Proceedings of the 65th Annual Conference of the South African Statistical Association for 2024

18 – 22 November 2024
Stellenbosch

Proceedings of the 65th Annual Conference of the South African Statistical Association for 2024 (SASA 2024)

ISBN 978-1-0370-7180-5

Editor

Charl Pretorius North-West Universtiy

Assistant Editors

Niladri Chakraborty University of the Free State
Ansie Smit University of Pretoria

Managing Editor

Neill Smit North-West University

Review Process

Thirteen (13) manuscripts were submitted for possible inclusion in the Proceedings of the 65th Annual Conference of the South African Statistical Association. All submitted papers were assessed by the editorial team for suitability, after which all papers were sent to be reviewed independently. Papers were reviewed according to the following criteria: relevance to conference themes, relevance to audience, standard of writing, originality and critical analysis. After consideration and incorporation of reviewer comments, eight manuscripts were judged to be suitable for inclusion in the proceedings of the conference.

Reviewers

The editorial team would like to thank the following reviewers:

Jimoh Ajadi	King Fahd University of Petroleum & Minerals
Delson Chikobvu	University of the Free State
Roelof Coetzer	North-West University
Aysegul Erem	Cyprus International University
Marien Graham	University of South Africa
Schalk Human	Deloitte
Akanksha Kashikar	Savitribai Phule Pune University
Shawn Liebenberg	North-West University
Chun Fai Lui	City University of Hong Kong
Tahir Mahmood	University of the West of Scotland
Jean-Claude Malela-Majika	University of Pretoria
Johané Nienkemper-Swanepoel	Stellenbosch University
Edmore Ranganai	University of South Africa
Michael Rogans	University of the Witwatersrand
Leonard Santana	North-West University
Ridwan Sanusi	King Fahd University of Petroleum & Minerals
Neill Smit	North-West University
Sean van der Merwe	University of the Free State
Janet van Niekerk	King Abdullah University of Science and Technology
Andréhette Verster	University of the Free State

Contact Information

Queries can be sent by email to the Managing Editor (managing.editor@sastat.org).

Table of Contents

An R Shiny application for optimising mixed medley relay teams in swimming 1
S. Ackerman, P. J. van Staden and I. Fabris-Rotelli

Integrating PCA with random search for variable importance in multivariate stratified sample allocation 17
G. Borros, Ş. Er and S. Salau

Quantifying the directional relationship between elliptical natural hydrogen depressions and geological lineaments in Mpumalanga, South Africa 33
C. J. Botha, A. Smit, A. J. Bumby, I. Fabris-Rotelli, B. O. Mac'Oduol and N. Nakhaeirad

Covariate-based distance outlier scoring for nonconvex domain estimation 49
K. Mahloromela and I. Fabris-Rotelli

Investigation on the robustness of clustered point pattern simulation 65
A. E. Pieters, R. Stander, K. Mahloromela, R. Thiede and I. Fabris-Rotelli

An approximate Bayesian computation threshold-search algorithm for parameter estimation 79
N. Smit

An improved similarity test for comparing spatial point patterns 93
R. Stander, I. Fabris-Rotelli, G. Breetzke and J. Stander

Spatial network analysis for predicting future densification within South African informal settlements 107
R. van der Walt, C. van Zyl, R. N. Thiede and I. N. Fabris-Rotelli

Submitted to the Proceedings of the 65th Annual Conference of the South African Statistical Association

An R Shiny application for optimising mixed medley relay teams in swimming

San-Mari Ackerman, Paul J. van Staden and Inger Fabris-Rotelli

Department of Statistics, University of Pretoria, Pretoria, South Africa

The allocation of Masters swimmers to relay teams presents a complex optimisation problem, due to various combinations of age groups, genders and swimming stroke type. This study explores the optimisation of team selection dynamics in Masters swimming, focusing on mixed medley relay teams. In this article, we aim to enhance performance through mathematical modeling and extend previous research by incorporating stroke-specific constraints using integer linear programming (ILP). We replicate and advance existing findings, using the R programming language, addressing complexities such as gender balance, stroke specialisation, and age categories. This research contributes to sports science by providing insights into optimal team compositions, applicable not only to Masters swimming but also serving as a template for team selection in other sports domains. Additionally, it contributes to statistics by demonstrating the practical application of optimisation techniques and combinatorial methods in solving real-world team selection problems.

Keywords: Integer Linear Programming, Masters Swimming, Mixed Medley Relay, Shiny App, Team Composition.

1. Introduction

This study explores the intersection of sports science and mathematical optimisation, focusing on team selection for Masters swimming mixed medley relay events. Masters swimming encompasses amateur competitive swimming for individuals aged 18 and older. In South Africa, the criteria for age and regulations vary by club or governing body[1]. These events are divided into age categories based on the cumulative age of four swimmers, adding a unique layer of complexity to team composition.

Unlike conventional ranking-based approaches to team formation, the inclusion of age categories, stroke specialisation, and gender balance introduces an optimisation problem that requires careful consideration. This research addresses these complexities using integer linear programming (ILP) to optimise team assignments. By minimising total race times while respecting age and stroke constraints, this study contributes a novel mathematical approach to enhancing performance in Masters swimming.

Team selection in sport plays a pivotal role in both individual development and overall team success. By applying advanced mathematical techniques, this research aims to not only improve team performance but also offers a framework for broader application in sports team management.

Corresponding author: Paul J. van Staden (paul.vanstaden@up.ac.za)

MSC2020 subject classifications: 62–06

[1] https://www.samastersswimming.com/

1.1 Literature Review

Integer programming (IP) models have been widely applied in team selection across various sports. These models optimise team composition by defining decision variables, objective functions, and constraints, such as non-negativity and technology conditions (Sierksma and Zwols, 2015).

Gerber and Sharp (2006) pioneered the use of IP for selecting cricket squads, applying it to a 15-player squad for one-day international tournaments. While innovative, the model faced challenges like multiple optimal solutions and the need for detailed cricket-specific knowledge to formulate coefficients. Building on this, Sharp et al. (2011) refined the IP approach by introducing precise ability indices, reducing subjective bias and extending the framework to Twenty20 (T20) cricket.

Further advancements were made by Chand et al. (2018), who developed a multi-objective IP model incorporating financial metrics, player performance, and star power for cricket team auctions. This model provided trade-off solutions, balancing performance and cost under budget constraints. Similarly, Gokul and Sundararaman (2023) optimised player selection in the Indian Premier League by using past performance data to maximise the likelihood of success against specific opponents.

Expanding beyond cricket, Fabris-Rotelli et al. (2022) applied an integer optimisation program to form optimal freestyle relay teams for Masters swimming competitions. The model, using swimmers' fastest 50m times, successfully minimised cumulative lap times across male, female, and mixed relay teams, demonstrating the effectiveness of IP in team composition for swimming. This is the only literature available that addresses integer optimisation in Masters swimming due to the complexities involved in the relay team structures. This paper thus contributes to the literature by expanding on the approach in Fabris-Rotelli et al. (2022). One should note that the use of genetic algorithms or dynamic programming could also be considered as alternatives, but would, however, make the problem more complex than necessary.

2. Methodology

This study aims to develop an IP model to assemble teams of four relay swimmers for a mixed medley relay swimming competition to minimise the overall cumulative time of the teams. Each member can participate in any of the following strokes within a single team: freestyle, backstroke, breaststroke, and butterfly. The question begs which combination of swimmers would achieve the best results.

To illustrate the complexity of this question, consider the following example: The Swedish relay is an athletics relay event that involves teams of four runners. The first team member runs 100m, followed by the second covering 200m. The third then runs 300m, and the final member finishes with a 400m lap, completing the 1km relay. This event is particularly interesting, since most athletes that focus on the 200m event, are skilled at either the 100m or 400m event as well. Additionally, the 300m event is an uncommon track event that few athletes specifically train for. In the context of South African athletics, would it be most effective for Wayde van Niekerk, a skilled athlete in both the 200m and 400m events, to run the 200m, 300m, or 400m leg of the race? This obviously depends on the other members of the team as well and evokes an optimisation model.

A similar conundrum is faced in swimming as some athletes compete well in multiple stroke-specific events. Also, the butterfly stroke is particularly difficult to master and consequently less popular among Masters swimmers. Additionally, for this relay event, each team of a club needs to adhere to a certain age constraint, specifically that the cumulative ages of the four swimmers in each

Table 1. Categories of year ranges.

Category	Years Range
1	100 to 119 years
2	120 to 159 years
3	160 to 199 years
4	200 to 239 years
5	240 to 279 years
6	280 to 319 years
7	320 to 359 years
8	360 to 399 years

team have to fall within a different age category. The categories, as specified by World Aquatics, are listed in Table 1. For example, a relay team with four swimmers aged 19, 48, 58 and 76 respectively, has cumulative age 19+48+58+76=201 and therefore falls, as indicated in Table 1, in Category 4: 200 to 239 years. Hence, there is a need for a mathematical algorithm to account for all the possibilities.

Any integer optimisation program (IOP) can be constructed by defining an objective function, decision variables, and various constraints (Wolsey and Nemhauser, 1988). The objective function is a value, expressed as a function of the decision variables, to be optimised, e.g., maximising an investment's profit or minimising energy consumption by a particular powerplant. To obtain the desired result, the specific values of the decision variables are modified. The range of the variables is limited by sets of equations, known as the constraints, for instance, the budget allocated for a specific investment.

To construct the IOP, define the **decision variable**,

$$x_{ijkl} = \begin{cases} 1, & \text{if swimmer } i \text{ is selected to swim in age category } j \text{ using stroke } k \text{ and is of gender } l \\ 0, & \text{otherwise,} \end{cases}$$

where $i \in \{1, ..., n_i\}, j \in \{1, ..., n_j\}, k \in \{1, ..., 4\}, l \in \{1, 2\}$ with n_i the number of swimmers and n_j the number of age categories.

To achieve the desired result, the **objective function** T is minimised:

$$T = \sum_{j=1}^{n_j} \left(\sum_{i=1}^{n_i} \sum_{k=1}^{4} \sum_{l=1}^{2} t_{ijkl} x_{ijkl} - r_j \right), \tag{1}$$

where t_{ijkl} is the recent fastest time for swimmer i selected for age group j to participate in stroke k, of gender l and r_j is the current record for age category j.

To guarantee that the team selection is practically possible, the following **constraints** are employed:

$$\sum_{j=1}^{n_j}\sum_{k=1}^{4} x_{ijkl} \leq 1 \forall i, l, \tag{2}$$

$$\sum_{i=1}^{n_i}\sum_{k=1}^{4} x_{ijkl} \leq 1 \forall j, \ell, \tag{3}$$

$$\sum_{i=1}^{n_i}\sum_{k=1}^{4}\sum_{l=1}^{2} x_{ijkl} = 4, \tag{4}$$

$$\sum_{i=1}^{n_i}\sum_{l=1}^{2} x_{ijkl} = 2 \forall j, k, \tag{5}$$

$$LL_j \leq \sum_{i=1}^{n_i}\sum_{k=1}^{4}\sum_{l=1}^{2} a_i x_{ijkl} \leq UL_j \forall_j, \tag{6}$$

where a_i is the age of swimmer i and LL_j and UL_j are the lower and upper limits of age group j.

Constraint (2) guarantees that each swimmer i is selected at most once across all events and age groups. Constraint (3) ensures that each stroke k within each age category j is utilised exactly once. Constraint (4) mandates that each team comprises exactly four swimmers. Constraint (5) ensures that each team consists of exactly two male and two female swimmers. Lastly, Constraint (6) restricts each team's total age to fall within predefined age group boundaries LL_j and UL_j.

The program is implemented using the `lpSolve` package (Berkelaar and Csárdi, 2023) in the `R` programming language, in conjunction with `RShiny` developed by Chang et al. (2023) to create a user-friendly application where data can be uploaded and results displayed.

2.1 Technical framework

`lpSolve` is a package for solving linear, integer, and mixed integer programming problems. Within `lpSolve`, the function `lp()` is used to solve linear programming problems and allows us to define the objective function, constraints, and variable bounds to find the optimal team assignment. The function `lp()` employs matrices $\mathbf{A_1}, \mathbf{A_2}, \mathbf{A_3}$ (the left-hand side of the constraints) along with right-hand vectors $\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3}, \mathbf{b_4}$ to encode constraints and formulates the objective function $\mathbf{C}$ based on swimmer times and performance records.

The constraint matrices are

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & & & \\ 0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{n \times (n \cdot s \cdot m)},$$

$$A_2 = \begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{(s\cdot m)\times(n\cdot s\cdot m)}$$

and

$$A_3 = \begin{bmatrix} \text{Age}_1 & \text{Age}_2 & \cdots & \text{Age}_n & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \text{Age}_1 & \text{Age}_2 & \cdots & \text{Age}_n \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \text{Age}_n \end{bmatrix}_{m\times(n\cdot s\cdot m)},$$

while the objective function is

$$C = T_1 - \frac{R}{4}.$$

Each constraint matrix has $n{\cdot}m{\cdot}s$ columns. We construct the complete matrix $\mathbf{A} = (\mathbf{A_1}, \mathbf{A_2}, \mathbf{A_3}, \mathbf{A_3})$ used in `lp`. Within the `lp` function, $\mathbf{A}$ is multiplied by an $n \cdot m \cdot s \times 1$ vector v of 1s, which is compared with a $(n + s \cdot m + m) \times 1$ right-hand vector $(\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3}, \mathbf{b_4})$, where each entry represents the respective constraints.

Focusing on $\mathbf{A_1}$, for any row i, the row contains $s \cdot m$ copies of a $(1 \times n)$ vector, having all zeros except for the i-th entry. The resulting vector of $\mathbf{A_v}$ is an $n \cdot m \cdot s \times 1$ vector. Each entry represents an outcome of a binary variable x_{ijk} being 1 or 0. For the first n entries of the vector, it is either 1 or 0 depending on whether swimmer i swims the first stroke in the first team. The subsequent n entries are either 1 or 0 depending on whether swimmer i swims the second stroke in the first team. This pattern continues for all 4 strokes and then repeats for all m teams.

$\mathbf{A_1}$ ensures each swimmer swims only one stroke per event. Each row corresponds to a swimmer, and each column corresponds to a specific swimmer-stroke-group combination. $\mathbf{A_2}$ ensures each stroke is covered exactly once per event. Each row corresponds to a specific stroke in a group, and each column corresponds to a swimmer-stroke-group combination. $\mathbf{A_3}$ enforces the age constraints for each group. Each row corresponds to a group, and each column represents a swimmer-stroke-group combination with the swimmer's age as the coefficient. Within the `lp()` constraint argument, $\mathbf{A_3}$ is incorporated twice to enforce both lower and upper age limits.

The coefficient vector C represents the objective function's coefficients, combining the swimmers' times and the record times, aiming to minimise the total adjusted time, T_1 is the vector of swimmers' times, and R is the vector of record times. These matrices are used as arguments for the `lp()` function, which minimises the objective function while satisfying the constraints encoded in the matrices.

2.2 Shiny App

Shiny is more than just a package in R (R Core Team, 2025), as it enables a comprehensive framework for the straightforward development of interactive web applications. A Shiny application (app) is composed of two main components: the user interface (UI) and the server. The UI determines the app's design and layout, while the server manages the app's logic and functionality.

In this specific application, the Shiny app allows users to upload data and interactively view results. Once uploaded, the data is processed using the `lpSolve` package to assign swimmers optimally. The results are then displayed in a user-friendly data table, with clear formatting to distinguish between different teams and strokes. Upon launching the application, users are presented with a dedicated instruction page featuring two main tabs: "Instructions" and "Data Input."

In the "Instructions" tab, users will find comprehensive guidelines for uploading data. This section ensures that users, regardless of their experience level, understand the specific data structures required for compatibility with the program. The instructions are accompanied by visual representations that illustrate how the Excel files should be formatted, providing clear examples to facilitate the upload process. By following these instructions, both novice and experienced R users can effectively navigate the app and successfully prepare their data.

The "Data Input" tab allows users to upload their pre-configured Excel files. Here the underlying code is ran to generate results based on the uploaded data. This streamlined workflow enables users to easily transition from understanding the requirements to actively engaging with the app's functionality.

Together, these tabs create an intuitive user experience, guiding users through the process of data preparation and analysis in a clear and structured manner. As illustrated in Figure 1, users can browse and upload a pre-configured Excel file through the app's interface. Upon successful upload, the data is loaded into the server, and entries from the relevant sheets are retrieved. These entries are then processed through an external assignment function, using the methods described above to optimise team and stroke assignments.

The assignment function begins by defining the relevant index parameters, including the number of swimmers n and teams m, as well as the lower and upper age limits for each team. Using this information, the constraint matrices are constructed according to the methods described in the previous section. The decision function is subsequently optimised, using `lp()`, to produce the output matrix.

The resulting output is a binary matrix where each element is either 0 or 1, indicating whether a particular swimmer is assigned to a specific stroke within a given team. This matrix is then multiplied by the vector containing the names of all swimmers in the club. This multiplication step, performed within the server environment, produces a final matrix where each row corresponds to a specific team and each column represents one of the four strokes as shown in Figure 1.

As a final note, it is common for swimmers to skip certain strokes or lack recorded best times when forming teams for an event. To address this, it must be ensured that missing data is marked with an asterisk (*) in the Excel file and the time for that stroke is then set to a large value (e.g., 10,000) in the program. This will prevent the swimmer from being assigned to that stroke.

The application can be further extended and adapted to various team-based sports, transforming what traditionally takes hours of discussion and debate around team selection into a streamlined process that can be completed in just a few seconds.

3. Application

This research focuses on data from a single swimming club in South Africa. The dataset consists of each swimmer's age and their most recent best (fastest) recorded time for each stroke. As mentioned

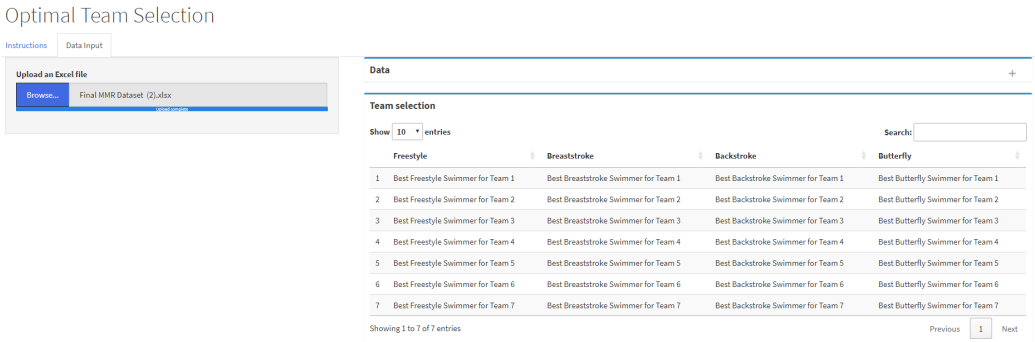


Figure 1. Team-stroke assignment table in the app.

before, the aim is to classify each member of the club into a team in such a way that maximum points will be scored in upcoming swimming relay events, or said otherwise, to minimise the total time per race over all the teams.

The dataset captures the best times of 74 swimmers, 42 females (F1-F42) and 32 males (M1-M32), from a South African swimming club, across four strokes: freestyle, backstroke, breaststroke, and butterfly. Ages range from 21 to 90 years, showing a diverse group of competitors. While some swimmers recorded times for all strokes, others have missing values, indicating non-participation or unavailable data. This dataset offers a clear view of the club's performance, highlighting individual strengths and stroke specialisations.

In mixed medley relay swimming competitions, teams score points based on their finishing position within their age group. The exact scoring system can vary by event organiser or competition, but generally, higher placements in the race earn more points. For example, a team that finishes first in its category might receive the maximum points (e.g. 10 or 20), with second-placed and third-placed teams earning progressively fewer points, such as 8 and 6 points, respectively.

3.1 Descriptive Analysis

Upon preliminary analysis of the data, based on descriptive statistics and boxplot diagrams shown in Figure 2, it became apparent that there is a noteworthy discrepancy in the variation in time among different strokes. Furthermore, Figure 2 indicates that for all four stroke times, and in particular for the butterfly stroke times, the medians and interquartile ranges (as measures of location and spread respectively) are larger for females compared to males. These observations prompt us to consider utilising stronger swimmers in strokes characterised by higher variation as the performance of weaker swimmers might drag down the overall team performance. Fortunately, the IOP accounts for this by delivering an optimal solution equivalent to considering all possible combinations, including scenarios where stronger swimmers are assigned to swim a specific stroke.

3.1.1 ANOVA on stroke and gender

Before using the Shiny app to select the relay teams, more detailed descriptive analyses were performed to study the effect of age, gender and stroke type on the swim times. Firstly, a two-way Analysis of Variance (ANOVA) was conducted to assess the differences in swim times based on stroke type, gender, and their interaction. In particular, it was used to test whether the swim times

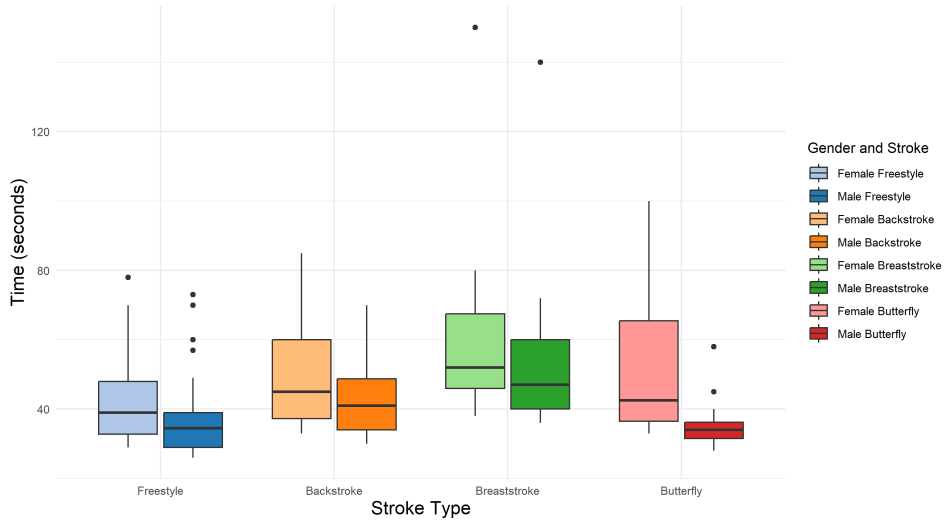


Figure 2. Boxplots of swim times by stroke and gender.

differ across the four stroke types (freestyle, backstroke, breaststroke, and butterfly) and between males and females. Furthermore, it evaluated whether the effect of stroke on swim time is different for males and females (i.e., the interaction between stroke and gender). Because not all the necessary assumptions for ANOVA – specifically the assumptions of normality and homogeneity of variance – were met, the Aligned Rank Transform (ART) procedure for performing nonparametric ANOVA was used. This robust procedure provides accurate nonparametric treatment for both main and interaction effects by first aligning all responses for each possible main effect or interaction before assigning ranks – see Wobbrock et al. (2011) for full details. The ART procedure for ANOVA was implemented in R using the `art()` function from the ARTool package by Kay et al. (2025), with swim time as the response variable and stroke type, gender, and their interaction as the explanatory variables. All observations with missing swim times were excluded. The results indicate that stroke type has a highly significant effect on swim times, with p-value < 0.001. This suggests that there are substantial differences in swim performance depending on the stroke type. The finding is consistent with expectations, as different strokes naturally involve varying physical demands and techniques, leading to differences in performance. Similarly, gender was also found to have a highly significant effect on swim times, with a p-value < 0.001. Given that physiological differences can contribute to variations in athletic performance, this result aligns with expectations that males and females perform differently in competitive swimming. The interaction between stroke type and gender, however, was not significant, with p-value = 0.7275.

3.1.2 Post-hoc pairwise comparisons

To further investigate the differences in mean swim times across various strokes, we conducted pairwise comparisons using the ART-C algorithm developed by Elkin et al. (2021). As shown by these authors through an extensive simulation-based validation study, the ART-C algorithm does not inflate Type I error rates. This algorithm can be applied in `R` with the `art.con()` function from

Table 2. Summary of significant results from post-hoc pairwise comparisons.

Comparison	Estimate	Std. Error	Adjusted p-value
Gender			
Female vs Male	28.9594	7.9082	< 0.001***
Stroke			
Freestyle vs Backstroke	-31.0002	9.2535	0.0050**
Freestyle vs Breaststroke	-56.6427	9.6206	<0.001***
Backstroke vs Breaststroke	-25.6425	10.7247	0.0537†
Breaststroke vs Butterfly	37.1930	11.7899	0.0076**
Gender*Stroke			
Female*Backstroke vs Male*Butterfly	57.7917	15.8571	0.0078**
Female*Backstroke vs Male*Freestyle	53.7344	12.2376	<0.001***
Male*Backstroke vs Female*Breaststroke	-44.9167	13.4454	0.0206*
Female*Breaststroke vs Male*Butterfly	79.8472	15.3301	<0.001***
Female*Breaststroke vs Female*Freestyle	50.7222	11.2492	<0.001***
Female*Breaststroke vs Male*Freestyle	75.7899	11.5466	<0.001***
Male*Breaststroke vs Male*Butterfly	65.6378	17.6886	0.0065**
Male*Breaststroke vs Male*Freestyle	61.5805	14.5327	<0.001***
Female*Butterfly vs Male*Butterfly	51.7292	16.8739	0.0481*
Female*Butterfly vs Male*Freestyle	47.6719	13.5292	0.0115*

† Marginally significant (p-value < 0.1)
* Moderately significant (p-value < 0.05)
** Significant (p-value < 0.01)
*** Highly significant (p-value < 0.001)

the `ARTool` package. There has recently been some debate in the literature regarding the validity of the usage of p-value adjustment corrections for multiple comparisons, specifically with conservative methods such as the Bonferroni correction – see for instance Barnett et al. (2022). Following their advice, since the multiple pairwise comparisons in our study are closely linked, we used the Holm-Bonferroni correction proposed by Holm (1979), which is a less conservative alternative to the Bonferroni correction and is also uniformly more powerful compared to the Bonferroni correction.

In Table 2, we present only the significant results from the post-hoc pairwise comparisons to highlight the most relevant differences in swim times. Note that the estimates given in Table 2 are on the scale of the ranks and not the observations, so these estimates are not the mean differences in swim times between the compared pairs. However, the sign of an estimate for a specific comparison does indicate which stroke has the faster swim times – a negative (positive) estimate implies faster swim times for the first (second) stroke in the comparison. The swim times are the fastest for freestyle compared to all three other stroke types. Notably, there is a highly significant difference between the swim times for freestyle and breaststroke (adjusted p-value < 0.001) and also a significant difference between the swim times for freestyle and backstroke (adjusted p-value = 0.0050). But no significant difference was found between the times for freestyle and butterfly (adjusted p-value = 0.1299).

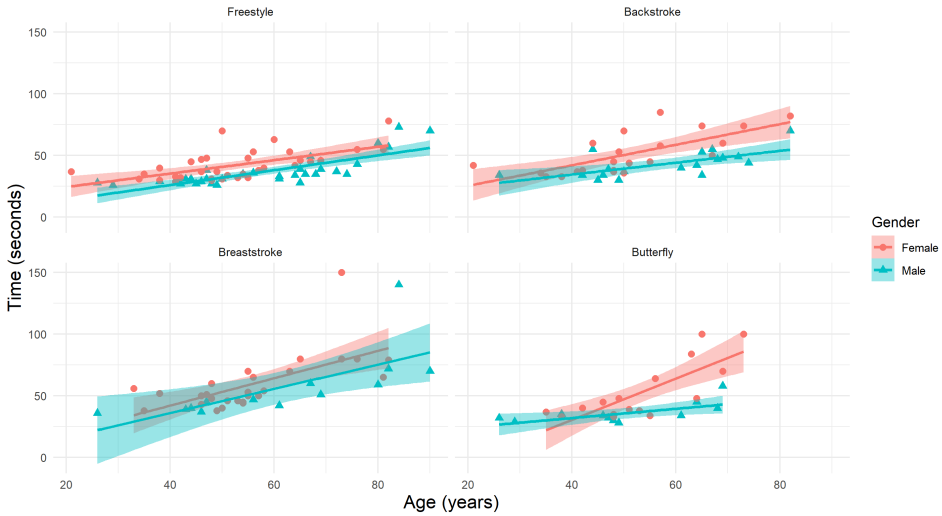


Figure 3. Scatterplots of the swim times over age by stroke for different genders.

