https://www.kaggle.com/unitednations/global-commodity-trade-statistics>.

U.S. Census Bureau (2016). *2015 ACS Public Use Microdata Sample (PUMS)*. Washington, D.C.
<https://factfinder.census.gov/faces/nav/jsf/pages/searchresults.xhtml?refresh=t#>.

Van Laarhoven, P.J., et Aarts, E.H. (1987). Simulated annealing. Dans *Simulated Annealing: Theory and Applications*, Springer.