The swim times between breaststroke and butterfly are also significantly different (adjusted p-value = 0.0076), while the swim times between backstroke and breaststroke are marginally significant (adjusted p-value = 0.0537). There is no significant difference between the swim times of backstroke and butterfly (adjusted p-value = 0.3163). Considering the pairwise comparisons for the various interactions of stroke type and gender, multiple significant differences are observed from Table 2. Of these significant results, the following differences are of particular interest:

- For both females and males, there are highly significant differences between the swim times of breaststroke and freestyle (adjusted p-values < 0.001 for both comparisons).
- There is a significant difference between the swim times of breaststroke and butterfly for males (adjusted p-value = 0.0065).
- Butterfly is the only specific stroke for which there is a (moderately) significant difference between the swim times of females and males (adjusted p-value = 0.0481).

3.1.3 Scatterplots and linear regression models

As further analysis and to better understand how the combination of the swimmers' ages in a team will influence team selection, it is helpful to examine the relationship between the ages of the swimmers and their respective times for each stroke. Therefore, scatterplots showing the relation between the swim times and the ages for the four different strokes are presented in Figure 3. Regression lines with 95% confidence intervals are added to the scatterplots to further elucidate these relationships for females and for males. The fitted models' regression coefficients and the corresponding R^2 values are summarised in Table 3. Overall, based on their R^2 values, the fitted regression lines for males' freestyle and for females' butterfly provide the best fits. The availability of the most data for freestyle compared to the other strokes, due to its commonality and also its popularity amongst swimmers,

Table 3. Regression model coefficients and R^2 values for different strokes and genders.

Stroke	Gender	Model (Intercept, Slope)	R^2
Freestyle	Female	(13.6568, 0.5438)	0.3670
	Male	(2.0253, 0.6021)	0.6222
Backstroke	Female	(8.7868, 0.8334)	0.5030
	Male	(15.4009, 0.4805)	0.4240
Breaststroke	Female	(-2.3489, 1.1102)	0.3986
	Male	(-3.3076, 0.9824)	0.4751
Butterfly	Female	(-36.5843, 1.6764)	0.6212
	Male	(16.9600, 0.3754)	0.4202

likely contributes to the robustness of the model for male swimmers. In contrast, the fitted regression lines for females' freestyle and breaststroke show the weakest fits, suggesting that factors beyond age play crucial roles in determining swimming times for these swimmers. From Table 3 it is noted that, whereas the slope coefficients of the regression lines for the females and males are in general equivalent for the different strokes, there is a clear numerical difference between the slope coefficients for butterfly. This difference is also visible in Figure 3. Recall from Section 3.1.2 that it was found that there is a (moderately) significant difference between the swim times of females and males for butterfly. These observations suggest that the relationship between age and swimming performance might not be the same for both genders within each stroke type. To fully investigate the effects of age and gender on swim times rigorously, a multiple linear regression model is utilised in Section 3.2, incorporating interaction terms between age and gender.

3.2 Application of Linear Mixed Model

Linear mixed models extend linear regression by incorporating both fixed and random effects. These models are particularly useful when dealing with data that has a nested or grouped structure, where observations within the same group may be correlated.

The general form of a linear mixed model can be written as

$$y_{ij} = \mathbf{X}_{ij}\boldsymbol{\beta} + \mathbf{Z}_{ij}\boldsymbol{u}_i + \epsilon_{ij}, \tag{7}$$

where

- y_{ij} is the outcome for the i-th individual in group j,
- $\mathbf{X}_{ij}$ is the vector of predictors for fixed effects,
- $\boldsymbol{\beta}$ is the vector of fixed-effect coefficients,
- $\mathbf{Z}_{ij}$ represents the random-effect predictors,
- $\boldsymbol{u}_i$ is the random effect for the i-th group, typically assumed to be normally distributed: $\boldsymbol{u}_i \sim N(0, \boldsymbol{\Sigma})$,
- ϵ_{ij} is the residual error term, with $\epsilon_{ij} \sim N(0, \sigma^2)$.

Table 4. Regression coefficients for each stroke.

Stroke	Coefficient	Estimate	Std. Error	*p*-value
Freestyle	(Intercept)	2.0253	5.8953	0.7323
	Age	0.6021	0.0971	< 0.001***
	Gender	11.6315	8.4576	0.1738
	Age*Gender	-0.0584	0.1482	0.6951
Backstroke	(Intercept)	15.4009	10.4872	0.1506
	Age	0.4805	0.1771	0.0101*
	Gender	-6.6142	13.5610	0.6287
	Age*Gender	0.3529	0.2403	0.1507
Breaststroke	(Intercept)	-3.3076	17.8312	0.8539
	Age	0.9824	0.2788	0.0012**
	Gender	0.9587	24.2702	0.9687
	Age*Gender	0.1278	0.4015	0.7521
Butterfly	(Intercept)	16.9600	13.2842	0.2139
	Age	0.3754	0.2537	0.1519
	Gender	-53.5443	20.4710	0.0152*
	Age*Gender	1.3011	0.3821	0.0023**

* Moderately significant (p-value < 0.05)
** Significant (p-value < 0.01)
*** Highly significant (p-value < 0.001)

In this model, the fixed effects $\boldsymbol{\beta}$ represent overall population-level effects, while the random effects $\boldsymbol{u}_i$ account for the variability within groups or clusters. The random effects allow different groups (or individuals) to have their own intercepts and/or slopes, which captures the correlation structure in the data.

The linear mixed model for the swim times is expressed as follows:

$$y_{ij} = \beta_1 + \beta_2 x_i + \beta_3 d_i + \beta_4 (x_i \cdot d_i) + \epsilon_{ij} \tag{8}$$

where

- y_{ij} represents the swim time for individual i at age j.
- x_i denotes the age of individual i.
- $d_i = \begin{cases} 1 & \text{if individual } i \text{ is female,} \\ 0 & \text{if individual } i \text{ is male.} \end{cases}$
- ϵ_{ij} is the random error term for individual i at age j.

The inclusion of the interaction term $x_i \cdot d_i$ allows us to explore how the relationship between age and swim time varies by gender. The results of the linear mixed model for each stroke (freestyle, backstroke, breaststroke, and butterfly) are summarised in Table 4.

Age significantly impacts swimming performance across all strokes, with older swimmers generally recording slower times. This effect is strongest in freestyle ($p < 0.001$), backstroke (p-value< 0.05), and breaststroke (p-value< 0.01), though gender differences are not significant for these strokes.

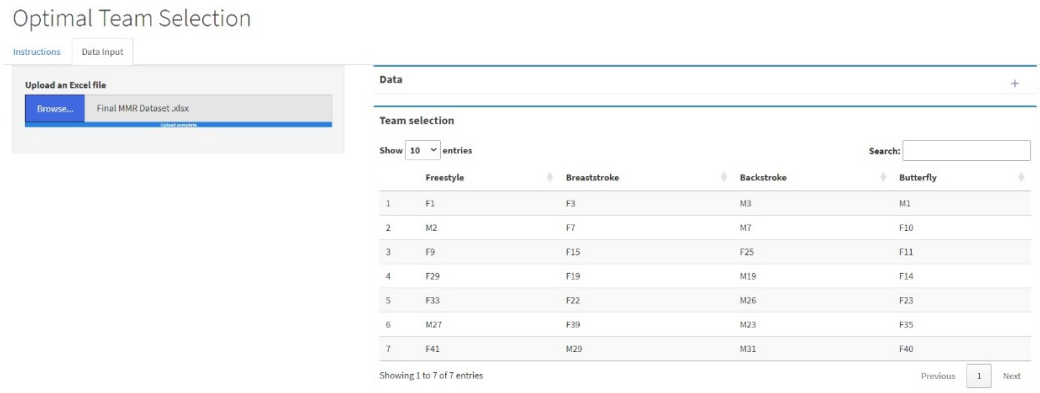


Figure 4. Final team-stroke assignment table from the app.

With the butterfly stroke, however, gender plays a crucial role (p-value<0.05), with males performing better, and the interaction between age and gender is also significant (p-value<0.01), indicating different aging effects between genders.

3.3 Optimisation of Swimming Relay Teams

The data was imported into the Shiny app from an Excel file using the toggle feature illustrated in Figure 1. The output generated by the Shiny app is shown in Figure 4. This process provides a straightforward and efficient method for making mathematically sound decisions regarding team selection.

It is noteworthy to examine the gender ratio in this selection process. Since the requirement of having exactly two men and two women on each team was not strictly enforced, it allowed for the possibility of an unequal representation of each gender. For instance, Team 3, which falls within the age category of 160 to 199, consists solely of females, three middle-aged women in their 40s and one woman aged 55. Similarly, the fourth team, categorised as 200 to 239, is almost exclusively female members.

The predominance of female swimmers in Teams 3 and 4 is attributed to the performance-focused selection criteria rather than any deliberate gender bias as team composition was based on swim times without enforcing gender balance. The female swimmers in these age groups achieved faster times than their male counterparts, leading to their inclusion. An alternative possible explanation is that women in these age categories may maintain competitive swim times for longer in certain strokes or distances. Additionally, there may have been a larger pool of female participants in these age ranges, increasing their chances of being selected based solely on their performance.

4. Conclusion

This research successfully applies data analysis techniques to optimise team composition for a South African swimming club. Using a dataset of swimmers' ages and best times, we developed an algorithm that minimises total race times for mixed medley relay events. Descriptive statistics, ANOVA, and post-hoc analyses revealed significant performance variations across strokes and genders, highlighting

the impact of stroke selection on overall team performance. Linear regression further established age as a key predictor of swim performance, enabling more precise team assignments.

A key outcome of this study is the development of a Shiny app that allows coaches to make data-driven team selections. The app enables users to upload swimmer data, visualise race results, and generate optimal team configurations through an interactive interface. Its adaptability means it can be expanded to accommodate different race formats and datasets, making it a practical tool for competition planning.

Future work could explore expanding the dataset to include multiple clubs or larger competitions, providing a broader perspective on performance optimisation. Additionally, integrating factors such as training history, swimmer health, and psychological readiness may offer deeper insights into performance variability. Enhancing the Shiny app with predictive models, real-time data updates, and customisable features would further empower coaches with greater control over team decisions, fostering improved performance outcomes in relay competitions.

5. Funding Acknowledgment

This work is partially based upon research supported by the South Africa National Research Foundation (NRF) CoE-MaSS #2023-021-STA-Sport. Opinions expressed and conclusions arrived at are those of the authors and are not necessarily to be attributed to the NRF.

References

Barnett, M., Doroudgar, S., Khosraviani, V., and Ip, E. (2022). Multiple comparisons: To compare or not to compare, that is the question. *Research in Social and Administrative Pharmacy*, **18** (2), 2331–2334.

Berkelaar, M. and Csárdi, G. (2023). lpSolve: Interface to lp_Solve v. 5.5 to solve linear/integer programs. *R package version 5.6.19*.

Chand, S., Singh, H., and Ray, T. (2018). Team selection using multi-/many-objective optimization with integer linear programming. *In 2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.

Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., and Borges, B. (2023). shiny: Web application framework for R. *R package version 1.7.5.1*.

Elkin, L., Kay, M., Higgins, J., and Wobbrock, J. (2021). An aligned rank transform procedure for multifactor contrast tests. *In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2021)*. 754–768.

Fabris-Rotelli, I., Sharp, G., Tudhope, A., van Staden, P., and Venter, M. (2022). A mathematical model to select the optimal age group classified freestyle relay teams for a masters swimming competition. *ORiON*, **38** (2), 95–105.

Gerber, H. and Sharp, G. (2006). Selecting a limited overs cricket squad using an integer programming model. *South African Journal for Research in Sport, Physical Education and Recreation*, **28** (2), 81–90.

Gokul, G. and Sundararaman, M. (2023). Determining the playing 11 based on opposition squad:

An IPL illustration. *Journal of Sports Analytics*, **9** (3), 1–13.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 65–70.

Kay, M., Elkin, L., Higgins, J., and Wobbrock, J. (2025). *ARTool: Aligned Rank Transform for nonparametric factorial ANOVAs*. R package version 0.11.2.

R Core Team (2025). R: A language and environment for statistical computing. Technical report, R Foundation for Statistical Computing, Vienna, Austria.

Sharp, G., Brettenny, W., Gonsalves, J. W., Lourens, M., and Stretch, R. (2011). Integer optimisation for the selection of a Twenty20 cricket team. *Journal of the Operational Research Society*, **62** (9), 1688–1694.

Sierksma, G. and Zwols, Y. (2015). *Linear and Integer Optimization: Theory and Practice*. CRC Press.

Wobbrock, J., Findlater, L., Gergle, D., and Higgins, J. (2011). The aligned rank transform for nonparametric factorial analyses using only Anova procedures. *In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2011)*. 143–146.

Wolsey, L. and Nemhauser, G. (1988). *Integer and Combinatorial Optimization*. John Wiley & Sons, Incorporated.

Submitted to the Proceedings of the 65th Annual Conference of the South African Statistical Association

Integrating PCA with random search for variable importance in multivariate stratified sample allocation

Georgi Borros, Şebnem Er and Sulaiman Salau

Statistical Sciences Department, University of Cape Town, Cape Town, South Africa

Multivariate stratification simultaneously partitions a heterogeneous population into more homogeneous subgroups based on multiple variables of interest. Once the subgroups are formed, the sample is allocated across strata, considering multiple outcome measures simultaneously. Multivariate stratified sampling therefore involves two optimisation problems: strata boundary determination and sample size allocation across strata. This study focuses on the allocation problem in multivariate stratified sampling, relevant to survey research with predetermined strata and multiple outcomes of interest, a common occurrence in large-scale surveys. An overarching optimal solution for this problem has not yet been established, as an allocation that optimises for one variable may not be optimal for the others, deemed a 'compromise' solution. A key characteristic of the compromise is an arbitrary or 'compromise' assignment of weights across outcome variables of interest. This study aims to contribute a more systematic weighting of the variables using principal component analysis. Additionally, we build on an established random search algorithm with modifications to enhance efficiency. The procedure generated generally produces more efficient estimates relative to a previous method and offers an intuitive approach for establishing variable importance weights.

Keywords: Multivariate, Sampling, Search Algorithm, Stratification.

1. Introduction

Multivariate stratification simultaneously partitions a heterogeneous population into more homogeneous subgroups based on multiple variables of interest. Once the subgroups are formed, the sample is allocated across strata, considering multiple outcome measures simultaneously. Multivariate stratified sampling therefore involves two key optimisation problems: strata boundary determination and sample size allocation across strata. The challenge lies in the fact that any configuration of strata affects the variance of the estimators and, ultimately, the precision achieved under the stratified sampling design.

The problem of strata boundary determination largely concerns stratifiers with an extensive range of potential cut points from which to form the strata, a particularly complex task for continuous stratification variables. Recent progress in the multivariate context for boundary determination is algorithmic and makes use of a novel grouping genetic algorithm methodology (O'Luing et al., 2018;

Corresponding author: Georgi Borros (georginaborros@gmail.com)
MSC2020 subject classifications: 62–06, 62D05

Ballin and Barcaroli, 2020). The univariate case for optimal boundary determination has also seen innovation including a biased random key genetic algorithm, variable neighbourhood search and greedy randomised adaptive search, with the opportunity to extend to the multivariate context (Brito et al., 2017, 2019, 2021).

Given a set of strata boundaries, optimal sample size allocation across strata is the next step to achieve precision. It is well known that equal and proportional allocation across strata can be suboptimal as strata variance is not accounted for in the allocation procedure (Neyman, 1934; Cochran, 1977; Er and Keskintürk, 2007). An optimal solution under univariate stratified sampling is provided in the seminal work of Neyman (1934), with a more generalised version of this solution presented by Cochran (1977). In the multivariate case, however, research remains ongoing. The difficulty in the multivariate context is that an optimal allocation with respect to one variable is not necessarily optimal for the others (Cochran, 1977; Khan and Ahsan, 2003; Varshney et al., 2015). The resulting allocation is considered a *compromise allocation* rather than the optimal allocation. To facilitate the compromise, a predetermined weight to denote variable importance can be used, variables can be mean-weighted or equal weights are implied when no weighting structure is specified (Mulvey, 1983; Khan et al., 2003; Kozak, 2006a,b; Brito et al., 2015; Gupta and Bari, 2017). Variable weighting in the multivariate context is therefore fairly arbitrary or focused on minimising average variance or the average coefficient of variation across estimators.

This paper aims to offer a new systematic approach to multivariate weighting in the allocation problem using principal component analysis (PCA) to weight stratification variables. The paper additionally builds on Kozak's (2006a) random search sample allocation method by applying modifications to enhance efficiency. The allocation methods presented in this paper are especially appropriate for cases where strata are already formed according to natural boundaries or simple categorical data, as the problem of optimal boundary determination falls away. These types of stratifiers with predetermined natural boundaries are commonly used in national surveys to achieve representation according to region and geography type (urban and rural) (StatsSA, 2024; UNICEF and NISA, 2023; FBoS, 2022). As such, this paper offers a new approach for practical implementation as well as an extension to the current literature on optimal stratified sampling allocation.

The following section presents the allocation optimisation problem and motivates for the inclusion of PCA to arrive at a more efficient compromise solution. Next, the original and enhanced search algorithms are presented and explained. Simulated results using the different objective functions and algorithms are then discussed and compared and the paper concludes with a discussion of limitations and areas for further study.

2. Optimisation problem

The multivariate allocation problem in stratified sampling concerns the optimal distribution of n sampled units across L strata with respect to K variables of interest, given a set of strata boundaries. The objective function for this problem has been conceptualised in various ways, all aiming to achieve optimal precision for the mean estimate for each variable of interest and often presented as a measure of variation (Cochran, 1977; Holmberg, 2003; Khan et al., 2003; Kozak, 2006a,b). In this paper, optimisation is considered in terms of minimising total survey variance, where we compare two variation-minimising objective functions and introduce the use of PCA.

Optimal multivariate allocation is therefore the choice of stratum sample size, n_h, that allows for the minimisation of variation with respect to $\mathbf{Y} = (\mathbf{Y_j}, ..., \mathbf{Y_K})$, where Y_k is the k^{th} survey variable of interest. Across objective functions, the variance of the mean for each estimator, $j = 1, ..., K$, with $h = 1, ..., L$ strata, is given as (Cochran, 1977):

$$V(\bar{Y}_{j\,st}) = \left\{ \sum_{h=1}^{L} W_h^2 \left(1 - \frac{n_h}{N_h}\right) \frac{S_{jh}^2}{n_h} \right\}, \tag{1}$$

where $V(\bar{Y}_{j\,st})$ is the estimated variance of the mean of Y_j under stratified sampling, N_h represents the total units in the population for stratum h, S_{jh}^2 is the variance of the mean of Y_j in stratum h and W_h is the proportion of the total population (N) in stratum h, such that $W_h = \frac{N_h}{N}$.

The following constraints are additionally applied to ensure that there are at least two sampled units per stratum and for the resulting solution to remain within the budgeted sample size, n (Kozak, 2006a):

$$2 \leq n_h \leq N_h, \tag{2}$$

and

$$\sum_{h=1}^{L} n_h = n. \tag{3}$$

2.1 Principal component analysis

PCA extracts the most important information from multi-dimensional data, making it an appealing choice for obtaining information from multivariate data. Principal components are calculated from the correlation matrix[1], $\boldsymbol{\rho}$, of $\boldsymbol{Y}$ with eigenvalue-eigenvector pairs $(\lambda_1, e_1), ..., (\lambda_K, e_K)$ where $\lambda_1 \geq \lambda_2 \geq ...\lambda_K \geq 0$. The first principal component has the largest variance, explaining the largest part of the data, given by (Johnson and Wichern, 2007):

$$PC_1 = \boldsymbol{e_1'Y} = e_{11}Y_1 + e_{12}Y_2 + ... + e_{1K}Y_K, \tag{4}$$

where

$$Var(PC_1) = \boldsymbol{e_1'\rho e_1} = \lambda_1, \tag{5}$$

and

$$Cov(PC_1, PC_{1+j}) = \boldsymbol{e_1'\rho e_{1+j}} = 0 \ \forall\ j = 1, ..., K-1. \tag{6}$$

From (4), it is seen that the coefficient vector or eigenvector, $\boldsymbol{e_1'} = [e_{11}, e_{12}, ..., e_{1k}]$, reflects the relative importance of each Y_k to the first principal component. This measure of relative importance demonstrates each variable's contribution to the largest portion of variance in the dataset (housed in the first principal component). We can therefore use $\boldsymbol{e_1'}$ as a vector of importance weights in the multivariate allocation problem, as these weights help to best explain what the allocation problem is trying minimise, total survey variance. To avoid negative weighting and retain the importance of the

[1] We use correlation PCA to accommodate variables with differing units of measurement (Abdi and Williams, 2010).

correlation regardless of direction, we consider the absolute value of $\boldsymbol{e_1'}$ and apply a transformation to ensure the weights sum to one[2]:

$$\boldsymbol{w} = \frac{\mid \boldsymbol{e_1'} \mid}{\sum_{k=1}^{K} e_{1k}}. \tag{7}$$

2.2 Objective functions

In this subsection we present two objective functions that can be used to for the optimisation problem. The first function uses an established method, while the second introduces a weighted approach that leverages PCA to assign variable importance.

2.2.1 Objective function 1

The first objective function considered aims to find the allocation across strata, $\mathbf{n}$=$n_1, ..., n_L$, that minimises the maximum variance of the K estimators (Kozak, 2006a):

$$f_1(\boldsymbol{n}) = \min\left(\max_{j,j=1,...,K}\left\{\sum_{h=1}^{L} W_h^2(1-\frac{n_h}{N_h})\frac{S_{jh}^2}{n_h}\right\}\right). \tag{8}$$

Since variables are not weighted according to any criteria, each Y_k is treated with equivalent importance.

2.2.2 Objective function 2

The second objective function uses a weighted sum of variance:

$$f_2(\boldsymbol{n}, \boldsymbol{w}) = \min\left(\sum_{j=1}^{K} w_j\left(\sum_{h=1}^{L} W_h^2(1-\frac{n_h}{N_h})\frac{S_{jh}^2}{n_h}\right)\right). \tag{9}$$

For the variable weights, $\boldsymbol{w} = (w_1, ..., w_K)$, PCA is leveraged using two approaches:

1. The first approach for $\boldsymbol{w}$ uses the principal component decomposition directly, with weights determined by the loadings of the first principal component of $\mathbf{Y}$, $\boldsymbol{e_1'}$:

$$\sum_{j=1}^{K} w_j = \boldsymbol{e_1'}. \tag{10}$$

2. In the second case, $\boldsymbol{w}$ is based on a random search process with principal component weights, $\boldsymbol{e_1'}$, given as the starting point. This approach aims to capture the potential for improvement based on adjustments to the initial principal component weights. The algorithm used for the search process is presented in the following section.

2.3 Algorithms

Kozak's (2006a) random search method has been shown to offer an efficient solution to the allocation problem in multivariate stratified sampling. The search procedure is easy to implement and a useful

[2]This can also be achieved by taking $\boldsymbol{e_1'}\boldsymbol{e_1}$, subject to slight adjustment to scale.

vehicle to test the objective functions presented in the previous section. We begin by presenting the search algorithm as it was initially constructed in Algorithm 1, given a total sample size n and L strata (Kozak, 2006a). The algorithm starts with an initial allocation based on proportional allocation from which the search function applies perturbations to reach a more optimal allocation[3]. The number of perturbations, R, is determined by the researcher and for this study we fixed R at 200 iterations.

Algorithm 1 Original Algorithm, A_1

Require: An initial allocation $\boldsymbol{n}$ based on proportional allocation ($n_h = [n\frac{N_h}{\sum_h N_h}]$), a fitness function f_t, L strata and R number of steps.

Ensure: Constraints (2) and (3) are upheld.

for $r \leq R$ **do** ▷ $r = 1, ..., R, r \in \mathbb{N}$

Randomly select two strata, L_1 and L_2.

Generate random number $j \in \mathbb{Z}, 1 \leq j \leq 5$

Generate $\mathbf{n}'$ by changing the sample allocation as follows:

$$n'_{L2} \leftarrow n_{L2} + j$$
$$n'_{L1} \leftarrow n_{L1} - j$$
$$n'_h \leftarrow n_h \; \forall \; h \neq \{L_1, L_2\}$$

if $f_t(\boldsymbol{n'}) \leq f_t(\boldsymbol{n})$ **then**

$\boldsymbol{n^r} \leftarrow \boldsymbol{n'}$

else if $f_t(\boldsymbol{n'}) > f_t(\boldsymbol{n})$ **then**

$\boldsymbol{n^r} \leftarrow \boldsymbol{n}$

end if

end for

return $\boldsymbol{n^R}$

Next, we consider a modified version of the search algorithm, A_2, as shown in Algorithm 2. The main modification concerns the selection of two strata, L_1 and L_2, which is no longer based on simple random selection but rather calculated with probability proportional to each stratum's contribution to total variation across the estimators of interest. This measure of variation is defined using $W_h S_h$, a weighted measure of the stratum standard deviated that is weighted by the size of the stratum. The motivation for using this measure of variation is its directly proportional relationship to variance as defined in Equation 1.

In Algorithm 2, L_1 is selected with probability proportional to stratum variation, $W_h S_h$. Therefore, the stratum contributing the most variation has the highest likelihood of being selected as L_1. The next stratum selected for perturbation, L_2, is selected with probability proportional to $\frac{1}{W_h S_h}$, where a stratum with lower variation is more likely to be selected as L_2. The selected strata are then perturbed by adding more sample to the stratum with likely higher variance (L_1), and less of the sample to

[3] Where the initial proportional allocation in Algorithm 1 does not provide an integer solution, we have rounded to integer sample sizes and allocated any remaining sampling units to the final stratum, L.

L_2. The aim here is to distribute the sample most efficiently by allocating a larger sample to higher variance strata and therefore reducing total variability.

A further change in Algorithm 2 is the size of the perturbation variable, j, which is modified to have a range anywhere between 1 and the sample size of the smallest stratum, $min(n_h)$. This change in range aims to reduce the chance of getting caught in a local optima by allowing the algorithm to potentially make bigger jumps in the search space than that of the fixed $j = 5$ in Algorithm 1. The range variable in Algorithm 2 is randomised rather than set as a larger fixed integer, as direct expansion of range increases the risk of overshooting optimal regions. Therefore, implementing a random selection of j in each iteration aims to balance the need to increase solution diversity and escape local optima while limiting the risk of poor convergence.

The modified procedure is therefore given as follows:

Algorithm 2 Modified Algorithm, A_2

Require: An initial allocation $\boldsymbol{n}$ based on proportional allocation ($n_h = [n\frac{N_h}{\sum_h N_h}]$), a fitness function f_t, L strata and R number of steps.

Ensure: Constraints (2) and (3) are upheld.

for $r \leq R$ **do** ▷ $r = 1, ..., R, r \in \mathbb{N}$

Select stratum L_1 proportional to $W_h S_h$.

Select stratum L_2 proportional to $\frac{1}{W_h S_h}$.

Generate random number $j \in \mathbb{Z}, 1 \leq j \leq min(n_h)$

Generate $\mathbf{n}'$ by changing the sample allocation as follows:

$$n'_{L2} \leftarrow n_{L2} - j$$
$$n'_{L1} \leftarrow n_{L1} + j$$
$$n'_h \leftarrow n_h \; \forall \; h \neq \{L_1, L_2\}$$

if $f_t(\boldsymbol{n'}) \leq f_t(\boldsymbol{n})$ **then**

$\boldsymbol{n^r} \leftarrow \boldsymbol{n'}$

else if $f_t(\boldsymbol{n'}) > f_t(\boldsymbol{n})$ **then**

$\boldsymbol{n^r} \leftarrow \boldsymbol{n}$

end if

end for

return $\boldsymbol{n^R}$

We lastly present a random search procedure that is followed in the case of random variable weights. In Algorithm 3, the choice of variable weights is included as part of the optimisation problem, using principal component weights as the starting point. This modification has been applied to test whether perturbations applied to the principal component weights might yield better results.

Algorithm 3 Weight Algorithm, A_3

Require: An initial weight $\boldsymbol{w} = \boldsymbol{e'_1}$, a fitness function with weights $f_t(\boldsymbol{n}, \boldsymbol{w})$, L strata, K variables and R number of steps.

Ensure: Constraints (2) and (3) are upheld.

for $r \leq R$ **do** ▷ $r = 1, ..., R, r \in \mathbb{N}$

Generate random number $v \in \mathbb{R}, 1 \leq v \leq min(\boldsymbol{w})$

Randomly select two variables, Y_p and Y_z ▷ $p, z \in [1 : K] \in \mathbb{N}$

Generate $\boldsymbol{w'}$ as follows:

$$w'_{Y_p} \leftarrow w_{Y_p} + v$$
$$w'_{Y_z} \leftarrow w_{Y_z} - v$$
$$w'_{Y_x} \leftarrow w_{Y_x} \; \forall \, x \neq \{p, z\}$$

if $f_t(\boldsymbol{n}, \boldsymbol{w'}) \leq f_t(\boldsymbol{n}, \boldsymbol{w})$ **then**

$\boldsymbol{w^r} \leftarrow \boldsymbol{w'}$

else if $f_t(\boldsymbol{n}, \boldsymbol{w'}) > f_t(\boldsymbol{n}, \boldsymbol{w})$ **then**

$\boldsymbol{w^r} \leftarrow \boldsymbol{w}$

end if

end for

return $\boldsymbol{w^R}$

3. Test set up

Based on the approaches outlined in the previous section, five methods are used across several datasets to explore the performance of the weighted objective functions and modified algorithm relative to the original random search procedure:

1. **RS**: Original random search method (Objective Function 1 (Subsection 2.2.1) and Algorithm 1) (Kozak, 2006a)
2. **PC_RS+**: Principal component weights with the modified search algorithm (Objective Function 2 (Subsection 2.2.2) and Algorithm 2)
3. **RW_RS+**: Random search weights with the modified search algorithm (Objective Function 2 (Subsection 2.2.2), Algorithm 2 and Algorithm 3)
4. **RS+**: Modified search algorithm with the original objective function (Objective Function 1 (Subsection 2.2.1) and Algorithm 2)
5. **PC_RS**: Principal component weights with the original random search algorithm (Objective Function 2 (Subsection 2.2.2) and Algorithm 1)

Each method is tested in cases with 2, 3, 4 and 5 variables of interest (K), with predetermined

strata boundaries according to the given *stratifier* and total sample size n[4]. The variable and dataset combinations used for each test are described in Table 1, where L refers to the total number of strata[5]. Here, $Y_1, ..., Y_k$ are the variables of interest for mean estimation using stratified sampling, given the predetermined boundaries according to L strata formed by the *stratifier*.

Table 1. Datasets, strata and variables of interest for precise mean estimation.

K	Dataset	Stratifier	$Y_1, ..., Y_K$	L	N	n
2	Agriculture Census 2002*	Province	Cereal area, potato area	16	1379311	20000
3	Agriculture Census 2002*	Province	Cattle count, pigs count, agriculture area	16	1140178	50000
4	Swiss Municipalities	Region	Municipality area, wood area, cultivation area, population total	6	2651	100
4	Boston Housing	Index of access to radial highways	Crime rate, residential land, property tax rate, nitric oxide concentration	9	506	100
4	Pima Diabetes	Number of pregnancies	Blood pressure, tricep skinfold thickness, glucose, BMI	10	710	100
5	Swiss Municipalities	Region	Municipality area, wood area, cultivation area, population total, mountain pasture area	6	2651	200
5	Boston Housing	Index of access to radial highways	Crime rate, residential land, property tax rate, nitric oxide concentration, pupil-teacher ratio	24	506	200
5	Pima Diabetes	Number of pregnancies	Blood pressure, tricep skinfold thickness, glucose, BMI, age	10	710	200
5	Child Health and Development Studies	Categorical variable on prior births and smoking	Birth weight, gestation period, mother age, mother height, mother weight	4	1174	200

*Simulated dataset based on parameters as in (Kozak, 2004, 2006a)

The datasets offer variety in terms of variable distribution, variable contribution to total variance, number of strata, subject area, population and sample size. The agriculture census data was simulated for both $K = 2$ and $K = 3$, as the authors did not have access to the original dataset. The simulations were based on the given stratification by province and corresponding strata means, standard deviations

[4] The n reported corresponds to the sample size used in the analysis. In some cases, this is a subsample of the dataset. All data used has been made publicly available for reproducibility.

[5] For example, where $K = 4$ for the Swiss Municipalities Dataset, *Region* is the stratifier, with 6 strata ($L = 6$) in total (corresponding to 6 regions).

and population sizes as supplied in the corresponding research (Kozak, 2004, 2006a). As the name suggests, these data consider agricultural measures such as livestock counts and land use area. The data for both $K = 2$ and $K = 3$ has a large sampling frame ($N > 20,000$), again in line with the original research conducted by Kozak (2004, 2006a).

To test cases where $K > 3$, commonly used and publicly available datasets were identified and included in the research. The first of these datasets is the Swiss municipalities dataset, which has been extensively used by other researchers in the stratification literature (Barcaroli et al., 2018; Ballin and Barcaroli, 2020). This dataset is municipality-level and the 'region' variable is used as the stratifier. Municipality-level variables that describe the nature and development of each region have been used as outcome variables. Notably, all of these outcome variables showcase extreme right skewness. The Boston housing data has been taken from a repository of datasets often used in machine learning (Blake and Merz, 1998; Leisch and Dimitriadou, 2024). These data also measure area-level indicators and are stratified based on the index of accessibility to radial highways, where the index ranges from 1-24 and corresponds to 9 unique index values or strata. The Boston dataset has the smallest sampling frame ($N = 506$) out of the datasets considered in this study. The Pima diabetes database shifts the subject matter focal area to health and is also taken from the machine learning repository (Leisch and Dimitriadou, 2024). The data are stratified based on the number of respondent pregnancies[6], resulting in $L = 10$. Again concerning health outcomes, the child health dataset measures child health indicators and is available for public use in a curated repository of health datasets (Rossi, 2024). This dataset corresponds to the fewest strata tested, $L = 4$.

Across the datasets, loadings for each Y_k on the first principal component, PC_1, are reported in Table 2. The loadings in this case reflect the coefficients of correlation between the variables and the first component (Abdi and Williams, 2010). The agriculture census data when $K = 2$ has the same loadings for each Y_k for PC_1. When $K = 3$, it is seen that Y_2 contributes slightly less to the variation in PC_1. The Swiss dataset indicates that Y_4 (population total) contributes the least to the variation in PC_1. The Boston data shows that Y_2 (residential land area) has an opposite relationship with PC_1, reflected by the negative correlation, in comparison to the other variables. The Pima data indicates that Y_5 (age) contributes the least to PC_1, while Y_4 (BMI) has the strongest relationship. Lastly, the child health data shows Y_1 and Y_4 (birth weight and mother height) have the strongest relationship while Y_3 (age) has the weakest relationship with PC_1.

For the respective dataset and K variable combinations, results were simulated over 50 runs. Within a run, each algorithm was allowed to iterate 200 times before reaching the stopping criteria. The relative performance of each method is measured by examining the ratio between the total average (aggregated over the 50 runs) variance of a given method and that of the original method, RS, calculated as follows for some method X:

$$R = \left(\frac{V_X}{V_{RS}}\right), \tag{11}$$

where V_X represents the average total variance for the mean estimates of $Y_1, ..., Y_k$ under method X across the 50 runs and V_{RS} represents the same measure of variance only it is derived when using the original random search method, RS.

[6] The range of pregnancies has been reduced to $[0, 9]$ to allow for variation in the dataset, as higher pregnancy numbers resulted in insufficient observations across strata.

Table 2. Variable loadings on PC_1 for each dataset and Y_k combination tested.

	K	Y_1	Y_2	Y_3	Y_4	Y_5
Agriculture Census	2	0.71	0.71	*NA*	*NA*	*NA*
Agriculture Census	3	0.65	0.44	0.62	*NA*	*NA*
Swiss Municipalities	4	0.61	0.63	0.44	0.18	*NA*
Boston Housing	4	0.46	-0.40	0.56	0.56	*NA*
Pima Diabetes	4	0.49	0.52	0.35	0.61	*NA*
Swiss Municipalities	5	0.57	0.55	0.29	0.09	0.53
Boston Housing	5	0.43	-0.40	0.53	0.49	0.37
Pima Diabetes	5	0.51	0.44	0.41	0.57	0.24
Child Health	5	0.53	0.39	0.10	0.54	0.51

In cases where $R > 1$, method X performed worse than *RS*, resulting in a larger total variance. $R = 1$ reflects equivalent performance between the methods. When $R < 1$, method X achieves lower variance than *RS*.

4. Results

Results are presented in Table 3 and Figure 1. The first two columns of Table 3 indicate the method, dataset and K combinations. The average variance over 50 runs for each variable Y_k is then presented as well as the total of the average variance across 50 runs for the K variables, $V(\boldsymbol{Y})$. Runtime in minutes is reflected in the *mins* column and relative performance in comparison to the original RS method is shown in the 'R' column[7]. Figure 1 offers a view of the spread of results across the 50 runs, a useful indication of consistency for each method.

In general, improvements in variance are observed relative to the original random search (RS) method, where the random weight (RW_RS+) method tends to most often achieve the lowest total variance. The size of the precision gain, however, tends to vary depending on the dataset and while efficiency gains are observed in terms of lower variance, computational cost also increases correspondingly.

Table 3. Results across 50 runs for each combination of dataset and K variables of interest for estimation.

Method	Data (k)	$V(Y_1)$	$V(Y_2)$	$V(Y_3)$	$V(Y_4)$	$V(Y_5)$	$V(\boldsymbol{Y})$	*mins*	R
PC_RS	Agri (2)	18.33	0.30	*NA*	*NA*	*NA*	18.62	0.02	1.000
PC_RS+	Agri (2)	15.79	0.28	*NA*	*NA*	*NA*	16.07	0.45	0.863
RS	Agri (2)	18.33	0.30	*NA*	*NA*	*NA*	18.63	0.03	1.000
RS+	Agri (2)	15.80	0.28	*NA*	*NA*	*NA*	16.08	0.45	0.863
RW_RS+	Agri (2)	15.78	0.28	*NA*	*NA*	*NA*	16.06	7.23	**0.862**

[7] All results have been generated using a MacBook Air (M2 Chip) with 8GB of RAM.

PC_RS	Agri (3)	0.00	0.19	130.22	*NA*	*NA*	130.42	0.03	1.000
PC_RS+	Agri (3)	0.00	0.18	124.77	*NA*	*NA*	124.96	0.57	0.958
RS	Agri (3)	0.00	0.19	130.22	*NA*	*NA*	130.42	0.03	1.000
RS+	Agri (3)	0.00	0.18	124.46	*NA*	*NA*	124.65	0.54	0.956
RW_RS+	Agri (3)	0.00	0.18	124.21	*NA*	*NA*	124.40	4.35	**0.954**
PC_RS	Swiss (4)	57027	4054	1348	672228	*NA*	734657	0.03	1.005
PC_RS+	Swiss (4)	56952	4050	1346	672421	*NA*	734769	0.28	1.005
RS	Swiss (4)	66973	4736	1551	658007	*NA*	731266	0.03	1.000
RS+	Swiss (4)	67152	4749	1555	657952	*NA*	731407	0.27	1.000
RW_RS+	Swiss (4)	62962	4479	1479	659945	*NA*	728865	2.14	**0.997**
PC_RS	Bost (4)	2.05	2.82	20.85	0.00	*NA*	25.72	0.03	**0.911**
PC_RS+	Bost (4)	0.60	3.50	25.34	0.00	*NA*	29.43	0.36	1.043
RS	Bost (4)	5.55	2.78	19.90	0.00	*NA*	28.23	0.03	1.000
RS+	Bost (4)	0.62	3.58	24.99	0.00	*NA*	29.18	0.36	1.034
RW_RS+	Bost (4)	0.56	3.47	25.64	0.00	*NA*	29.67	4.07	1.051
PC_RS	Pima (4)	2.98	2.12	8.64	0.52	*NA*	14.26	0.03	0.994
PC_RS+	Pima (4)	2.98	2.11	8.63	0.52	*NA*	14.25	0.42	**0.993**
RS	Pima (4)	3.12	2.15	8.54	0.53	*NA*	14.34	0.03	1.000
RS+	Pima (4)	3.10	2.15	8.53	0.53	*NA*	14.31	0.42	0.998
RW_RS+	Pima (4)	3.02	2.13	8.57	0.52	*NA*	14.24	5.52	**0.993**
PC_RS	Swiss (5)	25709	1835	616	326573	1572	356305	0.03	1.026
PC_RS+	Swiss (5)	25741	1834	614	326687	1574	356451	0.29	1.026
RS	Swiss (5)	32619	2306	755	309665	2096	347441	0.03	1.000
RS+	Swiss (5)	32624	2304	753	309834	2096	347612	0.28	1.000
RW_RS+	Swiss (5)	30433	2156	710	310865	1930	346093	3.00	**0.996**
PC_RS	Bost (5)	0.96	0.97	6.33	0.00	0.01	8.27	0.03	**0.756**
PC_RS+	Bost (5)	0.22	1.33	8.62	0.00	0.01	10.18	0.41	0.931
RS	Bost (5)	4.20	0.94	5.78	0.00	0.01	10.94	0.03	1.000
RS+	Bost (5)	0.23	1.40	8.60	0.00	0.01	10.24	0.41	0.936
RW_RS+	Bost (5)	0.23	1.29	8.61	0.00	0.01	10.14	4.56	0.927
PC_RS	Pima (5)	1.24	0.89	3.60	0.22	0.35	6.29	0.03	**0.991**
PC_RS+	Pima (5)	1.24	0.89	3.60	0.21	0.35	6.29	0.46	**0.991**
RS	Pima (5)	1.31	0.90	3.56	0.22	0.35	6.35	0.03	1.000
RS+	Pima (5)	1.31	0.90	3.56	0.22	0.35	6.35	0.46	1.000
RW_RS+	Pima (5)	1.25	0.89	3.59	0.22	0.35	6.29	5.51	0.992
PC_RS	Child (5)	1.31	1.05	0.12	0.03	1.74	4.25	0.03	**0.996**
PC_RS+	Child (5)	1.31	1.05	0.12	0.03	1.74	4.25	0.21	**0.996**
RS	Child (5)	1.33	1.06	0.12	0.03	1.73	4.27	0.03	1.000
RS+	Child (5)	1.33	1.06	0.12	0.03	1.73	4.27	0.21	1.001
RW_RS+	Child (5)	1.31	1.05	0.12	0.03	1.74	4.25	1.62	**0.996**

In the bivariate case, $K = 2$, simulated Agriculture Census 2002 data has been used. Consequently, the simulated dataset provided a large sampling frame (N=1379311) from which a consistently large sample was drawn (n=20000). The outcome variables of interest were also relatively normally distributed in comparison to the other datasets considered. In this case, the methods using the modified search algorithm displayed improvements in variance in comparison to the original RS method, with RW_RS+, PC_RS+ and RS+ achieving an average variance of approximately 14.7% less than that of RS. Figure 1 gives an indication of the narrow variation in performance across runs, further supporting the notable improvement observed. Here, the increased search space and targeted perturbations offered by the modified algorithm seem to allow for precision gains. This result could likely be due to the larger N and n enabled by the simulated dataset, where the smaller search space of the RS method could risk settling in a local optima.

When $K = 3$, similar performance patterns to the case of $K = 2$ are observed, however the change in precision across methods is of a smaller magnitude. The modified search algorithms offer an average precision gain of approximately 4.2-4.6% relative to RS. The principal component weights combined with the original search (PC_RS) achieve the same result as RS. Again, the relative improvement in performance offered by the modified algorithm could be due to the size of the dataset (N=1140178) and the expansion of the search space. It is important to note that while the random weight method (RW_RS+) displays gains in improvement, it has a substantially longer average runtime of 4.35 seconds in comparison to RS with 0.03 seconds. The other modified search algorithms are more computationally expensive than RS but less so than RW_RS+, with PC_RS+ and RS+ having average run times of 0.57 and 0.54 seconds respectively.

To test cases with more variables than those studied in Kozak 2006, new datasets are considered.

In instances when $K = 4$, results differ subject to the dataset. The highly skewed Swiss dataset results in similar average performance across methods, with RW_RS+ showing an almost negligible improvement relative RS of 0.3%. That being said the distribution of results for the Swiss dataset observed in Figure 1 show that RW_RS+ is a consistent performer with few outliers. The Boston housing data shows the modified algorithm performing relatively worse than the original algorithm, while the use of PC weights leads to the most precision out of all the methods with PC_RS achieving an average variance 8.9% lower than that of RS. Figure 1 further indicates the tight distribution of results achieved for the Boston data under PC_RS. The Pima Diabetes data leads to more favourable results for combinations using the modified algorithm, supported in Figure 1 although the relative precision gains in comparison to RS are marginal.

The case of $K = 5$ tends to show gains when using a weighted objective function relative to RS, however the magnitude continues to differ depending on dataset. The Boston Housing data shows the largest change in performance when using PC weights. PC_RS achieved 24.4% lower average variance than that of RS. Unlike when $K = 4$, the modified algorithms also show improvements (smaller than those seen in PC_RS) relative to RS. In line with previous results, the RW_RS+ method on the Swiss dataset shows a minor relative improvement of 0.4%. The Pima dataset has both the principal component weight methods (PC_RS and PC_RS+) performing equivalently with the lowest average variance (0.9% less than RS). Finally, the Child Health dataset shows all weighted methods (PC_RS, PC_RS+, RW_RS+) performing marginally better (0.4%) than RS.

Relative to the large Agriculture datasets, the average runtime for RW_RS+ has reduced to 1.84 seconds (from 4.35 and 7.23 seconds), however this remains markedly higher than RS at 0.02 seconds.

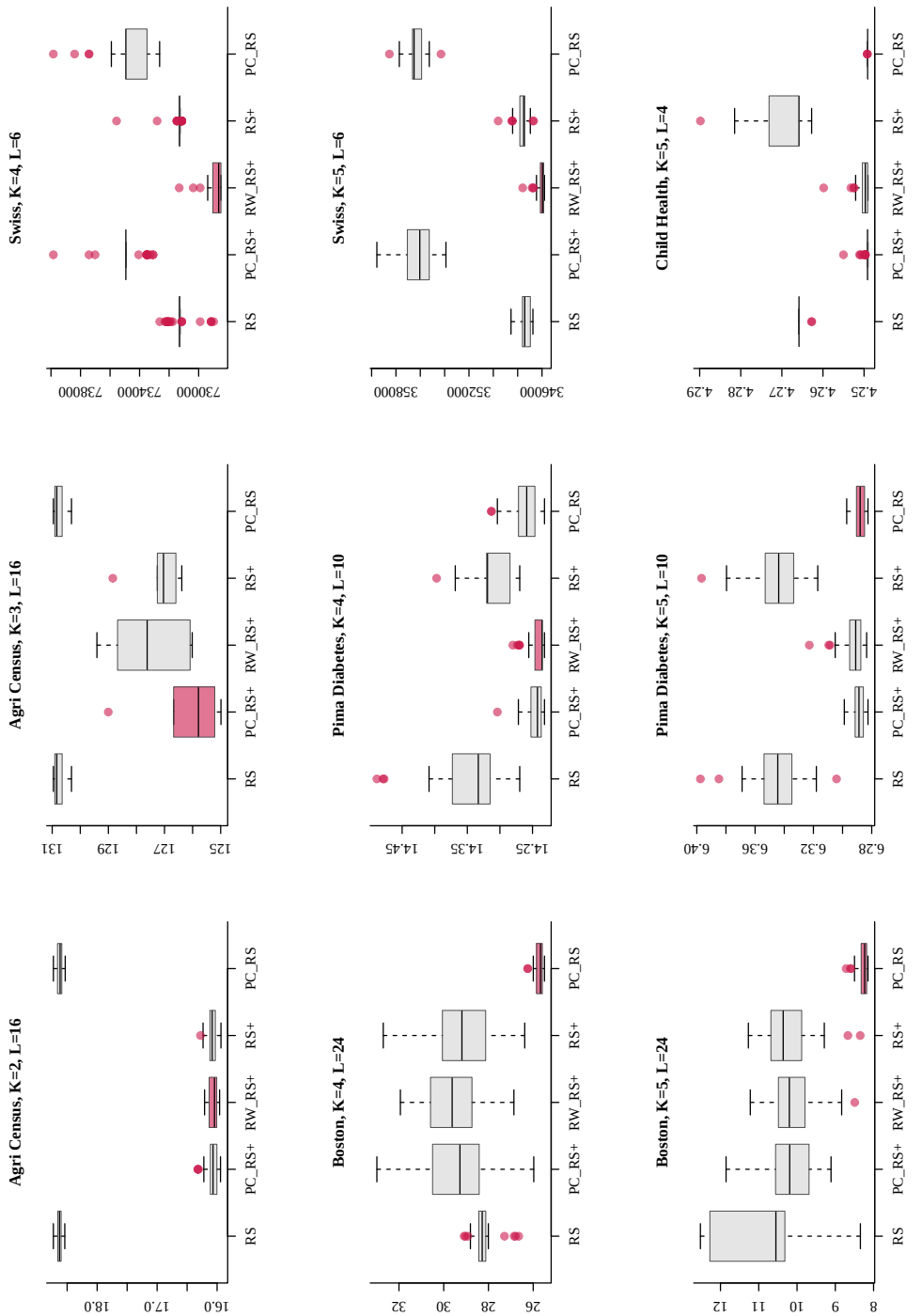


Figure 1. Total variance achieved according to each method, dataset and $Y_1, ..., Y_k$ combination across the 50 test runs.

Table 4 provides a summary of each method's performance across the 9 datasets and variable combinations. The results reflect that the weighted methods and the modified algorithm tend to most often achieve lower total variance in comparison to the original RS method ($R < 1$). The magnitude of these gains are, however, largely dataset dependent and seem to vary with factors such as sample frame size, variable loadings on PC_1 and dataset distribution. The modified search algorithm when combined with random search weights is seen to outperform RS in 8 out of the 9 dataset and variable combinations tested, although this result is accompanied by a trade off in terms of computational efficiency.

Table 4. Summary of method performance relative to the original random search procedure (RS).

Method	$R > 1$	$R = 1$	$R < 1$
PC_RS	2	2	5
RS+	3	2	4
PC_RS+	3	0	6
RW_RS+	1	0	8

5. Conclusion

This study provides a new method for variable importance weighting in the multivariate stratified sampling allocation problem, integrating Principal Component Analysis (PCA) and an enhanced random search algorithm. Prior to this contribution, variable weighting for the multivariate allocation problem is generally either not applied (resulting in implicitly equally weighted outcome variables) or based on some predetermined level of importance allocated by the survey practitioner (Cochran, 1977; Kozak, 2006a). The PCA approach leverages each outcome variable's loading on the first principal component, which provides a linear combination that explains most of the variance in the dataset or, in this case, sampling frame (Abdi and Williams, 2010). The weighted method is accompanied by a modified search algorithm, which targets strata according to variation and modifies the search space relative to the original search algorithm of Kozak (2006a), aiming to reach the minimum (optimum) total variance under a stratified sampling design.

Based on the datasets, strata and sample size combinations tested in this study, it was found that RW_RS+, the random weight method (using principal component weights as the starting point followed by an iterative search procedure) with the modified search algorithm, most often (in 8 out of 9 dataset combinations tested) achieved lower average total variance relative to the original random search method. This method, however, was the most computationally expensive as it uses a nested search procedure. The next top performing method in terms of variance reduction was PC_RS+, principal component weights combined with the modified search algorithm. This method achieved lower total variance relative to the original method in 6 out of the 9 test conditions and the compute was less expensive than RW_RS+, with runtime less than one minute on average.

Overall, the approach to multivariate stratified sampling allocation proposed in this study generally achieves similar or lower total variance relative to earlier research and offers an intuitive approach to variable weighting in the multivariate context. An important limitation of this study is the assumption of predetermined strata boundaries, as the optimal allocation is fundamentally dependent

on the choice of strata boundaries (Dalenius and Hodges, 1959). Future research should explore the joint optimisation of boundary determination and allocation to establish the overarching optimal solution for a given stratified sampling procedure (Khan and Sharma, 2015).

References

Abdi, H. and Williams, L. J. (2010). Principal Component Analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, **2**.

Ballin, M. and Barcaroli, G. (2020). R package SamplingStrata: new developments and extension to Spatial Sampling. ArXiv:2004.09366 [stat].
URL: *http://arxiv.org/abs/2004.09366*

Barcaroli, G., Ballin, M., Pagliuca, D., Willighagen, E., and Zardetto, D. (2018). Package 'SamplingStrata'.

Blake, C. L. and Merz, C. J. (1998). *UCI Repository of Machine Learning Databases*. University of California, Irvine, Department of Information and Computer Sciences, Irvine, CA.

Brito, J., De Lima, L., Henrique González, P., Oliveira, B., and Maculan, N. (2021). Heuristic approach applied to the optimum stratification problem. *RAIRO - Operations Research*, **55** (2), 979–996. doi:10.1051/ro/2021051.
URL: *https://www.rairo-ro.org/10.1051/ro/2021051*

Brito, J., Silva, P., Semaan, G., and Maculan, N. (2015). Integer programming formulations applied to optimal allocation in stratified sampling. *Survey Methodology*, **41**, 427–442.

Brito, J., Silva, P., and Veiga, T. (2017). R package stratvns: Optimal Stratification in Stratified Sampling.

Brito, J., Veiga, T., and Silva, P. (2019). An optimisation algorithm applied to the one-dimensional stratification problem. *Statistics Canada*.

Cochran, W. G. (1977). *Sampling Techniques*. John Wiley & Sons, Inc.

Dalenius, T. and Hodges, J. L. (1959). Minimum Variance Stratification. *Journal of the American Statistical Association*, **54** (285), 88–101. doi:10.2307/2282141. Publisher: [American Statistical Association, Taylor & Francis, Ltd.].
URL: *https://www.jstor.org/stable/2282141*

Er, Şebnem. and Keskintürk, T. (2007). A genetic algorithm approach to determine stratum boundaries and sample sizes of each stratum in stratified sampling. *Computational Statistics & Data Analysis*, **52** (1), 53–67. doi:10.1016/j.csda.2007.03.026.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S0167947307001363*

FBoS (2022). United Nations Children's Fund (UNICEF) Fiji Multiple Indicator Cluster Survey 2021. Technical report, UNICEF, Suva, Fiji.

Gupta, N. and Bari, A. (2017). Fuzzy multi-objective optimization for optimum allocation in multivariate stratified sampling with quadratic cost and parabolic fuzzy numbers. *Journal of Statistical Computation and Simulation*, **87** (12), 2372–2383. doi:10.1080/00949655.2017.1332195.

Holmberg, A. (2003). *Essays on Model Assisted Survey Planning*. Ph.D. thesis, Uppsala University, Uppsala.

Johnson, R. A. and Wichern, D. W. (2007). *Applied multivariate statistical analysis*. 6th ed edition. Pearson Prentice Hall, Upper Saddle River, N.J. OCLC: ocm70867129.

Khan, M. and Ahsan, M. (2003). A note on optimum allocation in multivariate stratified sampling. *The South Pacific Journal of Natural and Applied Sciences*, **21**, 91–95. doi:10.1071/SP03017.

Khan, M., Khan, E., and Ahsan, M. (2003). Theory & Methods: An Optimal Multivariate Stratified Sampling Design Using Dynamic Programming. *Australian & New Zealand Journal of Statistics*, **45** (1), 107–113. doi:10.1111/1467-842X.00264.

Khan, M. G. and Sharma, S. (2015). Determining optimum strata boundaries and optimum allocation in stratified sampling. *Aligarh Journal of Statistics*, **35**, 23–40.

Kozak, M. (2004). Method of multivariate sample allocation in agricultural surveys. *In Colloquium Biometryczne*, volume 34. -.

Kozak, M. (2006a). Multivariate Sample Allocation: Application of Random Search Method. *Statistics in Transition*, **7**, 889–900.

Kozak, M. (2006b). On Sample Allocation in Multivariate Surveys. *Communications in Statistics - Simulation and Computation*, **35** (4), 901–910. doi:10.1080/03610910600880286.
URL: *https://www.tandfonline.com/doi/full/10.1080/03610910600880286*

Leisch, F. and Dimitriadou, E. (2024). *mlbench: Machine Learning Benchmark Problems*.
URL: *https://CRAN.R-project.org/package=mlbench*

Mulvey, J. M. (1983). Multivariate Stratified Sampling by Optimization. *Management Science*, **29** (6), 715–724. doi:10.1287/mnsc.29.6.715.
URL: *https://pubsonline.informs.org/doi/10.1287/mnsc.29.6.715*

Neyman, J. (1934). On the Two Different Aspects of the Representative Method: The Method of Stratified Sampling and the Method of Purposive Selection. *Journal of the Royal Statistical Society*, **97** (4), 558–625.

O'Luing, M., Prestwich, S., and Tarim, S. A. (2018). A Grouping Genetic Algorithm for Joint Stratification and Sample Allocation Designs. ArXiv:1709.03076 [stat].
URL: *http://arxiv.org/abs/1709.03076*

Rossi, R. C. (2024). MedDataSets: Comprehensive Medical, Disease, Treatment, and Drug Datasets.
URL: *https://github.com/lightbluetitan/meddatasets*

StatsSA (2024). Quarterly Labour Force Survey 2024: Q1 [dataset]. doi:https://doi.org/10.25828/q1dh-zw53.

UNICEF and NISA (2023). United Nations Children's Fund (UNICEF) Afghanistan Multiple Indicator Cluster Survey 2022-23. Technical report, UNICEF, Kabul, Afghanistan.

Varshney, R., Khan, M. G. M., Fatima, U., and Ahsan, M. J. (2015). Integer compromise allocation in multivariate stratified surveys. *Annals of Operations Research*, **226** (1), 659–668. doi:10.1007/s10479-014-1734-z.
URL: *http://link.springer.com/10.1007/s10479-014-1734-z*

Submitted to the Proceedings of the 65th Annual Conference of the South African Statistical Association

Quantifying the directional relationship between elliptical natural hydrogen depressions and geological lineaments in Mpumalanga, South Africa

Calvin J. Botha[1], *Ansie Smit*[2], *Adam J. Bumby*[2], *Inger Fabris-Rotelli*[1], *Brenda O. Mac'Oduol*[1] *and Najmeh Nakhaeirad*[1]

[1]Department of Statistics, University of Pretoria, Pretoria, South Africa
[2]Department of Geology, University of Pretoria, Pretoria, South Africa

The global energy landscape is currently undergoing several profound changes, developing some paths towards accessible renewable resources. Natural hydrogen has emerged as a promising energy source, yet its formation in different geological environments remains uncertain. This research explores the directional relationship between seepages of natural hydrogen, known as sub-circular depressions, and geological lineaments in Mpumalanga, South Africa. The methodology employs circular statistics to analyse depression orientations and uses rose diagrams for data visualisation. Comparisons between the angles of depressions and geological lineaments are conducted using a proposed novel algorithm for data set division, followed by K-means clustering to classify the depressions by size. Hypothesis testing, tailored for circular data, is applied through the single-sample likelihood ratio test, along with the Watson-Williams and the Watson-Wheeler tests. Initial findings suggest that the average orientation between the depressions and geological lineaments in the study area differs significantly from zero. Additionally, the Watson-Wheeler test indicates that the three-cluster solutions exhibit no significant directional differences when comparing individual clusters, while the four-cluster solution reveals substantial variation in angular distributions. The findings provide a foundation for understanding these interactions and open the door to exciting possibilities for future research and exploration, potentially revolutionising the way we harness and study the implementation of natural hydrogen as a renewable energy.

Keywords: Natural Hydrogen, Directional Statistics, Renewable Energy.

1. Introduction

The global energy landscape is undergoing a period of ground-breaking development as we shift towards renewable resources, driven by the need to mitigate the side effects of non-renewable energy sources such as coal and oil (Gielen et al., 2019). Various agreements, such as the Paris Agreement of 2015 (Delbeke et al., 2019), have been put in place to combat the global issue of climate change. Despite these efforts, many countries are struggling to meet their targets, highlighting the ongoing challenge of transitioning to sustainable energy sources.

Corresponding author: Ansie Smit (ansie.smit@up.ac.za)
MSC2020 subject classifications: 62P99

Natural hydrogen has emerged as a promising alternative to non-renewable energy. This form of hydrogen is produced through geological processes and is considered a clean, renewable potential source of energy. Surface seepages of natural hydrogen are associated globally with sub-circular depressions that typically present as shallow, elliptically shaped, vegetation-free depressions on the landscape which are easily identifiable via satellite imagery (Moretti et al., 2022). Research indicates that depressions are frequently aligned with geological lineaments, suggesting a deep subsurface connection (Xiang et al., 2020).

Ongoing studies in South Africa are investigating the presence of natural hydrogen in sub-circular depressions identified in Mpumalanga. These depressions are typically filled with water and are colloquially referred to as pans[1]. The pans are identified on satellite imagery using Geographic Information System (GIS) techniques, while geological lineaments are determined using magnetic and other geophysical measurements.

Unlike traditional statistics, which deals with data on a linear scale, the data present in this research is of a directional and angular nature. Circular data arises whenever measuring direction is of interest and is usually expressed as angles relative to a fixed reference point, such as due north (Lee, 2010). Examples of where circular data is present include wind directions (e.g., Carnicero et al., 2013), animal movement (e.g., Rivest et al., 2016), and fault orientation (e.g., Michalak et al., 2021).

Due to the nature of circular data, conventional statistical methods used for linear numerical data are inadequate (Lee, 2010). Rose diagrams are generally constructed to detect deviations from a random or uniform distribution of directional data (Walsh and Martill, 2006). Additionally, they provide insight into the structural characteristics inherent in the data (Wells, 2000). These diagrams will be used to represent relationships between the orientations of the pans and the geological lineaments.

The von Mises distribution plays a similar role to the normal distribution in circular data analyses (Mardia and Jupp, 2009). This study investigates whether the angles between the major axes of pans and geological lineaments differ significantly, aiming to assess a potential directional relationship. Using the von Mises distribution, the null hypothesis, that the mean angular difference deviates from zero, is tested against the alternative. The test statistic is computed based on the fitted von Mises distribution, with large values indicating rejection of the null hypothesis, thus suggesting a meaningful directional relationship (Mardia and Jupp, 2009).

The study employs statistical tools to evaluate whether significant directional relationships exist between sub-circular depressions and nearby geological lineaments and proposes a novel algorithm for this purpose. To categorise the depressions, particularly by size, K-means clustering is applied as an unsupervised method, reducing subjectivity by grouping data based on similarities. The elbow method determines the optimal number of clusters (K), which serves as the basis for statistical comparisons. Two hypothesis tests are then employed: the Watson-Williams test to assess differences in mean directions across clusters, and the Watson-Wheeler test, a non-parametric hypothesis test, which does not rely on strict parametric assumptions (Mardia and Jupp, 2009; Sinaga and Yang, 2020). These methods provide a framework to understand how depressions may align with geological lineaments, contributing to insights into their formation and spatial relationships.

This study aims to contribute to the growing body of research on natural hydrogen by statistically

[1] UP scientists lead Mpumalanga study of natural hydrogen gas discovered under Earth's surface. (2023, December 1). Squared[2] UP Newsletter. Retrieved January 16, 2025, from https://www.up.ac.za/media/shared/11/ZP_NewsImages/squaredup_december-2023.zp245474.pdf

evaluating the directional relationship between sub-circular depressions (pans) and geological lineaments in Mpumalanga, South Africa. By leveraging circular statistics, including rose diagrams and the von Mises distribution, the study assesses whether significant alignment exists between these features. Additionally, K-means clustering categorises the pans by size, and hypothesis tests determine whether directional differences exist across clusters. The paper is structured as follows: Section 2 outlines the methodology, detailing data collection, transformation, and statistical techniques. Section 3 presents the results, followed by an analysis and discussion. Finally, Section 4 summarises key findings, acknowledges limitations, and suggests directions for future research.

2. Methodology

2.1 Visualisation and Data Transformation

Rose diagrams depict relationships between two variables: direction or angle around a circle and a scalar quantity representing distance from the centre (Sanderson and Peacock, 2020). Angles are typically represented in degrees or radians, with radians ensuring compatibility with statistical software like `R` (Ihaka and Gentleman, 1996). Therefore, all angles in this paper are converted to radians. Built-in functions such as `atan2` and `rose` facilitate direction calculation and rose diagram creation (Pewsey et al., 2013).

Let the pans be denoted by p_i for $i = 1, 2, ..., n$ and the geological lineaments by g_j for $j = 1, 2, .., m$, where n is the number of pans and m represents the number of geological lineaments. The angles for each p_i will then be given by θ_{p_i} and for each g_j by θ_{g_j}. Each geological lineament g_j is a line pattern such that the single line displayed consists of an arrangement of smaller lines that together form a specific pattern or trend in the data set. These line patterns will later be linked to specific pans which will be denoted by g_{ij}. The quantity g_{ij} represents the line pattern of the j^{th} geological lineament near pan p_i.

Circular statistics considers observations which are directions or angles. The directions can be represented as points on the unit circle where the choice of the initial direction and orientation for the unit circle is important (Mardia and Jupp, 2009). A conventional method is choosing the rightward direction as the initial starting point, known as the East direction. This aligns with the x-axis of the standard Cartesian coordinate system. A useful way of representing each point $\boldsymbol{x}$ on a unit circle involves using an angle θ and unit complex numbers z (Mardia and Jupp, 2009). The relation between $\boldsymbol{x}$, θ, and z is given by

$$\boldsymbol{x} = (\cos\theta, \sin\theta)^T \quad \text{and} \quad z = e^{i\theta} = \cos\theta + i\sin\theta. \tag{1}$$

Figure 1, shows a unit circle with a radius of 1 with direction $\boldsymbol{x}$ (from origin O to point z) represented by angles θ, measured anticlockwise from the East. It identifies z as a point on the unit circle corresponding to θ, calculated using Equation 1. The coordinates of z, $\cos\theta$ and $\sin\theta$, define the x and y positions on the circle's circumference. This approach, which uses angles and complex numbers to represent directions, is similarly applied to the pans in this study.

When analysing spatial data, applying the correct projection is essential. For this study, the Hartebeesthoek 94 projection, a geodetic reference system used in South Africa (Wonnacott, 1999), was chosen. This projection ensures accurate calculation of lengths and angles, allowing precise representation of spatial relationships between lineaments and pans.

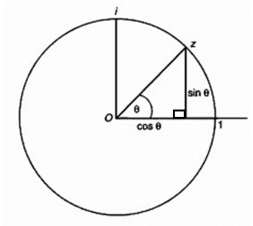


Figure 1. Representation of vector $\boldsymbol{x}$ using angle θ and complex number z (Mardia and Jupp, 2009).

(a) (b)

Figure 2. Calculation of angles of major axis of the two typical sub-circular depressions.

To calculate angles, a consistent directional measure is required. Each pan's orientation is defined by its major axis, determined as the maximum Euclidean distance between two boundary points of the pan. As these pans are elliptical in nature, we can refer to their major axis for this purpose. The major axis is represented as $y = mx + c$, where m is the slope and c is the intercept. The angle θ of the major axis is calculated via the slope m using the arc-tangent function: $\theta = \tan^{-1}(m)$, where θ is measured anticlockwise from the horizontal axis. Angles are constrained to $[0, \pi]$ because $\frac{\pi}{2}$ and $\frac{3\pi}{2}$ describe the same axis. Applying this to two randomly selected pans produces the results in Figure 2. Note that the example in Figure 2b illustrates a scenario where the angle exceeds $\frac{\pi}{2}$.

2.2 Circular Statistics Methods

For the calculation of the circular mean (denoted by $\bar{\theta}$), assume that $x_1, x_2, ..., x_n$ are the observed unit vectors with angles $\theta_1, \theta_2, ..., \theta_n$ respectively. The Cartesian coordinates of each unit vector x_j is represented as $(\cos\theta_j, \sin\theta_j)$ for $j = 1, 2, ..., n$ (Mardia and Jupp, 2009). The corresponding Cartesian coordinates of the circular mean (centre of mass) are $(\bar{C},\bar{S})$ where

$$\bar{C} = \frac{1}{n}\sum_{j=1}^{n}\cos\theta_j \quad \text{and} \quad \bar{S} = \frac{1}{n}\sum_{j=1}^{n}\sin\theta_j. \tag{2}$$

These quantities represent the components of the circular mean, which summarises the directional tendencies of the observed angles on the unit circle. The length of the centre of mass vector $\boldsymbol{x}$ is

$$\bar{R} = \sqrt{\bar{C}^2 + \bar{S}^2} \quad \text{where} \quad 0 \leq \bar{R} \leq 1,$$

which is analogous to the Euclidean distance formula used in linear statistics (Mardia and Jupp, 2009). The quantity $\bar{R}$ is a measure of concentration. $\bar{R}$ will be close to 1 when the angles are closely situated or 0 when the angles are widely spread out. Note that for calculating the mean direction $\bar{\theta}$, it

is assumed that $\bar{R} > 0$, resulting in the mean direction being calculated using the following formula:

$$\bar{\theta} = \begin{cases} \tan^{-1}(\frac{\bar{S}}{\bar{C}}) & \text{if } \bar{S} > 0, \bar{C} > 0, \\ \tan^{-1}(\frac{\bar{S}}{\bar{C}}) + \pi & \text{if } \bar{C} < 0, \\ \tan^{-1}(\frac{\bar{S}}{\bar{C}}) + 2\pi & \text{if } \bar{S} < 0, \bar{C} > 0, \end{cases} \tag{3}$$

where the inverse tangent function takes on values in the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$.

In addition, comparisons can be made using measures of dispersion along a line. The simplest of these is the sample circular variance, defined by Mardia and Jupp (2009) as

$$V = 1 - \bar{R}, \quad \text{where} \quad 0 \leq V \leq 1.$$

Other measures outlined by Mardia and Jupp (2009), such as the circular standard deviation and circular dispersion, while less pivotal for this study, offer additional insights into the distribution characteristics. The final set of descriptive statistics involves measures of skewness and kurtosis, providing insights into the distribution of the angles around the mean direction. As detailed by Bekker et al. (2022), the skewness coefficient is given by

$$\hat{s} = \frac{\hat{\gamma_2}}{(1-\bar{R})^{3/2}} \quad \text{where} \quad \hat{\gamma_2} = \frac{\bar{S}_2(\bar{C}^2 - \bar{S}^2) - 2\bar{C}\bar{C}_2\bar{S}}{\bar{R}^2} \quad \text{with} \quad \bar{C}_2 = \frac{1}{n}\sum_{j=1}^{n}\cos(2\theta_j),$$

where $\hat{\gamma_2}$ represents the second cosine moment about the mean direction $\bar{\theta}$. The kurtosis coefficient is derived as

$$\hat{k} = \frac{(\hat{\gamma_1} - \bar{R}^4)}{(1-\bar{R})^2} \quad \text{where} \quad \hat{\gamma_1} = \frac{\bar{C}_2(\bar{C}^2 - \bar{S}^2) + 2\bar{C}\bar{S}\bar{S}_2}{\bar{R}^2} \quad \text{with} \quad \bar{S}_2 = \frac{1}{n}\sum_{j=1}^{n}\sin(2\theta_j),$$

where $\hat{\gamma_1}$ is the second sine moment about the mean direction $\bar{\theta}$.

These measures offer insight into the distribution's asymmetry and tail lengths. They complement the visual information provided by the rose diagrams, offering a deeper understanding of the angular distribution characteristics observed in the study. While further investigation could involve fitting a circular distribution to the data, this falls outside the scope of this research.

2.3 Proposed Algorithm for Geological Lineaments Comparison

Analysing the directional statistics of the pans in isolation, while valuable, does not provide a complete understanding of the research questions. A similar analytical approach to establishing the relationship between the orientation of geological lineaments in relation to the direction of observed pans is followed for the geological lineaments. The challenge lies in integrating these two sets of data in a statistically robust manner to derive meaningful insights.

For simplicity, each pan is assumed to be independent, meaning factors such as orientation and proximity of pans do not influence others. This assumption should be revisited in future research. Unlike the pans, geological lineaments follow varied and irregular patterns beneath the surface. Some may lie directly under the pans, while others are nearby, making the determination of their

orientations more complex. The difference in orientations between pans and geological lineaments is critical to establish the underlying relationship between these two.

The method proposed is to create a buffer structure that is placed over the two sets of data, with one being the pans (P) and the other the geological lineaments (G). The process can be summarised in the following algorithm.

1. Initialisation: Set the initial scaling parameter $a = 1$, where a determines the initial buffer size in metres around each pan.
2. Create Bounding Boxes: For each pan $p_i \in P$:
   (a) Obtain the outer boundary of each pan p_i.
   (b) Calculate the initial bounding box around p_i with a buffer of a metres. In construction of the buffer it was chosen that it would increase uniformly in all directions resulting in a rounded shape around the original polygon, even if the pan has an irregular shape.
3. Increase the Scaling Parameter: Increase a by integer values until every bounding box contains at least one geological lineament $g_j \in G$. The buffer should be adjusted evenly around the entire perimeter of p_i, so that larger pans will naturally have proportionally larger bounding boxes.
4. Evaluate Directions:
   (a) For each bounding box around pan p_i, extract all lineament segments such that these segments fall entirely within the bounding box, denoted by g_{ij}.
   (b) The direction of each lineament is calculated as a single angle $\theta_{ij} \in [0, \pi)$, determined from the orientation of the line based on its start and end coordinates within the bounding box.
   (c) For each pan p_i, compute the following using all angles θ_{ij} relevant to its bounding box:

$$\bar{C}_i = \frac{1}{n(i)} \sum_{j=1}^{n(i)} \cos(\theta_{ij}), \quad \bar{S}_i = \frac{1}{n(i)} \sum_{j=1}^{n(i)} \sin(\theta_{ij}),$$

   where $n(i)$ is the number of lineaments within the bounding box of pan p_i.
   (d) Compute the average direction $\bar{\theta}_i$ for all geological lineaments in each bounding box using:

$$\bar{\theta}_i = \begin{cases} \tan^{-1}\left(\frac{\bar{S}_i}{\bar{C}_i}\right) & \text{if } \bar{S}_i > 0, \bar{C}_i > 0, \\ \tan^{-1}\left(\frac{\bar{S}_i}{\bar{C}_i}\right) + \pi & \text{if } \bar{C}_i < 0, \\ \tan^{-1}\left(\frac{\bar{S}_i}{\bar{C}_i}\right) + 2\pi & \text{if } \bar{S}_i < 0, \bar{C}_i > 0. \end{cases}$$

   The resulting $\bar{\theta}_i$ represents the overall average directional trend of all the geological lineaments in the bounding box around pan p_i.
5. Store the Directional Information: For each pan p_i, store the direction of the pan θ_{p_i}, the directions of all lineaments within it $\{\theta_{ij}\}_{j=1}^{n(i)}$, and the average direction of the lineaments $\bar{\theta}_i$.

2.4 Hypothesis Test for Significant Mean Differences

It is of interest to test if the mean of differences between the angles of various pans and the angles between a pan and associated geological lineaments differ significantly from zero. The single-sample hypothesis test for the mean direction of circular data, as discussed by Mardia and Jupp (2009), is applied. Since the population concentration parameter κ is unknown, the version of the test that accommodates this uncertainty is used, as the average direction of all pans is not available.

The hypothesis test focus is on the differences

$$d_i = \theta_{p_i} - \bar{\theta}_i, \tag{4}$$

for $i = 1, 2, \ldots, n$, where n is the total number of differences corresponding to n pans, and θ represents the angles of the pans and geological lineaments, respectively. The von Mises distribution is fitted to these angular differences d_i.

The von Mises distribution's density function involves two parameters: μ (mean direction) and κ (concentration). Special cases arise when $\kappa = 0$ (uniform distribution) or κ is large, approximating the normal distribution with mean μ and variance $\frac{1}{\kappa}$, and is defined by Mardia and Jupp (2009) as

$$f(d_i; \kappa, \mu) = \frac{e^{\kappa \cos(d_i - \mu)}}{2\pi I_0(\kappa)} \quad \text{with} \quad I_0(\kappa) = \frac{1}{2\pi} \int_0^{2\pi} e^{\kappa \cos(d_i - \mu)}, \tag{5}$$

where $I_0(\kappa)$ is the modified Bessel function of the first kind and order zero.

The parameters κ and μ are obtained by maximising the log-likelihood function using optimisation algorithms such as `optim` and `DEoptim` in R (Mullen et al., 2011).The mean direction μ, calculated as the circular mean of the angular differences d_i, serves as the initial value for the iterative optimisation.

When a mean direction μ_0 is specified, Equation 2 quantities are modified to (Mardia and Jupp, 2009)

$$\bar{C} = \frac{1}{n} \sum_{k=1}^{n} \cos(d_i - \mu_0) \quad \text{and} \quad \bar{S} = \frac{1}{n} \sum_{k=1}^{n} \sin(d_i - \mu_0).$$

The hypothesis statements follow as

$$H_0 : \mu_d = \mu_0 \quad \text{vs} \quad H_A : \mu_d \neq \mu_0,$$

where $\mu_0 = 0$ can be used to test for no significant difference. The likelihood ratio statistic for H_0 is

$$w = 2n(\hat{\kappa}\bar{R} - \tilde{\kappa}\bar{C} - \log_{I_0}(\hat{\kappa}) + \log_{I_0}(\tilde{\kappa})),$$

where $\tilde{k} = A^{-1}(\bar{C})$ denotes the ML estimate of κ under H_0 (Mardia and Jupp, 2009). Large values of w will result in rejecting the null hypothesis.

A simpler statistic is proposed as

$$w = 2n\hat{\kappa}(\bar{R} - \bar{C}), \tag{6}$$

where $\hat{\kappa}$ is the ML estimate of the concentration parameter. Under Wilks's theorem, and given a large sample size, w approximately follows a χ^2 distribution with 1 degree of freedom under the null hypothesis H_0 (Mardia and Jupp, 2009). Rejecting the null hypothesis suggests significant differences in angles, implying other factors may influence their orientation.

2.5 Clustering Analysis by Pan Size

The analysis can be further refined by examining how pan size influences the directional relationship between pans and nearby geological lineament(s). Classifying pan size is often subjective, which may lead to bias. To mitigate this, K-means clustering, an unsupervised classification method, is used to objectively group pans by size by minimizing the Within-Cluster Sum of Squares (WCSS). This ensures that observations within each cluster are as similar as possible while being as distinct as possible from those in other clusters (Sinaga and Yang, 2020).

Several methods can be used to identify the optimal number of clusters K e.g. the elbow method (Cui et al., 2020), silhouette analysis (Shutaywi and Kachouie, 2021), and gap statistics (Yang et al., 2019). The elbow method is examined in this paper due to its frequent use and practical relevance. The optimal K occurs at the point, where adding more clusters yields diminishing returns, showing a decrease in WCSS as K increases.

2.6 Testing Mean Direction Differences Across Clusters

The rose diagrams and circular means are recalculated for the respective clusters to compare their directional distributions and to assess whether the mean direction for each group differs significantly from zero. Assuming a sufficiently large sample size, the same approach as in Equation 6 is followed.

A new comparison takes the form of testing whether means are statistically different. Using the elbow method, if K clusters are identified as the optimal classification, the multiple-sample Watson-Williams test for the equality of mean directions is constructed as (Mardia and Jupp, 2009)

$$H_0 : \mu_{d_{C_1}} = \mu_{d_{C_2}} = ... = \mu_{d_{C_K}} \quad \text{vs} \quad H_A : \text{At least one mean pair is not equal}$$

where $\mu_{d_{C_i}}$ is the sample mean difference of the i^{th} group. This test is conducted using R packages such as `circular`. The three assumptions upheld while applying the test are: (1) the concentration parameters $\kappa_1, \kappa_2, ..., \kappa_K$ are assumed equal, (2) these concentration parameters must be sufficiently large ($\kappa > 1$), and (3) the observed angular differences are independent random samples, each following a von Mises distribution with parameters $\mu_{d_{C_i}}$ and κ_i where $i = 1, 2, ..., K$ (Mardia and Jupp, 2009). If these assumptions are violated, the test results may not be accurate. An alternative test is the non-parametric Watson-Wheeler test, which avoids these assumptions but requires at least 10 observations per cluster (Mardia and Jupp, 2009).

3. Application

A dataset containing 2735 pans or elliptical depressions was sourced from the LEAP-RE HyAfrica project using topographic maps from the South African Surveyor General. Geological lineaments were sourced from geophysical maps from the Council for Geoscience in South Africa. This data was used with approval from the University of Pretoria's Faculty of Natural and Agricultural Sciences Ethics Committee (NAS116/2019). By making use of fundamental geometric principles, the angles of the major axis of each pan were calculated and assigned to the pan under consideration. This information was used to calculate the relevant circular statistics and rose diagram.

As seen in Figure 3, the major axis of the pans is fairly spread out, with the mean direction pointing to NNE (North Northeast). The calculated circular mean is 1.4449 radians (or 82.79°), and the circular variance is 0.2239. These statistics highlight the dispersion and orientation of the pans'

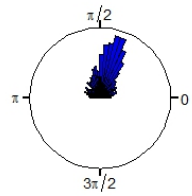
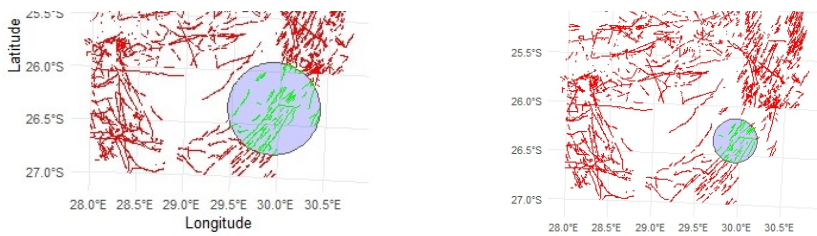


Figure 3. Rose diagram indicating the average direction of the major axis of all pans in the data set.



(a) (b)

Figure 4. (a) Buffer size of 48440 applied to a random pan (b) Example of a buffer with the highlighted lines used to calculate the average direction.

major axes. Unlike a traditional rose diagram that bins angles around the entire circle, the angles here are restricted between 0 and π, as explained in the methodology. The bins for the rose diagram have a width of 5° (0.09 radians) throughout the paper.

The average direction of the geological lineaments surrounding each pan is obtained using the algorithm, as outlined in Section 2.3. The buffer size required to meet the chosen criteria was 48440, shown in Figure 4a. Since the buffer size represents a one metre increase in the radius around the initial pan, a buffer of this size would mean that each buffer would cover around 0.96°, or roughly 96kms in diameter. A buffer of this magnitude would include a large proportion of the sample area and so adjustments were made to attempt to make this smaller, to restrict the geological lineaments to a more representative region around the pan of interest.

Further investigation revealed that some geological lineament data were missing, causing the buffer size to increase unnecessarily, as smaller pans required larger buffers to meet the criteria. After updating the data, the buffer size was nearly halved to 24855. Despite this reduction, the buffer size remained almost 50kms in diameter. To address this, an additional criterion was implemented: the buffers were adjusted so that 90% of the pans intersected with at least one geological lineament. This resulted in a smaller buffer size of 11016 (22kms diameter), while still ensuring the inclusion of most pans to minimise selection bias.

The pans with buffers that did not intersect were located in areas with few visible geological lineaments, likely due to geological factors that made them difficult to detect. This insight emphasizes the value of consultation with geological experts, and further investigation into this area will be considered in future work.

Figure 4b provides a clear representation of how the geological lineaments are selected for the calculation of average direction. In the calculation, each pan will get a single direction that represents

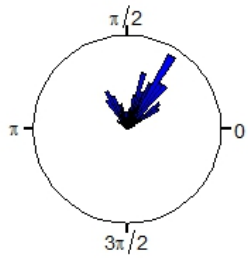


Figure 5. Rose diagram indicating the average direction of the geological lineaments per buffer.

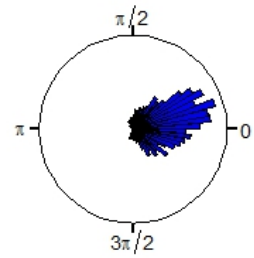


Figure 6. Rose diagram indicating the difference in the angles of pans and geological lineaments.

Table 1. Summary of circular statistics for the difference in directions.

Mean Direction	Circular Variance	Skewness Coefficient	Kurtosis Coefficient
0.1463 (8.38°)	0.3069	320.7871	7050.385

the average direction of the selected green geological lineament(s) within the buffer. The calculation of the angular difference follows from Equation 4 which entails subtracting this value from the calculated angle of the major axis of the pan. The rose diagram for the average direction of all the geological lineaments in the area is given in Figure 5.

The distribution of the average geological lineament directions is difficult to compare to that of the pans. The two seem to have a similar mean direction but this cannot be confirmed nor disregarded by means of the rose diagram in isolation. Based on the circular variance, the geological lineaments do display less variation.

The von Mises distribution was fitted to the angular differences d_i between the pans and geological lineaments. The ML estimate for the average angular difference is calculated as 0.1463 radians (8.38°), which serves as the initial value of μ in the optimisation procedure. The first hypothesis test aims to determine whether the mean angular difference is significantly different from zero.

Figure 6 and Table 1 display the rose diagram and circular statistics for the angular difference. In Figure 6, the angles are plotted on the right hemisphere, reflecting the nature of the angle difference calculation. If the average direction of the geological lineaments exceeds that of the pan, the result is a negative angle measured clockwise from zero. Angles near zero indicate similar orientations, while those further away suggest a larger discrepancy. However, a definitive conclusion cannot be drawn from the diagram alone.

The ML result, using both the `optim` and `DEoptim` functions in R, results in $\kappa = 1.9719$, directly applied in the test statistic displayed in Equation 6.

Table 2 presents the relevant test statistics for the hypothesis, conducted at the 5% significance level (the standard unless otherwise specified) using a buffer size of 11016. The null hypothesis is rejected, and it is concluded that the directional difference is significantly different from zero. This indicates that the average geological lineament direction and the major axis of the pans do not align. However, the small difference suggests some level of relationship. Test results may be influenced by

Table 2. Hypothesis test results: Testing mean direction deviation from zero.

TS	**CV**	p**-value**
71.9043	3.8415	2.2589×10^{-17}

Table 3. Comparison of circular statistics for three- and four-cluster solution.

Three-Cluster Solution	**Mean Direction**	**Circular Variance**	**Skewness**	**Kurtosis**
Small Cluster	1.4456	0.2251	-1935.704	18972.64
Medium Cluster	1.3824	0.1103	-71.7724	1352.909
Large Cluster	1.4504	0.0	N/A	N/A
Four-Cluster Solution	**Mean Direction**	**Circular Variance**	**Skewness**	**Kurtosis**
Small Cluster	1.4536	0.2319	-1687.6770	15825.82
Medium Cluster	1.3398	0.1186	-248.9124	8602.185
Large Cluster	1.4850	0.1164	-44.1764	490.5742
Very Large Cluster	1.4504	0.0	N/A	N/A

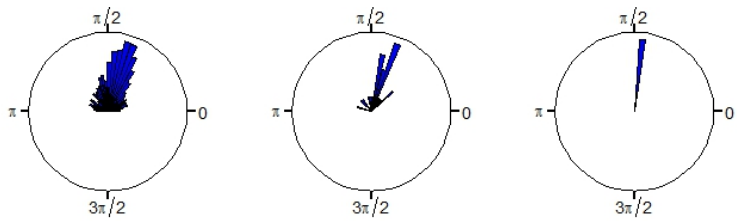
the selected buffer size, missing data, and the 90% criteria used to adjust the buffer size. Nonetheless, using a larger buffer size of 24855 yields a p-value of 1.5245×10^{-18}, which does not change the outcome of the test. Note that in the following tables, the test statistic is denoted as TS, and the critical value as CV.

Building on the finding that the directional difference is significantly different from zero, the potential effects of other factors, particularly pan size, should also be examined. Using WCSS, an elbow plot identifies three optimal clusters, grouping pans into small, medium, and large. Additionally, a four-cluster solution is explored for further insight, categorising pans as small, medium, large, and very large.

Sub-setting the data based on a three and four cluster solution is completed using the `kmeans` function in R. The data was split into 2708 small pans, 26 medium pans, and 1 large pan for the three-cluster solution. The four-cluster solution resulted in a respective split of 2536, 187, 11, and 1.

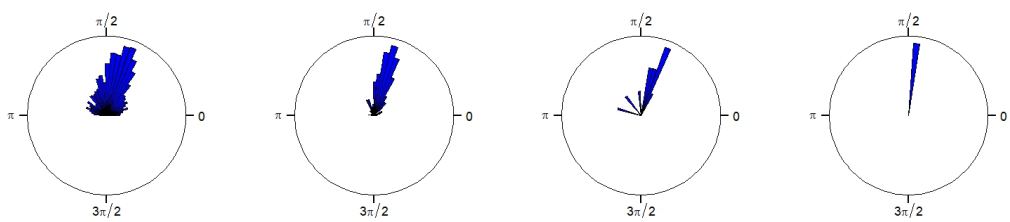
Table 3 presents the circular statistics for the three- and four-cluster solution, while the corresponding rose diagrams are shown in Figure 7 and Figure 8. Clusters with only one pan will have no circular variance, skewness coefficient, or kurtosis, and the mean direction represents the direction of the major axis of that pan. Similar to the previous analysis, buffers are fitted to the pans. However, in this case, each iterative procedure is applied to each cluster in isolation. Both the 90% criteria and the full data set criteria will be applied with the buffer sizes for the different cluster solutions provided in Table 4.

The direction of geological lineaments and the difference are calculated as before, within each cluster, and then combined for hypothesis testing. Once angles are calculated and the ML procedure completed, the test for cluster means differing from zero is conducted. The large (three- and four-cluster solutions) and very large cluster (four-cluster solution), consisting of only one and eleven pans respectively, cannot be tested due to insufficient small-sized clusters, leaving these untested. The fitted von Mises distributions and hypothesis test statistics for the cluster solutions are provided in Table 5, with results for both the standard and 90% criteria presented below one another.



(a) (b) (c)

Figure 7. Rose diagram indicating the average direction of the (a) Small pans, (b) Medium pans, and the individual (c) Large pan.



(a) Small Pans (b) Medium Pans (c) Large Pans (d) Very Large Pan

Figure 8. Rose diagram indicating the average direction of the (a) Small pans, (b) Medium pans, (c) Large pans, and the single (d) Very Large pan.

Table 4. Buffer size (in metres) for the full data and 90% criteria values for the cluster solutions.

Three-Cluster Solution	**Full Data**	**90% CV**
Small Cluster	24855	11028
Medium Cluster	8660	7892
Large Cluster	8660*	7892*
Four-Cluster Solution	**Full Data**	**90% CV**
Small Cluster	24855	11106
Medium Cluster	18110	8307
Large Cluster	8065	7666
Very Large Cluster	8065*	7666*

*The buffer size for these clusters was chosen to be the same as the previous cluster since there is already a geological lineament running through the pan. This ensures that this pan has more than one intersection.

Table 5. Test statistics and parameters: Cluster mean direction tested against zero.

Three-Cluster Solution	κ	μ	**TS**	**CV**	p**-value**
Small Cluster	2.2182	0.1307	74.8977	3.8415	4.96×10^{-18}
Medium Cluster	3.5256	0.1694	2.2112	3.8415	0.1370
Small Cluster (90% criteria)	1.9641	0.1451	69.5823	3.8415	7.33×10^{-17}
Medium Cluster (90% criteria)	3.2175	0.2883	5.2592	3.8415	0.0218
Four-Cluster Solution	κ	μ	**TS**	**CV**	p**-value**
Small Cluster	2.1621	0.1321	68.9956	3.8415	9.87×10^{-17}
Medium Cluster	2.1621	0.1412	6.5879	3.8415	0.0103
Small Cluster (90% criteria)	1.9170	0.1366	55.7723	3.8415	8.14×10^{-14}
Medium Cluster (90% criteria)	2.6172	0.2451	15.0475	3.8415	1.05×10^{-4}

Table 6. Watson-Wheeler results: Test for equality of mean direction across three cluster solution

Cluster Solution	**Test**	**Test Statistic**	**df**	p**-value**
Three-Cluster	Watson-Wheeler Test	4.1962	4	0.3801
Three-Cluster (90%)	Watson-Wheeler Test	4.6856	4	0.3211
Four-Cluster	Watson-Wheeler Test	30.43	6	3.256×10^{-5}
Four-Cluster (90%)	Watson-Wheeler Test	17.384	6	0.008

Based on Table 5, the results suggest that for all applicable clusters, the mean differences are significantly different from zero at any reasonable level of significance. The only exception is the medium cluster in the three-cluster solution. This result may be questionable due to the relatively small cluster size of 26, raising concerns about whether this sample size is sufficient for reliable conclusions and should be further investigated. Similarly, using the 90% criteria, the null hypothesis is rejected for all relevant tests, indicating that the mean directions of all tested clusters differ significantly from zero.

The final hypothesis test examines the homogeneity of means and angular distributions across the clusters. Originally, the Watson-Williams test was planned, but its assumption of equal concentration parameters (κ) across clusters are violated due to non-constant concentration parameters, leading to unequal dispersion of angular data. As a result, the Watson-Williams test is not applicable.

Therefore, the analysis switches to the Watson-Wheeler test, a non-parametric alternative that does not require equal concentration parameters. The tests were performed on both the original and the 90% criteria adjusted three- and four-cluster solutions. The results are summarised in Table 6.

The three cluster solution suggests no significant differences in angular distributions. However, the four cluster solution has a highly significant p-value of 3.256×10^{-5}, indicating substantial differences in angular distributions between the four clusters. In the 90% criteria solutions, the three-cluster suggests no significant differences, while the four-cluster solution still reveals significant differences in angular distributions even under the 90% criteria.

The significant p-values in the four cluster solution, for both the original and 90% criteria, strongly support the rejection of the null hypothesis, indicating heterogeneity in angular distributions across the four clusters. This is not the case for the three-cluster solutions.

While the Watson-Williams test could not be applied to assess homogeneity in mean directions due to the violation of its assumptions, the Watson-Wheeler test was successfully implemented and revealed significant differences in angular distributions, particularly for the four-cluster solution. This suggests that the average directional difference between geological features varies across different pan sizes, offering valuable insights into underlying patterns in the geological data.

It is important to note, though, that the Watson-Wheeler test assumes all clusters have more than 10 observations, a criterion that was not met in this analysis. As a result, while the test produced results, these should be interpreted with caution. Future work should aim to apply tests that meet all assumptions to confirm these findings and explore their practical implications in geological interpretation and modelling. **Note:** All relevant R code for the analysis can be found here.

4. Conclusion

This research aims to explore the directional relationship between elliptical natural hydrogen depressions and geological lineaments in Mpumalanga, South Africa, using circular statistics (circular mean, variance, skewness, and kurtosis), K-means clustering, hypothesis testing, and a novel algorithm for calculating the average direction of geological lineaments based only on segments within a defined bounding box. A dataset of 2735 pans was analysed, focusing on the angles of their major axes and their relation to surrounding geological features. Descriptive statistics revealed patterns in pan orientations, suggesting potential angular relationships with geological lineaments.

The unsupervised clustering, through K-means, yielded insights into the classification of pans based on area, identifying three- and four-cluster solutions with both significant and insignificant differences in angular distributions. Circular statistics were used to show that while larger clusters exhibited a more stable and consistent mean direction, smaller clusters showed considerable variability, reflecting the complex geological processes that may influence their formation and orientation, in addition to the effects of sample size. The accompanying rose diagrams represented the average orientation for each cluster, aiding in the understanding of the directional relationships in the data.

Hypothesis testing aimed to determine if the mean direction of each cluster significantly differed from zero. Most clusters showed significant angular differences, except for the medium-sized cluster in the three-cluster solution. Sample size issues prevented conclusions about the larger clusters, underscoring the need for caution in interpreting results.

The Watson-Williams test attempted to answer whether angular differences differed significantly across clusters. Here, the assumption of using populations with constant concentration parameters was violated, so the full hypothesis test could not be tested. Instead, the non-parametric Watson-Wheeler test was used, revealing significant heterogeneity in angular distributions, especially in the four-cluster solution. In contrast, the three-cluster solution showed no significant differences, indicating that the mean directions did not differ substantially across these clusters. These findings imply that geological processes influencing pan orientation vary by cluster size, though the assumption of at least 10 observations per cluster wasn't met, warranting caution and further investigation.

Future research should explore several key areas. One suggestion is to assign a weight to the average direction of the geological lineaments based on their lengths to provide a more accurate representation of the directional influence of the lineaments on the surrounding landscape. Similarly, the average direction of the major axes of the pans could be weighted according to their lengths. This

aims to investigate the potential significance of larger pans in shaping the spatial dynamics of the environment and provide deeper insights into the interplay between pan size and geological features.

Moreover, future studies should investigate potential spatial and geological dependence among pans such as the alignment, proximity, or shapes of surrounding pans.

This research has begun the process of laying important groundwork for understanding the intricate relationships between pan formations and geological lineaments, particularly in Mpumalanga. By continuing to employ interdisciplinary approaches and advanced statistical techniques, it opens the door for future investigations to further explore these natural systems.

Acknowledgements

This work was supported by the University of Pretoria Honors Merit Plus Bursary, Long-Term Joint European Union - African Union Research and Innovation Partnership on Renewable Energy LEAP-RE (https://www.leap-re.eu/hyafrica/), HYAFRICA project supported by the South African National Energy Development Institute (SANEDI), and the National Research Foundation (RA211213654339). A special acknowledgement to Paulo Mesquita from CONVERGE/Universidade de Évora for his assistance.

References

Bekker, A., Nakhaei Rad, N., Arashi, M., and Ley, C. (2022). Generalized skew-symmetric circular and toroidal distributions. *In Directional Statistics for Innovative Applications: A Bicentennial Tribute to Florence Nightingale*. Singapore: Springer, 161–186.

Carnicero, J. A., Ausin, M. C., and Wiper, M. P. (2013). Non-parametric copulas for circular–linear and circular–circular data: An application to wind directions. *Stochastic Environmental Research and Risk Assessment*, **27**, 1991–2002.

Cui, M. et al. (2020). Introduction to the k-means clustering algorithm based on the elbow method. *Accounting, Auditing and Finance*, **1** (1), 5–8.

Delbeke, J., Runge-Metzger, A., Slingenberg, Y., and Werksman, J. (2019). The Paris Agreement. *In Towards a Climate-Neutral Europe*. Routledge, 24–45.

Gielen, D., Boshell, F., Saygin, D., Bazilian, M. D., Wagner, N., and Gorini, R. (2019). The role of renewable energy in the global energy transformation. *Energy Strategy Reviews*, **24**, 38–50.

Ihaka, R. and Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, **5** (3), 299–314.

Lee, A. (2010). Circular data. *Wiley Interdisciplinary Reviews: Computational Statistics*, **2** (4), 477–486.

Mardia, K. V. and Jupp, P. E. (2009). *Directional Statistics*. John Wiley & Sons, West Sussex.

Michalak, M. P., Kuzak, R., Gladki, P., Kulawik, A., and Ge, Y. (2021). Constraining uncertainty of fault orientation using a combinatorial algorithm. *Computers & Geosciences*, **154**, 104777.

Moretti, I., Geymond, U., Pasquet, G., Aimar, L., and Rabaute, A. (2022). Natural hydrogen emanations in Namibia: Field acquisition and vegetation indexes from multispectral satellite image analysis. *International Journal of Hydrogen Energy*, **47** (84), 35588–35607.

Mullen, K., Ardia, D., Gil, D. L., Windover, D., and Cline, J. (2011). DEoptim: An R package for global optimization by differential evolution. *Journal of Statistical Software*, **40** (6), 1–26.

Pewsey, A., Neuhäuser, M., and Ruxton, G. D. (2013). *Circular Statistics in R*. Oxford.

Rivest, L.-P., Duchesne, T., Nicosia, A., and Fortin, D. (2016). A general angular regression model for the analysis of data on animal movement in ecology. *Journal of the Royal Statistical Society Series C: Applied Statistics*, **65** (3), 445–463.

Sanderson, D. J. and Peacock, D. C. (2020). Making rose diagrams fit-for-purpose. *Earth-Science Reviews*, **201**, 103055.

Shutaywi, M. and Kachouie, N. N. (2021). Silhouette analysis for performance evaluation in machine learning with applications to clustering. *Entropy*, **23** (6), 759.

Sinaga, K. P. and Yang, M.-S. (2020). Unsupervised K-means clustering algorithm. *IEEE Access*, **8**, 80716–80727.

Walsh, S. and Martill, D. (2006). A possible earthquake-triggered mega-boulder slide in a Chilean Mio-Pliocene marine sequence: Evidence for rapid uplift and bonebed genesis. *Journal of the Geological Society*, **163** (4), 697–705.

Wells, N. A. (2000). Are there better alternatives to standard rose diagrams? *Journal of Sedimentary Research*, **70** (1), 37–46.

Wonnacott, R. (1999). The implementation of the Hartebeesthoek94 co-ordinate system in South Africa. *Survey Review*, **35** (274), 243–250.

Xiang, Y., Sun, X., Liu, D., Yan, L., Wang, B., and Gao, X. (2020). Spatial distribution of Rn, CO_2, Hg, and H_2 concentrations in soil gas across a thrust fault in Xinjiang, China. *Frontiers in Earth Science*, **8**.

Yang, J., Lee, J.-Y., Choi, M., and Joo, Y. (2019). A new approach to determine the optimal number of clusters based on the gap statistic. *In International Conference on Machine Learning for Networking*. Springer, 227–239.

Submitted to the Proceedings of the 65th Annual Conference of the South African Statistical Association

Covariate-based distance outlier scoring for nonconvex domain estimation

Kabelo Mahloromela and Inger Fabris-Rotelli

Department of Statistics, University of Pretoria, Pretoria, South Africa

To conduct an appropriate analysis of point pattern data, knowledge of the location of events and the spatial domain, termed a window, in which these locations are observed is required. The window defines the extent for analysis and is directly involved in estimating and inferring on the first-and-second order properties of the point process that generated the data. Often, the window is known and recorded during data collection. Otherwise, it must be chosen objectively by the researcher. When the window is unknown, the typical approach is to use the minimum bounding box or the convex hull of the observed locations. Choosing too large a window, however, may lead to spurious estimation and inference in regions where points cannot occur. In a setting where points are restricted by physical or other phenomena, a nonconvex window that accounts for these constraints provides a more representative domain for analysis. Herein, we propose an algorithm for estimating nonconvex windows for point pattern data using covariate-based distance outlier scores and Otsu thresholding. The robustness of the algorithm is evaluated on various point pattern types, namely regular, clustered, inhomogeneous, and completely spatially random point patterns. An application to rural village household locations in Tanzania is then considered.

Keywords: Covariate, Nonconvex, Point pattern, Window domain.

1. Introduction

Spatial point pattern data consists of a set of locations of events (Cressie, 1993; Baddeley et al., 2015). The locations are observed within a spatial region termed a window. Conventionally, the point pattern is assumed to arise from a point process, a stochastic mechanism, whose characteristics are of scientific interest (Cressie, 1993; Baddeley et al., 2015).

The window is often given a priori. In this setting, the window boundaries are known and can be recorded during data collection. For example, in epidemiological applications, disease incidence locations may form points in a point pattern with municipal boundaries delineating the window's perimeter (Gatrell et al., 1996; Reader, 2000). In other cases, the window boundaries may not be clearly defined and must be chosen objectively by the researcher, e.g., the boundary of nesting territories in an ecological study (Newton et al., 1977; Dare and Barry, 1990). Herein, we consider the latter.

Corresponding author: Kabelo Mahloromela (kabelo.mahloromela@up.ac.za)
MSC2020 subject classifications: 62H11, 62P99

Selecting the window must be done carefully as it gives information about where observations were and were not made and where they could be predicted (Baddeley et al., 2015). A large window choice may result in spurious estimation and incorrect conclusions about the properties of the underlying point process since the description of these properties rely implicitly or explicitly on the specification of the window, e.g., when computing density measures based on the area of the window, when correcting for edge effects (Baddeley et al., 2015; Diggle, 1985; Baddeley et al., 2022), and when using distance to quantify spatial dependence and correlation (Diggle et al., 1976; Chiu and Stoyan, 1998).

Efforts to infer the window based on observed data locations have been made. In Ripley and Rasson (1977), a technique for window selection is proposed to reconstruct an unknown compact convex set from a realisation of a homogeneous planar Poisson process observed from this unknown set. The solution therein is the dilation of the convex hull about its centroid. Moore (1984) presents a method for addressing a similar problem, i.e., estimate a compact convex set, given independent observations sampled uniformly from the unknown set. Rasson et al. (1996), Rasson et al. (1994), and Remon (1994) extend these methods to estimate convex sets with observations that are inside and outside the convex set. Other literature that address determining a convex set from a random set of points include (and are not limited to) Efron (1965) and Dattorro (2010).

While standard methods for window selection typically consider rectangular or convex domains, real-world applications may involve nonconvex windows that emerge due to phenomenon that constrain the occurrence of points (Baddeley et al., 2012; Myllymäki et al., 2020). For example, features such as mountains, valleys, rivers, or deserts may be obstacles that create discontinuities or boundaries in space that limit where points can move to or be placed. In Mahloromela et al. (2023), a method was developed to construct nonconvex windows by using covariate information in an algorithm that is based on moving window statistics. The algorithm, however, was only tested on real cases, and its suitability for different point patterns was not investigated.

In this paper, a new algorithm for nonconvex window selection is provided that makes use of a weighted repeated sample nearest neighbour distance outlier score and Otsu's method for automatic threshold selection. We work in a setting where points in a point pattern are constrained by physical or other phenomenon, represented as spatial covariates. The algorithm's robustness is evaluated on point patterns with various first- and second-order characteristics to assess its behaviour, reliability, and generalisability across different point pattern types. The algorithm is then applied to select the appropriate window of village household locations in rural Tanzania.

The remainder of this paper is organised as follows. Section 2 provides some definitions and a presentation of the proposed nonconvex window selection algorithm. The performance of the algorithm is evaluated through a simulation study in Section 3. An application to rural village household locations in Tanzania is given in Section 4, a discussion in Section 5, and concluding remarks in Section 6.

2. Methodology

In this section a formal definition of a point pattern data set is provided in Section 2.1. The proposed nonconvex window selection algorithm uses a repeated sample nearest neighbour distance outlier score and Otsu's method for image thresholding; thus, a discussion of these topics is provided in

Sections 2.2 and 2.3, respectively. The nonconvex window selection algorithm based on covariate data is outlined in Section 2.4.

2.1 Point patterns

A point pattern dataset typically comprises a set of locations, $\{x_1, ..., x_n\}$, observed in a spatial domain $W \subset \mathbb{R}^2$, where $n \geq 0$ is not predetermined (Baddeley et al., 2015; Cressie, 1992; Illian et al., 2008). Additional covariate information, as a spatial measurement $z(u)$ with $u \in W$, may also be recorded. In practice, the points u usually form a regular lattice extracted from continuous data over W and do not necessarily coincide with the data points. When covariate information is available, the dependence of a point pattern on the covariate should be investigated and quantified. For a thorough discussion of these techniques, see Baddeley et al. (2015) and Myllymäki et al. (2020).

2.2 Nearest neighbour distance outlier score

Covariates at observed locations represent a random sample[1] of possible data values of a point pattern. When points are constrained by covariates, the distribution of covariate values at observed locations will differ significantly from those at random locations generated by a mechanism independent of the covariate. A value that deviates significantly from the set of covariates at observed locations may thus provide an indication that it arose from a mechanism that is different to that of the possible data values. Owing to this, the use of a distance-based outlier detection framework is proposed to classify covariate values into two sets: inliers, which coincide with the possible data locations, and outliers, locations where points cannot occur. In particular, the outlier-score introduced in Pang et al. (2015) is used. The score is defined as the distance between a given point, u, and its nearest neighbour in repeated independent random samples of the data, i.e.,

$$q(u) := \frac{1}{r} \sum_{j=1}^{r} \min_{s \in S_j(x)} d(u, s),$$

where $S_j(x)$ is the j-th random sample (selected with replacement) from the data set $x = \{x_i\}_{i=1}^{n}$, $d(u, s)$ is the Euclidean distance between u and s, and r is the number of random subsets selected. This technique combines multiple nearest-neighbour outlier scores from data re-samples. Data values that obtain relatively large scores are considered outliers. In order to classify a data point as an inlier or outlier based on its outlier score, a threshold value must be chosen. For this task, we use Otsu thresholding.

2.3 Otsu thresholding

Otsu thresholding (Otsu et al., 1975) is a technique that is widely used for automatic greyscale image segmentation. The image pixels are divided into two groups, namely foreground and background, by a threshold τ, i.e., a set of pixel values less than or equal to τ and a set of pixel values greater than τ (relabelled as 0 and 1, respectively).

Consider an image represented as an $M \times K$ array of pixels with greyscale values $\{y_i\}_{i=1}^{L}$, where L is the total number of pixels. Let $I(\cdot)$ be an indicator function that is 1 when a condition is true

[1] Albeit in some cases a biased sample, e.g., for clustered point patterns.

and 0 otherwise. Otsu's thresholding method exhaustively searches for a threshold that maximises the inter-class variance,

$$\sigma^2(\tau) = \omega_0(\tau)\omega_1(\tau)(\mu_0(\tau) - \mu_1(\tau))^2,$$

where

$$\omega_0(\tau) = \frac{1}{L}\sum_{i=1}^{L} I(y_i \leq \tau)$$

is the fraction of background pixels,

$$\omega_1(\tau) = 1 - \omega_0(\tau) = \frac{1}{L}\sum_{i=1}^{L} I(y_i > \tau)$$

is the fraction of foreground pixels,

$$\mu_0(\tau) = \sum_{i=1}^{L} \frac{y_i I(y_i \leq \tau)}{L\omega_0(\tau)}$$

is the average of the background pixel values, and

$$\mu_1(\tau) = \sum_{i=1}^{L} \frac{y_i I(y_i > \tau)}{L\omega_1(\tau)}$$

is the average of the foreground pixel values.

Otsu thresholding aims to find the pixel value that best separates an image into foreground and background pixels. The inter-class variance is the criterion used to measure separability between these group of pixels, with larger values indicating better separation. The process tests each pixel value as a threshold, grouping pixels into foreground or background based on whether they are below or above this threshold. The proportion of background and foreground pixels is determined along with the average of the pixel values for each group. These are then used to compute the inter-class variance. The pixel value that yields the highest inter-class variance is chosen as the optimal threshold. In our proposed setting, Otsu thresholding is applied to automatically select a threshold value that separates the outlier scores into those associated with inlier and outlier pixels, where the pixel values are the outlier scores determined based on the covariates at observed locations.

2.4 Window selection algorithm

In this section, we present a new algorithm for nonconvex window selection. The algorithm constructs nonconvex spatial domains by making use of covariate data. The algorithm differs in approach from that of Mahloromela et al. (2023) and better accounts for the bias in sampled covariates that is induced by heterogeneity that arises in certain point pattern types, which was not considered in Mahloromela et al. (2023). It is important to investigate the dependence of the point pattern on the covariate before implementation of the algorithm. Ideally, covariates believed to impact the distribution and abundance of points, or that are correlated to them should be used (see Baddeley et al. (2015) and Myllymäki et al. (2020)).

Let $x = \{x_1, ..., x_n\}$ denote the set of observed locations in the point pattern. The points are assumed to be observed from an unknown domain $W \subset \mathbb{R}^2$. Let $z(u)$ be the value of a covariate

at location $u \in W^*$, where $W^* \supset W$. The covariate is represented as an image with m pixel cells, denoted by P_j, $j = 1, ..., m$, each holding the value of the covariate, i.e., $z(u_j)$ is the value of the covariate associated with the j-th pixel, where u_j is the location of the pixel center. Without loss of generality we suppose that W^* is the smallest bounding rectangular window that contains all the observed data points.

The algorithm uses the notion that covariate values at observed points are a random subset of potential data values within the point pattern, although in some instances, the sample may be biased. When the placement of points is influenced by covariates, the covariate distribution at observed points will deviate substantially from that at randomly chosen points generated by a process not dependent on the covariate. Thus, to determine whether a given value of a covariate is "anomalous" relative to the covariate values at observed locations, the repeated sampled nearest neighbour distance outlier score is used, given by

$$q(u) = \frac{1}{r} \sum_{k=1}^{r} \min_{s \in S_k(x)} d(z(u), z(s)),$$

where $S_k(x)$ is the k-th weighted random sample of the data set $x = \{x_i\}_{i=1}^n$, $d(z(u), z(s))$ is the absolute difference between the covariate values $z(u)$ and $z(s)$, and r is the number of random subsets. To account for the potential selection bias of the covariate values that is induced by point patterns with high clustering and inhomogeneous intensity, $S_k(x)$ is chosen as a weighted random sample (with replacement) where the weight w_i at a location x_i is the multiplicative inverse of the kernel density estimate of the locations $\{x_i\}_{i=1}^n$,

$$w_i = \left(n^{-1} \sum_{v=1}^{n} K_h(x_i - x_v) \right)^{-1},$$

where $K_h(x) = \frac{1}{h} K(\frac{x}{h})$ is a scaled kernel weighting function, $K(\cdot)$ is a probability density on $\mathbb{R}^2$, and $h > 0$ is a smoothing bandwidth.

When the value of the covariates at a given location differs substantially from those at at observed locations, the outlier score will be significantly large. To increase the contrast between inlier and outlier scores, we apply a log transformation to the values of the scores and then use Otsu's method to classify the pixels associated with the values of the transformed scores into an inlier and outlier group. The final window constructed is the union of all pixel cells associated with covariates that are part of the inlier group that is produced via Otsu's method. The threshold, τ, chosen maximises

$$\sigma_q^2(\tau) = \alpha_0(\tau)\alpha_1(\tau)(\overline{q}_0(\tau) - \overline{q}_1(\tau))^2,$$

where

$$\alpha_0(\tau) = \frac{1}{m} \sum_{j=1}^{m} I(-\ln(q(u_j)) \leq \tau)$$

is the fraction of transformed outlier scores less than or equal to τ,

$$\alpha_1(\tau) = 1 - \alpha_0(\tau) = \frac{1}{m} \sum_{j=1}^{m} I(-\ln(q(u_j) > \tau)$$

is the fraction of transformed outlier scores greater than τ,

$$\overline{q}_0(\tau) = \sum_{j=1}^{m} \frac{-\ln(q(u_j))I(-\ln(q(u_j) \leq \tau)}{m\alpha_0(\tau)}$$

is the average of transformed outlier scores less than or equal to τ, and

$$\overline{q}_1(\tau) = \sum_{j=1}^{m} \frac{-\ln(q(u_j))I(-\ln(q(u_j)) > \tau)}{m\alpha_1(\tau)}$$

is the average of transformed outlier scores greater than τ. The resultant window is given by

$$\widehat{W} = \bigcup_{\forall j \ni \ln(q(u_j)^{-1}) \leq \tau} P_j.$$

Opening and closing (mathematical morphological operations) can be applied to the constructed window as a post-processing step to remove noise and improve the window smoothness (Najman and Talbot, 2013). Next, we perform a simulation study to evaluate the robustness of the algorithm to different point patterns.

3. Simulation study

In this section, we perform a simulation study to test the robustness of the window selection algorithm to point patterns with different first- and second-order properties. All computations are performed using the R programming language (R Core Team, 2021). Simulations of point patterns and spatial covariates are done using functions in the `spatstat` (Baddeley et al., 2015) and `SpatialExtremes` (Ribatet, 2022) packages, respectively.

3.1 Simulation outline

The performance of the proposed methodology is evaluated with simulated data. The basic outline of the simulation is as follows. A spatial covariate is simulated using a Gaussian random field (Stein, 2012). A threshold for the covariate values is chosen to delineate the simulation domain for point pattern locations. The window is defined as the set of locations that coincide with covariate values based on the following three cases: points may only occur at locations with a covariate value that is less than the chosen threshold; points may only occur at locations with a covariate value that is greater than the chosen threshold; and points may only occur a locations with covariate values that lie between two chosen threshold limits. Point patterns with different first- and second-order characteristics, namely point patterns that are completely spatially random, regular, and clustered are then simulated on the defined domain. The window construction algorithm is then applied and the quality of the estimated window is evaluated using statistics derived from the area of the set difference between the simulation extent and the estimated window.

3.2 Simulating the covariate

A Gaussian random field with a Whittle-Matern covariance model (Stein, 2012; Whittle, 1954) is used to simulate the spatial covariate in a window $W^* = [0, 10] \times [0, 10]$. Two different values, 0.5

and 1.5, are used as range parameters in the Whittle-Matern covariance model to simulate spatial covariates with short and moderate autocorrelation structures, respectively. For ease of reference, we label the covariates and refer to these labels in the rest of this document. Two spatial covariates with a range parameter of 0.5 are generated and labelled Covariate I and Covariate II and two spatial covariates with a range parameter of 1.5 are generated and labelled Covariate III and Covariate IV.

3.3 Using the covariate to define the window

To define the simulation window, W, on which the points are generated, we use thresholds based on the covariate values to specify the locations at which points should be simulated. The following three cases are considered. In the first case, points only occur at locations that coincide with covariate values that are less than a chosen threshold. Figure 1(a) shows the result of applying this case to Covariate I. In the next case, points only occur at locations that coincide with covariate values that are greater than a chosen threshold. Figures 1(b) and (c) depict the result of applying this case to Covariate II and III, respectively. In the final case, points only occur at locations that coincide with covariate values that lie between two chosen limits. Figure 1(d) illustrates the result of applying this case to Covariate IV.

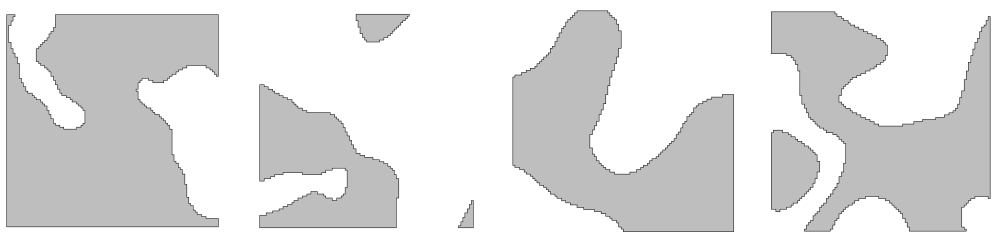


Figure 1. Simulation extent of points based on selecting thresholds for (a) Covariate I, (b) Covariate II, (c) Covariate III, and (d) Covariate IV. The grey regions indicate where points can be simulated, while the white regions represent areas where points cannot occur.

3.4 Simulating the point pattern

For the point pattern simulation, we consider four different spatial point process models, namely a homogeneous Poisson point process, heterogeneous Poisson point process, a simple sequential inhibition process, and a Matérn cluster process. The simulation of point patterns from these models are performed using the functions `rpoispp`, `rSSI`, and `rMatClust` in the `spatstat` package in R. For each window, we simulate 1000 point patterns from the different models and consider small (S), medium (M), and large (L) sized point patterns with 100, 500, and 1000 points, respectively.

3.5 Evaluation metrics

To assess the algorithms performances, statistics based on the area of the set difference between the simulation window, W, and the constructed window, $\widehat{W}$, are computed for each simulated point

pattern[2].

3.6 Results

The results of the simulation are presented in Figures 2–4.

In Figure 2, a high mean accuracy for the window selection algorithm can be observed, with values ranging from 91% to 99% across various point patterns, sizes, and covariates. Accuracy increases as the size of the point patterns increases from small to large across all point pattern types and covariates. Small clustered point patterns show the poorest performance overall. The algorithm is relatively stable, as demonstrated by low standard deviation values ranging from 0.01% and 4.41% for the different point patterns and covariates.

The result in Figure 3 show that, on average, the algorithm has high F_1-score values that are fairly close to 1. This suggests that the algorithm does well in identifying regions to retain in the window while also being accurate in doing so. The pattern of improved performance of the algorithm as the size of point patterns increases is also notable here.

Figure 4 indicates a high degree of overlap between the predicted and true window, implying that the algorithm typically performs well in localising the extent of the window domain. Poor performance in the case of small point patterns is observed especially in the case of clustered point patterns.

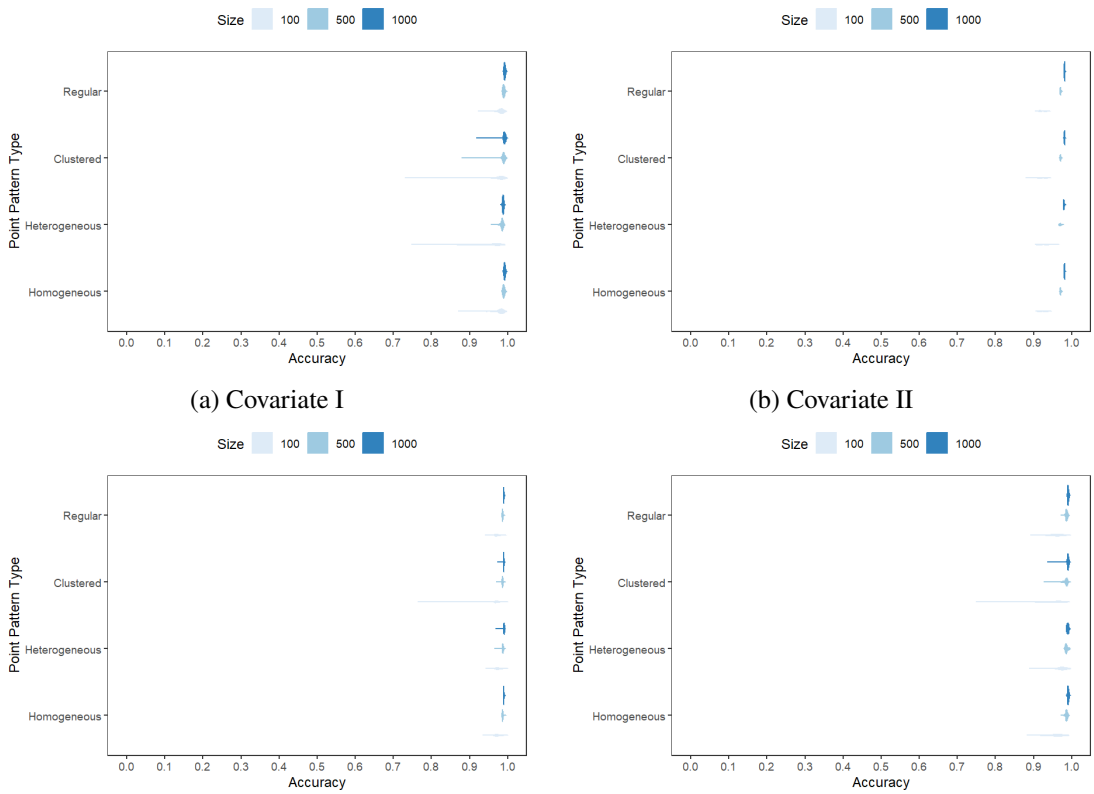
The window selection algorithm shows relatively good performance across all evaluation metrics. The algorithm has strong capabilities in retaining relevant regions, with improvements observed as point pattern sizes increase. The Intersection over Union (IoU) shows large overlap between the estimated and true simulation extent, demonstrating the algorithm's ability to appropriately delineate the boundary of the window domain. Although performance is lower for small clustered point patterns, the algorithm performs relatively well in most of the considered scenarios.

4. Application

We now apply the proposed methodology to estimate the spatial domain of rural household locations in Tanzania. The data used was collected in a census in the Serengeti District, Mara province, Northern Tanzania[3]. The data comprises georeferenced locations for 35 947 households spread across 88 villages. The locations are given in latitude and longitude decimal degree coordinates. Three villages are considered for this paper: namely Iseresere, Kono and Nyamakobiti with household locations numbering 295, 320 and 412, respectively. Spatial covariate data of terrain elevation for Tanzania is extracted from a Digital Elevation Model (DEM): a raster grid with each cell containing a value of the elevation (in meters) of the earth's surface above sea level. The data was collected in the Shuttle Radar Topographic Mission (SRTM). The SRTM data were sampled over a grid of 1 arc-second by 1 arc-second (approximately 30m by 30m). The top pane of Figure 5 shows the terrain slope of the elevation data from the DEM.

[2] True Positive (TP) = $|W \cap \widehat{W}|$, True Negative (TN) = $|W' \cap \widehat{W}'|$, False Positive (FP) = $|W' \cap \widehat{W}|$, False Negative (FN) = $|W \cap \widehat{W}'|$, where the symbol $'$ denotes the complement. These values are then used to compute the accuracy, F_1-score, and the intersection over union score.

[3] Provided by Katie Hampson, http://www.gla.ac.uk/researchinstitutes/bahcm/staff/katiehampson, http://www.katiehampson.com/#intro, and approved for use by the Faculty of Natural and Agricultural Science Research Ethics committee under the reference NAS33/2019.



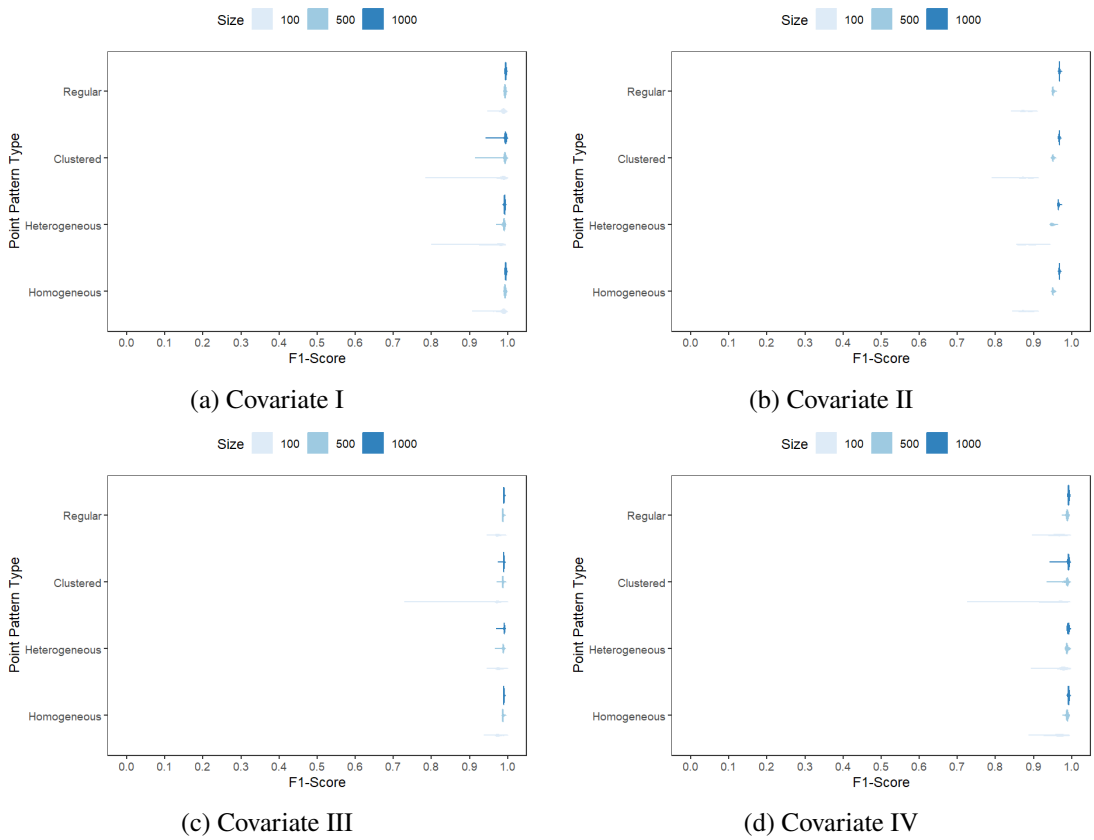
(a) Covariate I

(b) Covariate II

(c) Covariate III

(d) Covariate IV

Figure 2. Violin plots of the accuracy of the window selection algorithm for point patterns of different types and sizes across Covariates I–IV.



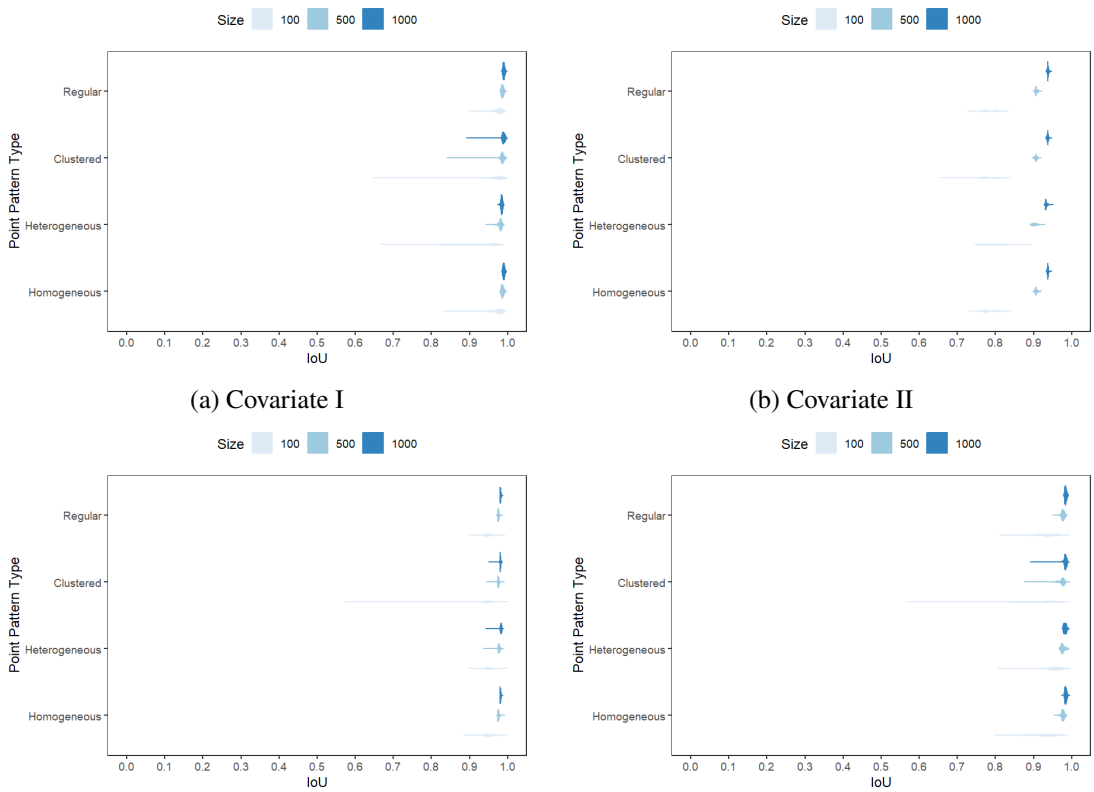
(a) Covariate I

(b) Covariate II

(c) Covariate III

(d) Covariate IV

Figure 3. Violin plots of the F_1-score of the window selection algorithm for point patterns of different types and sizes across Covariates I–IV.



(a) Covariate I

(b) Covariate II

(c) Covariate III

(d) Covariate IV

Figure 4. Violin plots of the intersection over union of the window selection algorithm for point patterns of different types and sizes across Covariates I–IV.

Human settlements in rural areas often depend on some attractive and limiting features of the natural landscape. Topographic properties of terrain influence the distribution of environmental phenomena and the nature of environmental processes. In rural areas, steep slopes present many challenges which make it impracticable for building houses. Steep slopes have greater requirements in terms of structural planning and costs. Flat land is easier and cheaper to build on since less time and expenses are incurred in getting the land suitable for building. Owing to these reasons, terrain slope is useful for describing terrain viable for (new) house locations and is used here as a covariate to characterise land that is suitable for building.

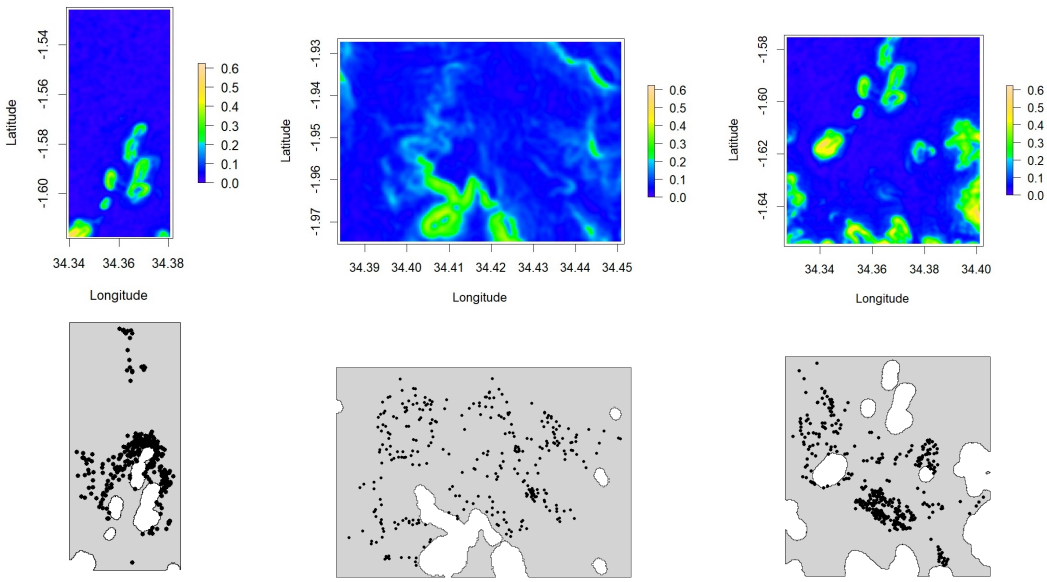
The algorithm is applied to each of the villages and the results are presented in the bottom pane of Figure 5. Opening and closing (mathematical morphological operations) are applied to the resultant windows to remove noise and smooth the window edges (Najman and Talbot, 2013). For each village, the algorithm detects and removes regions with high terrain slope. Regions where terrain slope values are close to covariate values at observed household locations have been identified by the algorithm as low terrain slope areas and are retained in the final window. Even though there are no realisations of the point process in certain areas, the algorithm accounts for the possibility of a point occurring there as long as it satisfies the definition of viable land that is a function of covariate values at observed locations in the pattern.

Terrain features such as hills (areas of high ground), ridges (sloping line of high ground) and flat plains (even landmass of relatively uniform elevation) are identifiable in the figures. We observe that households are spread along the edge of terrain with high relief. The occurrence of households is only seen on flat plain areas and at the base of the mountainous regions. The households cluster on plains adjacent to scarps (i.e. steep slopes). The plan of the village is mostly adjusted to the relief features of the region, some along the edges of the hill slopes. The algorithm shows a strong ability to identify terrain that are suitable for household locations to occur on. Thus the algorithm has admitted a data-driven domain selection approach that aligns relatively well with the underlying processes generating the point pattern.

5. Discussion

In this paper, a spatial domain estimation technique for point pattern data is considered. For an appropriate analysis of point pattern data, the window must be defined since estimation and prediction rely on it. When the window is unknown, the typical approach, when inferring on it, is to assume that it is convex and that the point process that generated the data is a homogeneous Poisson process, i.e., assumptions which may not be true in practice. In real world applications, the distribution of points may be constrained by some underlying process, expressed as a covariate, resulting in more complex spatial windows. When covariate information is available, the dependence of the point pattern on the covariate should be investigated. Parametric models that incorporate this dependence and formal hypothesis testing procedures, under parametric assumptions, are well developed. Nonparametric methods have received some attention including extending the kernel smoothed intensity estimate to allow for covariate effects.

An algorithm is proposed to estimate the domain of a point pattern dataset without the assumption of convexity or a point process model. The robustness of the algorithm to different point pattern types is investigated. The algorithm performs well for most cases considered in the simulation experiment,



(a) Iseresere (b) Kono (c) Nyamakobiti

Figure 5. Constructed window (bottom) and terrain slope (top) for villages in Tanzania's Mara province.

with the exception of small clustered point patterns. An application to rural household locations is considered with the use of elevation data from a DEM. In other cases, it may be appropriate to include other covariates that characterise a feature of the landscape that is unsuitable for building new houses, such as areas with rivers, dams or marshes.

6. Conclusion

Window selection for spatial data is a complex process, most often requiring expert knowledge if not obtained using a data-driven approach, such as herein. The common generic approaches used are the smallest rectangular bounding window and convex windows. A chosen window must however cover the true domain of the sampled spatial data in order to facilitate modelling. Here, we presented a new algorithm for estimating the spatial point pattern domain without the restriction of a convexity assumption. The algorithm works by using a reweighted repeatedly sampled nearest neighbour distance outlier score and Otsu's method for automatic threshold selection on covariates at observed point locations. Using a feature of the spatial covariate in regions at observed points in the pattern, the proposed method constructs a nonconvex window. The robustness of the algorithm to various point pattern types and covariates was tested in a simulation study. The algorithm performs well in most settings with the poorest performance observed for small clustered point patterns. An application to rural village households in Tanzania's Mara province was considered. Remotely sensed data from a DEM was used as a covariate in this case. The algorithm performed well in detecting and filtering areas of high relief and steep slopes, which were observed characteristics that suggested the low likelihood of household occurrence in these regions. When the movement between points

is constrained to such a nonconvex window, the Euclidean distance will not give a representative distance measure of the path between points. Consequently, the Euclidean shortest path distance calculated on the nonconvex window is a measure better suited to quantifying the proximity between points and should therefore be used in any further spatial analysis. In future work, the algorithm could be extended to allow for an ensemble of spatial covariate effects. One could also investigate other automatic threshold selection techniques.

Acknowledgements

This research received support from the National Research Foundation of South Africa (Grant Number 137785). The content and opinions expressed herein are the sole responsibility of the authors and do not necessarily represent the official views of the NRF.

References

Baddeley, A., Chang, Y.-M., Song, Y., and Turner, R. (2012). Nonparametric estimation of the dependence of a spatial point process on spatial covariates. *Statistics and its Interface*, **5** (2), 221–236.

Baddeley, A., Davies, T. M., Rakshit, S., Nair, G., and McSwiggan, G. (2022). Diffusion smoothing for spatial point patterns. *Statistical Science*, **37** (1), 123–142.

Baddeley, A., Rubak, E., and Turner, R. (2015). *Spatial Point Patterns: Methodology and Applications with R*. CRC Press.

Chiu, S. and Stoyan, D. (1998). Estimators of distance distributions for spatial patterns. *Statistica Neerlandica*, **52** (2), 239–246.

Cressie, N. (1992). Statistics for spatial data. *Terra Nova*, **4** (5), 613–617.

Cressie, N. (1993). *Statistics for Spatial Data*. John Wiley & Sons.

Dare, P. and Barry, J. (1990). Population size, density and regularity in nest spacing of Buzzards Buteo Buteo in two upland regions of North Wales. *Bird Study*, **37** (1), 23–29.

Dattorro, J. (2010). *Convex Optimization & Euclidean Distance Geometry*. Lulu.

Diggle, P. (1985). A kernel method for smoothing point process data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **34** (2), 138–147.

Diggle, P. J., Besag, J., and Gleaves, J. T. (1976). Statistical analysis of spatial point patterns by means of distance methods. *Biometrics*, 659–667.

Efron, B. (1965). The convex hull of a random set of points. *Biometrika*, **52** (3-4), 331–343.

Gatrell, A. C., Bailey, T. C., Diggle, P. J., and Rowlingson, B. S. (1996). Spatial point pattern analysis and its application in geographical epidemiology. *Transactions of the Institute of British Geographers*, 256–274.

Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008). *Statistical Analysis and Modelling of Spatial Point Patterns*, volume 70. John Wiley & Sons.

Mahloromela, K., Fabris-Rotelli, I. N., and Kraamwinkel, C. (2023). Covariate construction of nonconvex windows for spatial point patterns. *South African Statistical Journal*, **57** (2), 65–87.

Moore, M. (1984). On the estimation of a convex set. *The Annals of Statistics*, **12**, 1090–1099.

Myllymäki, M., Kuronen, M., and Mrkvička, T. (2020). Testing global and local dependence of point patterns on covariates in parametric models. *Spatial Statistics*, **42**, 100436.

Najman, L. and Talbot, H. (2013). *Mathematical Morphology: From Theory to Applications*. John Wiley & Sons.

Newton, I., Marquiss, M., Weir, D., and Moss, D. (1977). Spacing of Sparrowhawk nesting territories. *The Journal of Animal Ecology*, 425–441.

Otsu, N. et al. (1975). A threshold selection method from gray-level histograms. *Automatica*, **11** (285-296), 23–27.

Pang, G., Ting, K. M., and Albrecht, D. (2015). Lesinn: Detecting anomalies by identifying least similar nearest neighbours. *In 2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 623–630.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Rasson, J., Rémon, M., and Henry, F. (1996). Finding the edge of a Poisson forest with inside and outside observations: The discriminant analysis point of view. *In From Data to Knowledge*. Springer, 94–101.

Rasson, J.-P., Remon, M., Kubushishi, T., and Henry, F. (1994). Finding the edge of a Poisson forest with inside and outside observations: a theoretical point of view. *In Internal Report 94/22*. Department of Mathematics, FUNDP Namur.

Reader, S. (2000). Using survival analysis to study spatial point patterns in geographical epidemiology. *Social Science & Medicine*, **50** (7-8), 985–1000.

Remon, M. (1994). The estimation of a convex domain when inside and outside observations are available. *Supplemento ai Rendiconti del Circolo Matematico di Palermo*, **35**, 227–235.

Ribatet, M. (2022). *SpatialExtremes: Modelling Spatial Extremes*. R package version 2.1-0.
URL: *https://CRAN.R-project.org/package=SpatialExtremes*

Ripley, B. and Rasson, J.-P. (1977). Finding the edge of a Poisson forest. *Journal of Applied Probability*, **14** (3), 483–491.

Stein, M. L. (2012). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media.

Whittle, P. (1954). On stationary processes in the plane. *Biometrika*, **41**, 434–449.

Submitted to the Proceedings of the 65th Annual Conference of the South African Statistical Association

Investigation on the robustness of clustered point pattern simulation

Amy E. Pieters, René Stander, Kabelo Mahloromela, Renate Thiede and Inger Fabris-Rotelli

Department of Statistics, University of Pretoria, Pretoria, South Africa

Spatial point pattern analysis considers the arrangement of spatial locations and whether there is an underlying pattern. In this research, we consider clustered point patterns, spatial point patterns where the points attract each other. Investigating clustered point patterns can highlight problems in the simulation and fitting of such point patterns. Various cluster point pattern types are simulated and the robustness of the simulation is examined by using functions available in the `spatstat` package in R. The point patterns are simulated, then a point process model is fitted to the data and the fitted parameters are used to simulate a new point pattern. The simulated and resimulated point patterns are compared using the K-function and Kolmogorov-Smirnov tests. We conclude with a proposed methodology to use when simulating or fitting clustered point pattern data. The results only consider the Matern point pattern as it is the most accessible and widely used point pattern.

Keywords: Clustered point pattern, K-function, Point pattern simulation.

1. Introduction

Spatial statistics makes inferences about the spatial nature of data with location information. It assumes there exists some kind of spatial dependency within the data. Spatial statistical data can take on one of three forms, namely, point pattern data, lattice data, or geostatistical data (Cressie, 2015). Point pattern data, the only spatial data type considered in this research, is zero-dimensional and pertains to the location at which a data point occurred.

Point pattern analysis is a part of spatial statistics that considers whether or not there is an underlying pattern to the location at which data points (events) occur. A point pattern is a finite collection $x = \{x_1, x_2, ..., x_n\}$ of points $x_i \in \mathbb{R}^2$, (Baddeley et al., 2015). A point process, denoted by $X = \{X_1, X_2, ..., X_n\}$, is a random process whose realizations are point patterns, (Baddeley et al., 2015). Cressie (2015) and Baddeley et al. (2015) note that there are three types of patterns present in spatial point patterns; random, clustered and regular as can be seen in Figure 1. Random points (see Figure 1a) have an equal chance of occurring anywhere in the area under consideration, there is no discernible pattern to the points. Clustered points (see Figure 1b) are points that form groups. They attract some points and repel others. Regular points (see Figure 1c) are points that repel each other.

Corresponding author: Inger N. Fabris-Rotelli (inger.fabris-rotelli@up.ac.za)
MSC2020 subject classifications: 62H11, 62M30, 91C20

They do not occur completely randomly, but they also do not form groups and appear to be equally spaced.

Diggle (2013) defines Complete Spatial Randomness (CSR) as a point process where all events have equal probability of occurring anywhere in the area of interest and are independent of each other. Cressie (2015) notes that if the points are CSR they will be represented by a homogeneous Poisson point process. A homogeneous Poisson point process has the following properties: the points have no spatial location preference (homogeneity), and the points in one region have no influence on the spatial location of the points in another region (independence) (Baddeley et al., 2015; Cressie, 2015).

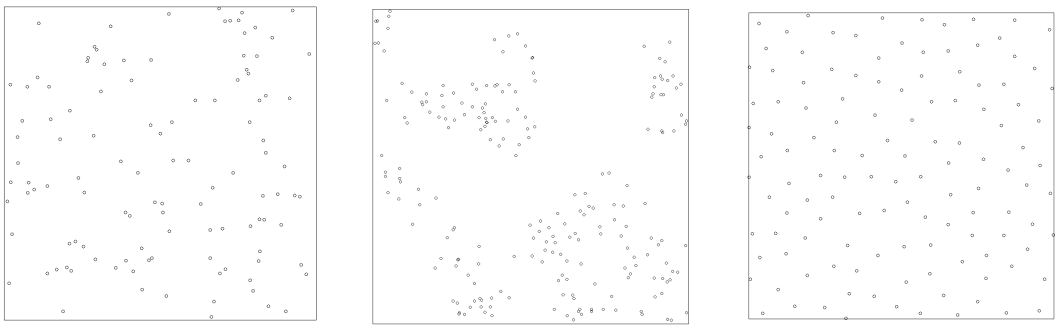
The typical focus of spatial point pattern analysis is the spatial arrangement of points (Baddeley et al., 2015). We want to establish whether or not the location of the datapoints tells us something more about the points, whether or not there is a pattern that can be picked up. To do this, we test for CSR by testing the null hypothesis that the points are randomly distributed, realised from a Poisson point process against the alternative hypothesis that the points are either clustered or regular (Moraga, 2023). Two methods are used namely, the quadrat method (Ripley, 1977), and Ripley's K-function (Ripley, 1976). Both of which tells us about the point patterns distribution.

This research explores clustered point patterns and their simulation. As will be discussed in more detail later, clustered point patterns are a versatile and valuable type of point pattern and as such, having accurate simulating functions is beneficial. Often times in real world scenarios, we do not have access to adequate data to make a conclusion about the nature of the point patterns, so we make use of simulating functions to create more data. Simulations and coded statistical methods will be done with the `spatstat` package in `R` (Baddeley et al., 2015). The `spatstat` package has the K-function, quadrat and nearest neighbour distance methods, as well as functions to simulate clustered point patterns; `rMatClust, rThomas, rVarGamma, rCauchy`. All of these functions will be investigated, and the robustness of each will be tested by changing the parameters required for the different point process models, simulation and fitting functions. We investigate different parameters to ensure the simulated models are both robust and statistically similar to the original point patterns. Additionally, identifying problematic parameters is beneficial, as it helps pinpoint potential problem areas.

In Section 2 we define and explain all the functions and methods we plan to use, as well as comparing the clustered point pattern types we will evaluate. Next, Section 3 explains the tests that we will perform, and discusses their results and the implications thereof. Finally, Section 4 we presents a methodology for simulating and fitting clustered point patterns.

2. Background Theory

Random point patterns have no spatial dependency (Baddeley et al., 2015). They are completely spatially random (CSR). Clustered point patterns, the focus of this research, form groups, or clusters (Baddeley et al., 2015). Points in the point pattern attract each other; so there is a clear spatial dependency. In regular point patterns, points repel each other (Baddeley et al., 2015), so there is a clear spatial dependency as they do not occur randomly within the area under consideration. As depicted in Figure 1, point patterns may be random, clustered or regular. This research will only consider clustered point patterns as they are particularly difficult to model, and due to the time constraints for this research we could only focus on them. The term complete spatial randomness is



(a) Random (b) Clustered (c) Regular

Figure 1. Types of point patterns.

used frequently when considering if a point process is clustered, regular or random. It forms the null hypothesis.

The quadrat method (Ripley, 1977) is one of the most common methods used in spatial statistics to determine whether a point pattern is CSR. The quadrat method works by partitioning the spatial region in which a point pattern is observed into non-overlapping quadrats of equal size. The test statistic based on the number of points in each quadrat is then computed. This is given by,

$$\chi^2 = \sum_{i=1}^{m} \frac{(n_i - n^*)^2}{n^*},$$

where n_i is the observed number of points for each quadrat $i = 1, 2, ..., m$, and n^* is the expected number of points in each quadrat. The expected number of points in each quadrat, assuming complete spatial randomness, is the total number of points divided by the number of quadrats. Under the null hypothesis, H_0 : the point pattern is CSR, the test statistic χ^2 is approximately $\chi^2(m-1)$ distributed. The quadrat method aids us in determining if the point pattern under consideration is clustered, regular or random.

Two additional methods used to indicate clustering in point patterns are the index of clumping (ICS) (David and Moore, 1954) and Ripley's K-function. The index of clumping gives us a way to quantifiably measure whether a point pattern is clustered, or regular, and is given by

$$ICS = \frac{s^2}{\bar{x}} - 1,$$

where $\bar{x}$ and s^2 are the sample mean and sample variance respectively. The idea of the index of clumping is that if the ICS is greater than one (Embarak, 2022), the point pattern is clustered, and if it is less than zero, the point pattern is regular (Ripley, 2005). If the ICS is very close to zero, the point pattern is CSR. This is because, if the point pattern has an underlying Poisson distribution, the mean and variance are approximately equal. Ripley's K-function tells us whether or not a point pattern is more clustered or regular than would be expected under CSR. Ripley (1977) defines the K-function in Definition 1.

Definition 1 (K-function)**.** If X is a stationary point process, with intensity $\lambda > 0$ then, for any $r \geq 0$

$$K(r) = \frac{1}{\lambda} E[\text{number of } r\text{-neighbours of } u | X \text{ has a point at location } u]$$

does not depend on the location of u, and is called the K-function, where λ, the intensity, is the amount of points per unit area, of the point process. Under the complete spatial randomness, the theoretical K-function attains a value of πr^2 (Baddeley et al., 2015). $K(r) = \pi r^2$ indicates CSR, $K(r) > \pi r^2$ indicates clustering, and $K(r) < \pi r^2$ indicates regularity in the points.

The empirical K-function (Ripley, 1977) is given by the following equation

$$\hat{K}(r) = \frac{|W|}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{1}\{d_{ij} \leq r\} e_{ij}(r),$$

where $|W|$ is the area of the observation window W, n is the number of points and, $\mathbf{1}\{d_{ij} \leq r\}$ is an indicator function which is equal to one if the distance between the point under consideration, x_i, and any other point, x_j, $j \neq i$, falls within the circle with radius r centered at the point x_i and zero otherwise. Here, $e_{ij}(r)$ is the edge correction weight which corrects for boundary effects imposed by the observation window. The Fry plot (Fry, 1979; Hanna and Fry, 1979), also referred to as the Patterson plot (Patterson, 1934, 1935) can be thought of as a visualisation of the K-function. It is essentially a scatter plot of all the differences between all possible point pairs in the point patterns and is useful in determining correlation in the point pattern.

A necessary and important assumption made by Poisson point process models is that of independence of points (Baddeley et al., 2015). Restating the properties of a Poisson point process, it can be seen that a Poisson point process model has homogeneous intensity. Meaning, the number of points expected to occur within any bounded region of space B is,

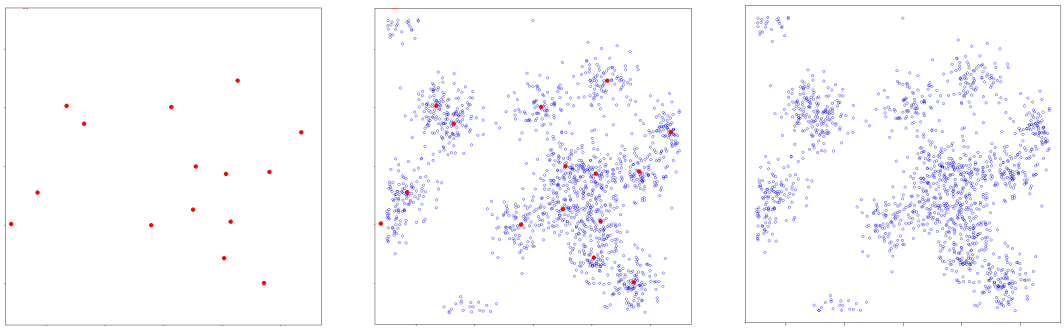
$$\mathbb{E}[n(X \cup B)] = \lambda \cdot |B|,$$

where λ is the intensity of the point process model. Baddeley et al. (2015) explains that an intensity function $\lambda(u)$ completed describes its Poisson point process and all that is needed to fit a Poisson point process model to a point pattern dataset is the form of the intensity function. Thus, all that is required to change the model, is to change the intensity.

The assumption of independence required by Poisson point process models does not hold for clustered data. As such, the models proposed by Cox (1955), Neyman and Scott (1958), Møller (2003), Møller and Torrisi (2005), Brix (1999) and Yau and Loh (2012) are used instead. The model proposed by Cox (1955) is a variation of the Poisson point process. The model put forth by Neyman and Scott (1958) is a special case of the Cox process, and all other models that will be considered in this research are generalisations of the Neyman-Scott process, referred to as shot noise Cox processes. The Cox process (Cox, 1955), also referred to as a *doubly stochastic Poisson process* is defined in Definition 2,

Definition 2 (Cox process)**.** Suppose that $Z = \{Z(\xi) : \xi \in S\}$ is a non-negative random field so that with probability one, $\xi \to Z(\xi)$ is a locally integrable function, If the conditional distribution of X given Z is a Poisson process on S with intensity function Z, then X is said to be a Cox process driven by Z.

$$X|Z \sim Poisson(Z).$$



(a) Parent points (b) Parent and offspring points (c) Offspring points

Figure 2. Parent points, denoted in red, and offspring points, denoted in blue, of a Thomas process.

Baddeley et al. (2015) explains that a Cox process is in essence a Poisson process whose intensity function is random. The Cox processes assume there is a random, underlying intensity function $\Lambda(u)$. If the intensity function were known, it would form a Poisson point process with an intensity function $\Lambda(u)$. A Neyman-Scott cluster process, commonly referred to as just a *cluster process* is created in two steps. Firstly, a set of parent points, a point process W, are created. Next, every parent point, w_i, generates a random point pattern of offspring points, x_{ij}. We only observe the offspring points in a cluster process (Baddeley et al., 2015), so we remove the parent points. This set, made up only of offspring points, is denoted by X. This process is shown in Figure 2, using a generalisation of the Neyman-Scott process, called the Thomas process (Thomas, 1949; Diggle, 1978). The Neyman-Scott model (Neyman and Scott, 1958) chooses the parents to be generated from a Poisson process with intensity γ, and the offspring to be generated with another Poisson process of intensity β. A number of assumptions are needed before we can continue. Namely, that clusters are independent and identically distributed, and offspring points within the clusters are themselves independent.

The set of Neyman and Scott (1958) clustering processes considered in this research are also known as shot noise Cox processes (Møller, 2003), defined in Definition 3,

Definition 3 (Shot noise Cox process)**.** Let X be a Cox process in $\mathbb{R}^m$ driven by

$$Z(\xi) = \sum_{(c,\gamma)\in\Phi} \gamma k(c,\xi),$$

where $k(\cdot,\cdot)$ is a kernel for a m-dimensional point process X and Φ is a Poisson point process on $\mathbb{R}^2 \times (0,\infty)$ with a locally integrable intensity function ζ. Then X is called a shot noise Cox process (SNCP).

Shot noise processes, and generalised shot noise processes, as presented by Møller (2003). Møller and Torrisi (2005) are incredibly useful and versatile classes of point process models for clustered point patterns. Møller and Torrisi (2005) explains that generalised shot noise processes are shot noise processes that are extended in two key ways. The first, the parent points are not necessarily Poisson processes. The second, the kernel of the shot noise process can be random. An example of this can be seen with the generalised Neyman-Scott (GNS) process from Yau and Loh (2012). Examples

Table 1. Underlying distribution of parent and offspring points of generalisations of the Neyman-Scott process.

Point process model	**Parents**	**Offspring**
Neyman-Scott	Independent Poisson process, intensity κ	Independent Poisson process, intensity μ
Variance gamma	Independent Poisson process, intensity κ	Poisson(μ), independently and uniformly according to a Variance Gamma kernel
Cauchy	Independent Poisson process, intensity κ	Poisson (μ), independently and uniformly according to a Cauchy kernel
Thomas	Independent Poisson process, intensity κ	Poisson(μ), with the positions being isotropic Gaussian displacements from the cluster parent location
Matérn	Independent Poisson process, intensity κ	Poisson(μ), independently and uniformly in a disc centered around parent
Yau and Loh (2012) GNS	Strauss process	Poisson process, intensity μ

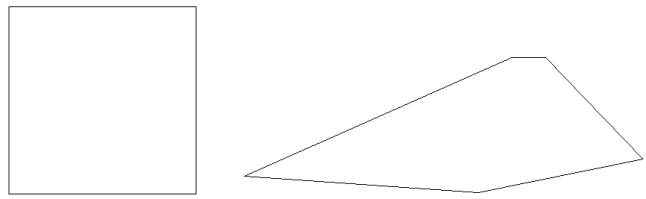
include Neyman-Scott processes (Neyman and Scott, 1958), shot noise G Cox processes (Brix, 1999) and Poisson-gamma processes (Wolpert and Ickstadt, 1998). Table 1 compares the Neyman-Scott process with generalisations of the Neyman-Scott process.

The methods discussed above will all be used to both test the robustness of point pattern fitting functions, and to evaluate the differences between an original point pattern dataset and simulated point pattern from a fitted model. In Section 3, we will explain the impact that different parameters have on the fitting functions, and how they can be changed to achieve different point patterns. Furthermore, these differences will be tested to determine if they are significant.

3. Application

This research aims to investigate the robustness of the available cluster models. We discuss point process fitting functions in the `spatstat` library (Baddeley et al., 2015). Specifically making use of the `kppm` function which fits a Neyman-Scott or Cox cluster process model. Cox and cluster processes are used when there is positive association (clustering) between the points in a point pattern. These models and generalisations thereof are the only ones that will be considered in this research. Using `kppm` a model can be fitted in `spatstat` and the parameter values of the point pattern are estimated.

The robustness of a simulated fitted model will be evaluated using the Kolmogorov-Smirnov test (Kolmogorov, 1933; Darling, 1957), via the use of the `ks.test()` function in `R`, to determine if the underlying distributions are the same. This is done by comparing the K-function of the simulated



(a) Square (b) Convex

Figure 3. Two windows used to simulate point patterns.

point patterns to the K-function of the original point patterns.

$$D = \max|F_1(X) - F_2(X)|,$$

where F_1 and F_2 are the K-functions of the two point patterns being compared. The K-function can be used as a summary of a point pattern (Ripley, 1976). In `R`, `$un` returns the uncorrected estimate of the K-function, the value calculated without edge correction. Since the K-function is a cumulative function, the values stored in `$un` can be thought of as CDF (cumulative density function) values and thus the Kolmogorov-Smirnov test can easily be applied to determine if the underlying distribution of two point patterns is the same. A test statistic and p-value are returned that determine if the point patterns come from the same underlying distribution.

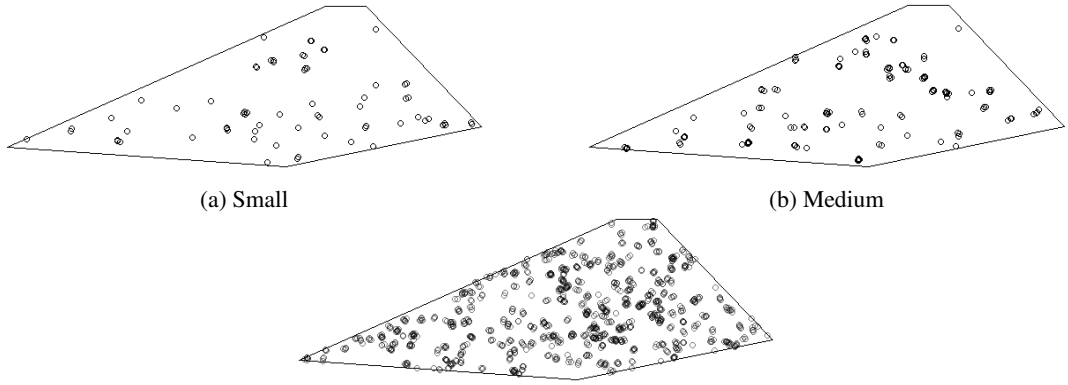
To obtain the results that will be discussed in this research, a seed value of 303 and 404 were used in the original simulation and resimulation of the point patterns respectively. Explanation of the results will only consider the Matern Cluster case. Thomas Cluster, Variance-Gamma and Cauchy point process models and the results for these as well as all tables referred to and discussed in this section of the research can be found in an appendix pdf[1].

Two windows are considered. The first, Figure 3a, is a square with area 100 and the second, Figure 3b, is a convex shape with area 100.1. The windows are of different shapes, but approximately the same area, to determine if the window shape plays a role in the point pattern creation, and parameter estimation. In real world scenarios, it will not always be the case that the observation window under consideration is a square, or uniform shape, so it is beneficial to investigate the impact of irregularly shaped windows in clustered point pattern simulation. While this is not done in this research, it should be considered in future work.

Three sizes of point patterns are considered, to determine the effect of size on the point pattern creation and resimulation, namely Small: $50 \leq$ no. points ≤ 100 (see Figure 4a), medium: $100 <$ no. points ≤ 500 (see Figure 4b) and large: no. points ≥ 1000 (see Figure 4c).

Each point pattern type (Matern, Thomas, Variance-Gamma and Cauchy) have three parameters; namely, `kappa`, `scale` and `mu`. `Kappa` is the intensity of the underlying parent process, `scale` is the radius of the offspring clusters around the parent points and `mu` is the expected number of points in each cluster. Two values for each of these parameters are considered, specifically the boundary values which would give point patterns with a number of points very close to the size boundaries. The

[1] The code to produce the results discussed in this research is available at this link. The full tables, for all four point pattern types, can be found here. And, the results referred to but not included in this research can be found here.



(a) Small (b) Medium

(c) Large

Figure 4. Matern point patterns displaying the three point pattern sizes in a convex window.

base model parameters values (`kappa`, `scale` and `mu`) for the small, medium and large point patterns are: $(0.9, 0.1, 0.9)$, $(1.5, 0.1, 1.5)$ and $(4, 0.1, 4)$ respectively. For a small point pattern, the range of `kappa`, `scale` and `mu` values we consider are, $(0.8, 1.1)$, $(0.1, 100)$ and $(0.8, 1.4)$ respectively. Similarly, for a medium point pattern we consider $(0.8, 3)$, $(0.1, 100)$ and $(0.8, 3)$ respectively. And lastly, for a large point pattern we consider $(2.3, 5)$, $(0.1, 100)$ and $(2.5, 5)$ respectively. In each case we use the boundary values to simulate the actual point pattern. When each parameter value is selected, its value is changed from the base model value to the boundary value of the appropriate interval, and the other two parameters keep their base model values. This is repeated for the different window shapes and point pattern sizes.

In summary, we have four point pattern types, two windows, three point pattern sizes, three parameters each with two values. Which results in 144 original point patterns meaning 36 original point patterns per point pattern type.

The original Matern point patterns are given in Table 2. The index refers to the number of point pattern that is created, in all further tables the indices are the same and refer to the same point pattern. In this case the point pattern is Matern Cluster and the size and window refers to the size and window of the point pattern in the i^{th} index. The parameter column indicates the parameter that is changed to a boundary value and the `kappa`, `scale` and `mu` value columns store the values of the parameters for the i^{th} point pattern.

There are three ways the robustness of the fitting and simulation functions is tested. The first, by considering the number of points in the point patterns. The second, by comparing the original parameter values to the fitted parameter values. And, the third by conducting a Kolmogorov-Smirnov test on the simulated and fitted K-functions to determine if the underlying distributions are the same. Where applicable, a 10% significance level is used.

To test if there is a significant difference in the total points in the point pattern we add 9×9 quadrats over the point patterns, the number of points in each quadrat is determined and we then sum over all the quadrats to get the total points in the point pattern.

Using the method explained above, Table 3, is obtained and displays the index number; in accordance with the index numbers in Table 2 and the number of points in the point pattern. If the

number of points between the rectangular-shaped point pattern and the convex-shaped point pattern is significantly different the row is flagged for further investigation. In each case 10% of the upper boundary for the point pattern size is considered significant. That is, 10 for small, 50 for medium and 100 for large. Applying this rule to Table 3 we obtain Table 4, of the point patterns where the difference in points is significant. The second column is the number of points in the point pattern at the index in the index column. The third column is the number of points in the point pattern at the index value plus 1, and the fourth column is the absolute value of the difference between the number of points in the two point patterns.

Interestingly the quantity of points in the convex shape is larger than the quantity of points in the rectangular shape three out of the four times, which is not what we would expect. As there are more sharp points in the convex shape we expect it to be more difficult to fit the points in the window. Each row in Table 4 is resimulated 10 times to determine if the difference is an abnormality or if it always occurs. Table 5, and Table 6, show the 10 resimulations of rows 7 and 23 respectively.

The amount of times the difference exceeds 10 in Table 5, is four and the amount of times the difference is greater than 50 in Table 6, is five. Thus, in index 7 the difference is an abnormality, while in index 23 the difference occurs half the time so there is an issue with point pattern creation at that index. Out of four flagged rows, two are for small point patterns, one for a medium and one for a large, which is to be expected since it makes sense for smaller point patterns to be less robust to changes.

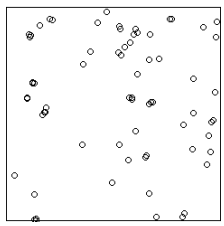
The original point patterns are fitted using `kppm` for two cases. The first where the point pattern type is specified, in this case as Matern, and the second where the point pattern type is unspecified. In Table 7, Table 8 and Table 8, the original parameter values are compared to the parameter values for the fitted point pattern where the point pattern type is specified and to the parameter value for the fitted point pattern where the point pattern type is unspecified.

When all three parameter values are significantly different; meaning the difference is larger than 10%; the point pattern is investigated further. Four of the flagged rows are resimulated 10 times to test if the differences are abnormal or not and the results for one of the resimulations; performed on row 9; are stored in Table 10, Table 11 and Table 12. The second column in each table stores the original parameter value; from the original point pattern; which is compared to the third and fourth columns.

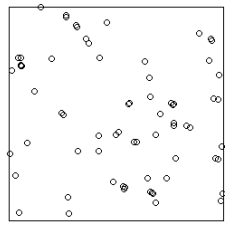
These columns store the parameter values when the point pattern is fitted and specified and when it is fitted and unspecified. If we compare these values we find that in six the kappa and scale values differ significantly and in eight cases the mu values differ significantly. All three parameter values differ significantly five times in both cases. Meaning the resimulation indicates that there is a $50 : 50$ chance this is not an abnormality, so further investigation should be done.

To determine if the original point pattern and the fitted point patterns are from the same distribution we get the K-functions and perform a Kolmogorov-Smrinov test on the K-function values, Figure 5 shows this process. Figure 5a is an original Matern point pattern and Figures 5b and 5c are the resimulated Matern point patterns when the point pattern type is specified and unspecified in `kppm`, respectively. Figures 5d, 5e and 5f are the K-functions for the original, resimulated and specified and resimulated and unspecified point patterns respectively.

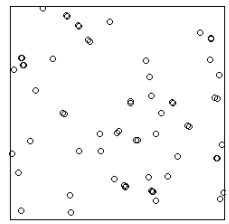
In Table 13, point pattern 1 is the point pattern type of the point pattern that was originally created, point pattern 2 is the fitted point pattern where the point pattern type is specified. And, point pattern



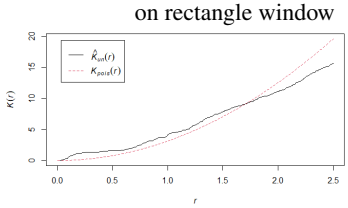
(a) Matern point pattern on rectangle window



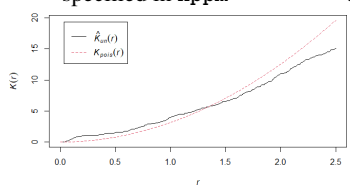
(b) Resimulated, type specified in `kppm`



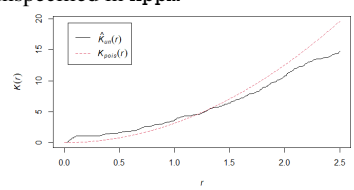
(c) Resimulated, type unspecified in `kppm`



(d) K-function of the original Matern point pattern



(e) K-function of resimulated, type specified



(f) K-function of resimulated, type unspecified

Figure 5. Comparison of K-functions of original and resimulated (where type is both specified and unspecified in `kppm`) point patterns for Kolmogorov-Smirnov test.

3 is the fitted point pattern where the point pattern type is unspecified. PV12 is the p-value of the Kolmogorov-Smirnov test of the K-functions of point pattern 1 and point pattern 2. From Table 13, it is evident that the Kolmogorov-Smrinov test rejects more often when the point pattern size is small, and when the point pattern size is large the null hypothesis is almost always not rejected.

In Table 14, 10 resimulations of the i^{th} row of Table 13 are displayed. The first column stores the indices of the original point pattern and the resimulations. The second column stores which point patterns are not equal, in this case "$1 \neq 3$" means that the original point pattern, denoted here with a "1", does not have the same underlying distribution as the fitted and resimulated point pattern where the point pattern type is not specified, denoted here with a "3". The third and fourth columns store the point pattern type of the original point pattern, point pattern 1, and the point pattern type of the resimulated point pattern where the point pattern type is unspecified, point pattern 3, respectively. The final column stores the p-values from the Kolmogorov-Smrinov test.

As the scale values do not seem to have an influence on the number of points, a table of ICS values are obtained to determine if there is an effect on the "clusteredness" of the point pattern when the scale values vary. Table 15, shows that there does not seem to be a significant change in the "clusteredness" of the point patterns when the scale values are changed. As such, there is no clear reason why the scale values have such a large difference when fitted.

Some additional interesting things that are observed in the simulated data include:

- In all cases when the point patterns are fitted and the point pattern type is unspecified, `kppm` chooses Thomas.
- In the Matern and Thomas cases, the rectangular window is the window on which most original point patterns experience significant differences.

- In the Variance-Gamma and Cauchy cases, the convex window is the window on which most original point patterns experience significant differences.
- In all cases the small point patterns observe the largest number of significant differences, roughly 45% of the time the differences are from a small point pattern.
- In the Matern, Thomas and Cauchy cases, scale is the parameter which causes the largest number of significant differences, roughly 58% of the time the differences are from a point pattern in which the scale value is changed.
- In the Variance-Gamma case, kappa is the parameter which causes the largest number of significant differences.
- In all cases, the amount of times point pattern 1 equals point pattern 2 is the same as the amount of times point pattern 1 equals point pattern 3. Based on the results of the Kolmogorov-Smirnov test results.

The Thomas point pattern is not the default point pattern type for `kppm`, as no default is specified in the write up of the function. However, when the point pattern type is not specified it is always classified as a Thomas process by the `kppm` function, no matter what the original point pattern type is. This is something that should be investigated further.

When simulating clustered point patterns it is beneficial to simulate many times over a number of different windows, especially if the point pattern is small. When fitting clustered point patterns using `kppm`, all four point pattern types should be fitted and compared to each other and the best one chosen. It should not be left up to `kppm` to determine the best point pattern type, as it will always be classified a Thomas.

4. Conclusion

In this research, we considered the robustness of fitting and simulating functions for clustered point pattern models. A total of 36 simulated clustered point patterns were created using different windows, point pattern sizes and parameter values. The clustered point patterns were then fitted to test the robustness of the fitting functions by comparing the fitted parameter values to the original ones. Using the fitted parameter values, the clustered point patterns were resimulated to test the robustness of the simulating functions by comparing the underlying distributions of the simulated and original clustered point patterns using a Kolmogorov-Smirnov test on the K-functions. The robustness tests were all performed using a 10% level of significance. This gives an indication of how well the fitting and simulating functions work.

When simulating and fitting clustered point patterns, it is beneficial to simulate using multiple different point patterns, windows and sizes. And when fitting models the same is true. Models should be fitted where the clustered point pattern type is specified, and where it is unspecified and the results should be compared to each other to determine which is the best fit. As there is no benefit in specifying or not specifying the clustered point pattern type in `kppm`, both cases should always be considered. As was seen, small clustered point patterns produce the most differences, so extra care should be taken when working with clustered point patterns with less than 100 points. The window does not seem to have a significant effect on the results, but it is still safer to simulate on a few different

window shapes. The results should be compared in a number of ways. By considering the difference in number of points, difference in the original and fitted parameter values and by comparing the underlying distributions. It is clear that `kppm` chooses Thomas as the underlying clustered point process the vast majority of the time, even when the Thomas fit is a worse fit than the fit when the clustered point pattern type is specified. Small clustered point patterns do not always behave, so extra care should be taken when simulating and fitting them. Large clustered point patterns are more robust and can withstand more parameter changes.

The proposed methodology when simulating or fitting clustered point pattern is as follows:

1. Simulate many point patterns on different windows.
2. Fit using all four point pattern types.
3. Resimulate using the fitted values many times.
4. Compare the fitted parameter values to the original parameter values.
5. Perform a Kolmogorov-Smirnov test on the K-functions to compare the original and resimulated point patterns.
6. Select the resimulated point pattern that best fits the original point pattern.

This research should be extended to find out why the Thomas process is always classified as the default point pattern when no point pattern is specified in `kppm`, and perhaps new methodology can be written to better determine the most appropriate point pattern type. Further, it would be beneficial to investigate the impact of varying window sizes and shapes, as well as considering irregular window shapes. Only similarly sized windows were considered in this research, but it may be worthwhile to consider the impact of a small window against a larger one. Lastly, the scale parameter should be further investigated to determine what it impacts in the point pattern and why the fitted estimates are so different to the original parameter values.

Acknowledgements

This work is partially based upon research supported by the South Africa National Research Foundation (NRF) Grant number 137785. Opinions expressed and conclusions arrived at are those of the authors and are not necessarily to be attributed to the NRF. Ethics number NAS116/2019.

And my heartfelt thanks to my supervisors, I truly appreciate all your feedback and support throughout this research.

References

Baddeley, A., Rubak, E., and Turner, R. (2015). *Spatial Point Patterns: Methodology and Applications with R*. CRC press.

Brix, A. (1999). Generalized gamma measures and shot-noise Cox processes. *Advances in Applied Probability*, **31** (4), 929–953.

Cox, D. R. (1955). Some statistical models related with series of events. *Journal of the Royal Statistical Society Series B*, **17**, 129–164.

Cressie, N. (2015). *Statistics for Spatial Data*. John Wiley & Sons.

Darling, D. A. (1957). The Kolmogorov-Smirnov, Cramer-Von Mises tests. *The Annals of Mathematical Statistics*, 823–838.

David, F. N. and Moore, P. G. (1954). Notes on contagious distributions in plant populations. *Annals of Botany*, **18** (69), 47–53.

Diggle, P. J. (1978). On parameter estimation for spatial point processes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **40** (2), 178–181.

Diggle, P. J. (2013). *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns*. CRC press.

Embarak, M. (2022). Spatial distribution of thrips tabaci lindeman (thysanoptera: Thripidae) on onion plants at different irrigation intervals. *Egyptian Academic Journal of Biological Sciences. A, Entomology*, **15** (4), 47–55.

Fry, N. (1979). Random point distributions and strain measurement in rocks. *Tectonophysics*, **60** (1-2), 89–105.

Hanna, S. and Fry, N. (1979). A comparison of methods of strain determination in rocks from southwest Dyfed (Pembrokeshire) and adjacent areas. *Journal of Structural Geology*, **1** (2), 155–162.

Kolmogorov, A. N. (1933). Sulla determinazione empirica di una legge didistribuzione. *Giornale dell'Istituto Italiano degli Attuari*, **4**, 89–91.

Møller, J. (2003). Shot noise Cox processes. *Advances in Applied Probability*, **35** (3), 614–640.

Møller, J. and Torrisi, G. L. (2005). Generalised shot noise Cox processes. *Advances in Applied Probability*, **37** (1), 48–74.

Moraga, P. (2023). *Spatial Statistics for Data Science: Theory and Practice with R*. Chapman & Hall/CRC Data Science Series. CRC Press.

Neyman, J. and Scott, E. L. (1958). Statistical approach to problems of cosmology. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **20** (1), 1–29.

Patterson, A. L. (1934). A Fourier series method for the determination of the components of interatomic distances in crystals. *Physical Review*, **46** (5), 372.

Patterson, A. L. (1935). A direct method for the determination of the components of interatomic distances in crystals. *Zeitschrift für Kristallographie-Crystalline Materials*, **90** (1-6), 517–542.

Ripley, B. D. (1976). The second-order analysis of stationary point processes. *Journal of Applied Probability*, **13** (2), 255–266. doi:10.2307/3212829.

Ripley, B. D. (1977). Modelling spatial patterns. *Journal of the Royal Statistical Society: Series B (Methodological)*, **39** (2), 172–192.

Ripley, B. D. (2005). *Spatial Statistics*. John Wiley & Sons.

Thomas, M. (1949). A generalization of Poisson's binomial limit for use in ecology. *Biometrika*, **36** (1/2), 18–25.

Wolpert, R. L. and Ickstadt, K. (1998). Poisson/gamma random field models for spatial statistics. *Biometrika*, **85** (2), 251–267.

Yau, C. Y. and Loh, J. M. (2012). A generalization of the Neyman-Scott process. *Statistica Sinica*, 1717–1736.

Submitted to the Proceedings of the 65th Annual Conference of the South African Statistical Association

An approximate Bayesian computation threshold-search algorithm for parameter estimation

Neill Smit

Centre for Business Mathematics and Informatics, North-West University, South Africa

In this paper, a new approximate Bayesian computation (ABC) algorithm based on rejection sampling is introduced, where the tolerance threshold is adaptively adjusted as candidate particles are accepted or rejected. The adaptive threshold eliminates the importance of choosing a suitable fixed tolerance threshold and results in the acceptance of more preferable candidate particles via appropriate hyperparameter choices. This modification can also act as a search mechanism for determining a suitable fixed tolerance threshold for the standard ABC rejection sampling algorithm. By means of a simulation study on parameter estimation for widely used life distributions, it is shown that the new ABC algorithm has comparable performance to maximum likelihood estimation.

Keywords: Approximate Bayesian computation, Life distributions, Maximum likelihood estimation, Parameter estimation, Reliability.

1. Introduction

Approximate Bayesian computation (ABC) is a modern class of likelihood-free methods formally introduced by Beaumont et al. (2002), following on the work of Tavaré et al. (1997) and Pritchard et al. (1999). Since ABC methods attempt to directly approximate the marginal posteriors without the closed-form specification of the likelihood function, these methods are particularly useful in cases where the likelihood function is difficult to compute analytically or even computationally intractable. Other Bayesian approximation methods, such as Markov chain Monte Carlo methods and variational Bayesian methods, require the explicit specification and evaluation of the likelihood function (see, for example, Brooks et al., 2011; Blei et al., 2017). ABC is now widely used to perform parameter estimation, model selection, and other inferences for complex problems in fields such as quantitative finance, molecular epidemiology, systems biology, population genetics, ecological modelling, and nuclear imaging (see, for example, Sisson et al., 2019).

The basic idea behind ABC methods and how they can be used to perform parameter estimation is quite simple. A candidate particle, which is a set of potential model parameters, is generated from the prior distributions and used to simulate a candidate dataset from a specified model. The level of agreement between the simulated data and the observed data is then evaluated using some distance function or summary statistic. The candidate particle is considered acceptable if the distance function

Corresponding author: Neill Smit (neillsmit1@gmail.com)
MSC2020 subject classifications: 62F15, 62N02, 62N05, 65C60

or summary statistic indicates a high level of agreement. A large number of acceptable particles are generated until some convergence is achieved and then used to approximate the marginal posteriors.

Bayesian inference relies on the use of Bayes' theorem, which is given by

$$\mathrm{P}(\Theta \,|\mathcal{D}) = \frac{\mathrm{P}(\mathcal{D}\,|\Theta)\,\mathrm{P}(\Theta)}{\mathrm{P}(\mathcal{D})},$$

where Θ is the parameter vector associated with some model $\mathcal{M}$, $\mathcal{D}$ is the observed data, $\mathrm{P}(\Theta|\mathcal{D})$ is the posterior, $\mathrm{P}(\mathcal{D}|\Theta)$ is the likelihood, $\mathrm{P}(\Theta)$ is the prior, and $\mathrm{P}(\mathcal{D})$ is the normalising constant. Analytical parameter estimation in the Bayesian paradigm is often not possible, since this typically requires a mathematically tractable likelihood and prior, as well as an explicit solution for the normalising constant.

In the ABC setup using some distance function $g(\cdot)$, the discrepancy between the observed data $\mathcal{D}$ and the simulated data $\mathcal{D}^*$ can be assessed by comparing $g(\mathcal{D}, \mathcal{D}^*)$ to some tolerance threshold $\epsilon > 0$. A candidate particle is acceptable if $g(\mathcal{D}, \mathcal{D}^*) \leq \epsilon$, where $\mathcal{D}^*$ is simulated using the candidate particle as the parameters for the model $\mathcal{M}$. Considering the distance function rather than only the observed data, Bayes' theorem can be modified to

$$\mathrm{P}(\Theta \,|g(\mathcal{D}, \mathcal{D}^*) \leq \epsilon) = \frac{\mathrm{P}(g(\mathcal{D}, \mathcal{D}^*) \leq \epsilon \,|\Theta)\,\mathrm{P}(\Theta)}{\mathrm{P}(g(\mathcal{D}, \mathcal{D}^*) \leq \epsilon)}.$$

ABC algorithms allow for the direct simulation from $\mathrm{P}(\Theta|g(\mathcal{D}, \mathcal{D}^*) \leq \epsilon)$, where inference is then based on approximate marginal posteriors consisting of a large number of acceptable particles. The level of approximation is determined by the tolerance threshold ϵ, where $\mathrm{P}(\Theta|g(\mathcal{D}, \mathcal{D}^*) \leq \epsilon)$ converges to $\mathrm{P}(\Theta|\mathcal{D})$ as $\epsilon \to 0$.

A major limitation of the standard ABC rejection sampling (ABC-RS) algorithm is that is has a fixed tolerance threshold which should be carefully chosen. If the tolerance threshold is too small, the algorithm might take too long to generate accepted particles, whereas if the tolerance threshold is too large, undesirable particles might be accepted (i.e., particles for which the discrepancy between the observed and simulated data is not small enough). Several improvements and modifications of the ABC-RS algorithm, as well as alternative algorithms based on importance sampling, Markov chain Monte Carlo methods, and sequential Monte Carlo methods, have been suggested in the literature (see, for example, Abdessalem et al., 2019; Sisson et al., 2019). However, these algorithms are often complicated or have several tuning parameters which should be chosen appropriately.

In this paper, a simple modification of the ABC-RS algorithm is introduced, called the ABC threshold-search (ABC-TS) algorithm. The ABC-TS algorithm overcomes the fixed-threshold limitation of the ABC-RS algorithm by adaptively adjusting the tolerance threshold as candidate particles are accepted or rejected. This modification enables the acceptance of preferable candidate particles and can act as a search mechanism for determining an appropriate fixed tolerance threshold for the ABC-RS algorithm. Furthermore, there is a clear optimal choice for the tuning parameters of the ABC-TS algorithm, where suitable adjustments can be made to increase the computational efficiency of the algorithm. A simulation study is conducted to compare the performance of the ABC-TS algorithm against maximum likelihood estimation (MLE). In the simulation study, parameter estimation is performed for life distributions widely used in reliability analysis.

The layout of the paper is as follows. In Section 2, the ABC-TS algorithm and the distance functions used are defined. The threshold-search property of the ABC-TS algorithm is also illustrated. Section 3

Algorithm 1 Standard ABC-RS algorithm.

1. Input required: data $\mathcal{D}$, model choice $\mathcal{M}$, tolerance threshold ϵ, number of accepted particles N, distance function $g(\cdot)$, priors $\mathrm{P}(\theta_s|\mathcal{M})$, $s = 1, ..., S$.
2. For $i = 1$ to N do
   - Sample a candidate particle Θ^* from the priors $\mathrm{P}(\theta_s|\mathcal{M})$, $s = 1, ..., S$.
   - Simulate a candidate dataset $\mathcal{D}^*$ from $F(\cdot|\Theta^*, \mathcal{M})$.
   - Determine the value of $g(\mathcal{D}, \mathcal{D}^*)$. Then, for the tolerence threshold ϵ do
     - If $g(\mathcal{D}, \mathcal{D}^*) \leq \epsilon$, accept and store $\Theta^{(i)} = \Theta^*$. Set $i = i + 1$.
     - If $g(\mathcal{D}, \mathcal{D}^*) > \epsilon$, reject and discard Θ^*. Set $i = i$.
3. Output provided: approximate marginal posteriors for $\theta_1, ..., \theta_S$ from $\{\Theta^{(1)}, ..., \Theta^{(N)}\}$.

provides a short overview of the life distributions and their associated log-likelihood functions, which are used to perform MLE. The simulation settings and the performance of the ABC-TS algorithm against MLE based on specified metrics, are discussed in Section 4. The paper is concluded with some final remarks in Section 5.

2. The Modified ABC-TS Algorithm

2.1 ABC algorithms and notation

Before defining the ABC-TS algorithm, some notation used throughout the paper is introduced. Suppose that one has an observed dataset $\mathcal{D}$, a model choice $\mathcal{M}$ with its associated parameter vector $\Theta = \{\theta_1, ..., \theta_S\}$, and priors for the model parameters $\mathrm{P}(\theta_s|\mathcal{M})$, $s = 1, ..., S$. A candidate particle sampled from the priors is denoted by $\Theta^* = \{\theta_1^*, ..., \theta_S^*\}$ and the dataset simulated from the model $\mathcal{M}$ using the candidate particle is denoted by $\mathcal{D}^*$. Let N denote the number of particles that should be accepted by the ABC algorithm using some distance function $g(\cdot)$.

Let us first consider the standard ABC-RS algorithm, provided in Algorithm 1, where a fixed tolerance threshold $\epsilon > 0$ is selected (see, for example, Beaumont et al., 2002). A candidate particle is sampled from the priors and used to simulate a candidate dataset from the given model. The candidate particle is accepted if $g(\mathcal{D}, \mathcal{D}^*) \leq \epsilon$ and rejected if $g(\mathcal{D}, \mathcal{D}^*) > \epsilon$. This process is repeated until N particles are accepted, which can then be used to approximate the marginal posteriors. The importance of the choice for ϵ is clear from the algorithm, where an inappropriate tolerance threshold may result in either very few particles being accepted or undesirable particles being accepted.

The modified ABC-TS algorithm, where the tolerance threshold is adaptively adjusted as candidate particles are accepted or rejected, is defined in Algorithm 2. The algorithm has three tuning parameters, which include an initial tolerance threshold $\epsilon_1 \geq 0$, and two tolerance threshold adjustments $\delta_{\text{Accept}} > 0$ and $\delta_{\text{Reject}} > 0$. Due to the high rejection rate of rejection sampling-based ABC algorithms, it is required that $\delta_{\text{Reject}} \ll \delta_{\text{Accept}}$ to ensure convergence of the algorithm. The

Algorithm 2 Modified ABC-TS algorithm.

1. Input required: data $\mathcal{D}$, model choice $\mathcal{M}$, initial tolerance threshold ϵ_1, tolerance threshold adjustments δ_{Accept} and δ_{Reject}, number of accepted particles N, distance function $g(\cdot)$, priors $\text{P}(\theta_s|\mathcal{M})$, $s = 1, ..., S$.
2. For $i = 1$ to N do
   - Sample a candidate particle Θ^* from the priors $\text{P}(\theta_s|\mathcal{M})$, $s = 1, ..., S$.
   - Simulate a candidate dataset $\mathcal{D}^*$ from $F(\cdot|\Theta^*, \mathcal{M})$.
   - Determine the value of $g(\mathcal{D}, \mathcal{D}^*)$. Then, for the current tolerence threshold ϵ_i do
     - If $g(\mathcal{D}, \mathcal{D}^*) \leq \epsilon_i$, accept and store $\Theta^{(i)} = \Theta^*$. Set $\epsilon_i = \epsilon_i - \delta_{\text{Accept}}$, then set $i = i + 1$.
     - If $g(\mathcal{D}, \mathcal{D}^*) > \epsilon_i$, reject and discard Θ^*. Set $\epsilon_i = \epsilon_i + \delta_{\text{Reject}}$, then set $i = i$.
3. Output provided: approximate marginal posteriors for $\theta_1, ..., \theta_S$ from $\{\Theta^{(1)}, ..., \Theta^{(N)}\}$.

ABC-TS algorithm is executed similar to the ABC-RS algorithm, with the exception of the tolerance threshold at each iteration being adjusted based on whether a candidate particle was accepted or rejected in the preceding iteration. The tolerance threshold is decreased by δ_{Accept} if a particle was accepted in the previous iteration and increased by δ_{Reject} if a particle was rejected in the previous iteration.

Although the ABC-TS algorithm has three tuning parameters, there are optimal choices in terms of accepting the most preferable particles. The optimal setting effectively eliminates the importance of choosing suitable tuning parameters, but may significantly increase the computational cost. From Algorithm 2, the optimal tuning parameter choices are $\epsilon_1 = 0$, $\delta_{\text{Accept}} = \epsilon_i$, and δ_{Reject} as small as the available computational power allows. That is, after a particle is accepted, the tolerance threshold is set to zero and then very gradually increased for each rejected particle as the algorithm searches for the next preferable particle to accept. Depending on the specific application, the tuning parameters can however be adjusted to a certain degree in order to increase the computational efficiency without significantly decreasing the level of approximation.

Furthermore, the ABC-TS algorithm can act as a search mechanism for an appropriate fixed tolerance threshold in the ABC-RS algorithm. Through suitable choices for the tuning parameters of the ABC-TS algorithm, a convergence level can be identified by monitoring the tolerance thresholds at which candidate particles are accepted. This converged tolerance threshold should then be an appropriate choice for the fixed tolerance threshold in the ABC-RS algorithm. The result is that, once convergence is achieved through the ABC-TS algorithm, one can revert back to the ABC-RS algorithm using the identified ϵ to reduce the computational cost.

2.2 Distance functions

There are several approaches in the ABC literature for selecting a suitable distance function or summary statistic (see, for example, Lintusaari et al., 2017). In this paper, the distance functions considered are based on the discrepancy between the cumulative distribution function (CDF),

denoted by $F(x)$, and the empirical cumulative distribution function (ECDF), denoted by $F_n(x)$. The Kolmogorov-Smirnov (KS), Cramér-von Mises (CvM), and Anderson-Darling (AD) distance functions are used in this paper, since they have explicit expressions for an ordered finite sample (see, for example, Stephens, 1974).

The KS distance function measures the maximum absolute difference between the ECDF and the CDF. The KS distance function is defined as

$$g_{KS} = \sup_x \left| F_n(x) - F(x) \right|,$$

where for an ordered sample, $x_{(1)} < x_{(2)} < \cdots < x_{(n)}$, the KS distance function can be written as

$$g_{KS} = \max\left[\max_{1 \le i \le n} \left(\frac{i}{n} - F\left(x_{(i)}\right) \right), \max_{1 \le i \le n} \left(F\left(x_{(i)}\right) - \frac{i-1}{n} \right) \right].$$

The CvM distance function measures the difference between $F(x)$ and $F_n(x)$ over the domain of F, where more weight is placed on the centre of the distribution. The CvM distance function is defined as

$$g_{CvM} = n \int \left(F_n(x) - F(x) \right)^2 dF(x).$$

For an ordered sample, $x_{(1)} < x_{(2)} < \cdots < x_{(n)}$, the CvM distance function has the closed-form expression

$$g_{CvM} = \frac{1}{12n} + \sum_{i=1}^{n} \left(\frac{2i-1}{2n} - F\left(x_{(i)}\right) \right)^2.$$

The AD distance function is an extension of the CvM distance function, where more weight placed on the tails of the distribution. The AD distance function is defined as

$$g_{AD} = n \int \frac{\left(F_n(x) - F(x) \right)^2}{F(x)\left(1 - F(x)\right)} dF(x).$$

Considering an ordered sample, $x_{(1)} < x_{(2)} < \cdots < x_{(n)}$, the AD distance function simplifies to

$$g_{AD} = -n - \frac{1}{n} \sum_{i=1}^{n} (2i-1) \left[\ln\left(F\left(x_{(i)}\right)\right) + \ln\left(1 - F\left(x_{(n+1-i)}\right)\right) \right].$$

2.3 Threshold-search property

To illustrate the use of the ABC-TS algorithm and its threshold-search property, consider a simple example using the exponential distribution with parameter λ. Details on the sample size, true value of λ, and the prior construction are not important for the discussion of the threshold-search property. In Figure 1, we observe plots of the tolerance threshold at which 200 particles are accepted under various settings of the hyperparameters and using the CvM distance function. Throughout the discussion that follows, note that particles accepted at lower tolerance thresholds are typically more preferable particles, since the discrepancy between the observed data $\mathcal{D}$ and the simulated data $\mathcal{D}^*$ is typically smaller.

The stability of the tolerance threshold is investigated in Figure 1a, where the initial tolerance thresholds are set equal in all cases but with different tolerance threshold adjustments. Note that

for Case 1 and Case 2 (to a lesser extent), the tolerance threshold is not stable and less preferable particles are often accepted. Case 3 displays a stable behaviour and the consistent acceptance of preferable particles, which is achieved by setting δ_{Reject} sufficiently smaller than δ_{Accept}. Regarding the threshold-search property, the converged and stable level of the tolerance threshold in Case 3 can be used as an appropriate choice for the fixed tolerance threshold in the ABC-RS algorithm for this specific application.

In Figures 1b to 1d, the convergence of the tolerance threshold is investigated by modifying the tolerance threshold adjustments, while $\epsilon_1 = 1$ for Case 1, $\epsilon_1 = 0.5$ for Case 2, and $\epsilon_1 = 0.0001$ for Case 3. Divergence of the tolerance thresholds are observed in Figure 1b, where $\delta_{\text{Accept}} = \delta_{\text{Reject}} = 0.1$. To enable convergence, one needs to decrease the value of δ_{Reject}. In Figure 1c, convergence is achieved by setting $\delta_{\text{Reject}} = 0.001$, while the tolerance threshold only converges around 150 accepted particles for Case 3. Faster convergence can be achieved for Case 2 and Case 3 when increasing δ_{Accept}, as shown in Figure 1d. The discussions from these figures again highlight the optimal tuning parameter choices discussed earlier.

3. Life distributions

In this section, some life distributions widely used in reliability analysis are discussed, where the probability density function (PDF) and log-likelihood function for each are provided. The likelihood function is required to perform MLE, while it is not required for the ABC-TS algorithm. Suppose that n items are tested and that the life test is terminated when all items have failed. The log-likelihood function for some model $\mathcal{M}$ with parameter vector Θ is then given by

$$\mathscr{L} = \ln\left(\prod_{i=1}^{n} f\left(x_i \,|\Theta, \mathcal{M}\right)\right),$$

where $f(\cdot)$ denotes the PDF and x_i, $i = 1, ..., n$ are the failure times.

Exponential distribution

The PDF of the exponential distribution with parameter λ ($\lambda > 0$) is

$$f_{\text{E}}\left(x \,|\lambda\right) = \lambda \exp\left(-\lambda x\right), \qquad x \geq 0,$$

and the log-likelihood function is given by

$$\mathscr{L}_{\text{E}} = n \ln\left(\lambda\right) - \lambda \sum_{i=1}^{n} x_i.$$

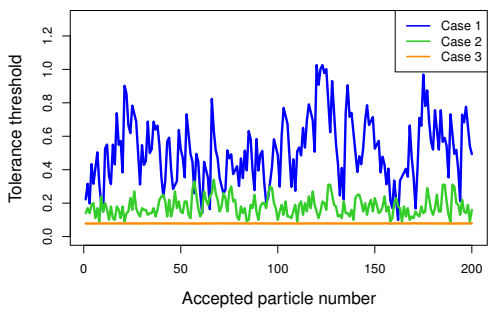
Weibull distribution

The Weibull distribution with scale parameter α and shape parameter β ($\alpha > 0, \beta > 0$) has the PDF

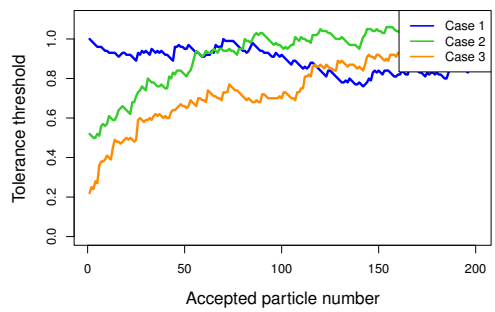
$$f_{\text{W}}\left(x \,|\alpha, \beta\right) = \frac{\beta}{\alpha}\left(\frac{x}{\alpha}\right)^{\beta-1} \exp\left(-\left(\frac{x}{\alpha}\right)^{\beta}\right), \qquad x \geq 0,$$

with the log-likelihood function of the Weibull distribution given by

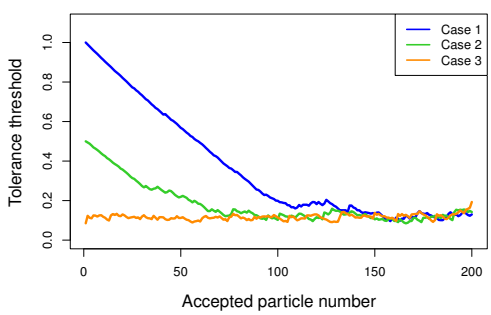
$$\mathscr{L}_{\text{W}} = n \ln\left(\beta\right) - n\beta \ln\left(\alpha\right) + \left(\beta - 1\right) \sum_{i=1}^{n} \ln\left(x_i\right) - \sum_{i=1}^{n} \left(\frac{x_i}{\alpha}\right)^{\beta}.$$



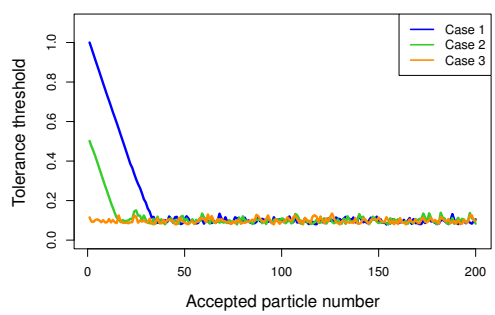
(a) **Threshold instability example.** Case 1: $\epsilon_1 = 0.01$, $\delta_{\text{Accept}} = 0.4346$, $\delta_{\text{Reject}} = 0.0707$. Case 2: $\epsilon_1 = 0.01$, $\delta_{\text{Accept}} = 0.1$, $\delta_{\text{Reject}} = 0.01$. Case 3: $\epsilon_1 = 0.01$, $\delta_{\text{Accept}} = 0.01$, $\delta_{\text{Reject}} = 0.000005$.



(b) **Threshold divergence example.** Case 1: $\epsilon_1 = 1$, $\delta_{\text{Accept}} = 0.01$, $\delta_{\text{Reject}} = 0.01$. Case 2: $\epsilon_1 = 0.5$, $\delta_{\text{Accept}} = 0.01$, $\delta_{\text{Reject}} = 0.01$. Case 3: $\epsilon_1 = 0.0001$, $\delta_{\text{Accept}} = 0.01$, $\delta_{\text{Reject}} = 0.01$.



(c) **Slow threshold convergence example.** Case 1: $\epsilon_1 = 1$, $\delta_{\text{Accept}} = 0.01$, $\delta_{\text{Reject}} = 0.001$. Case 2: $\epsilon_1 = 0.5$, $\delta_{\text{Accept}} = 0.01$, $\delta_{\text{Reject}} = 0.001$. Case 3: $\epsilon_1 = 0.0001$, $\delta_{\text{Accept}} = 0.01$, $\delta_{\text{Reject}} = 0.001$.



(d) **Fast threshold convergence example.** Case 1: $\epsilon_1 = 1$, $\delta_{\text{Accept}} = 0.03$, $\delta_{\text{Reject}} = 0.001$. Case 2: $\epsilon_1 = 0.5$, $\delta_{\text{Accept}} = 0.03$, $\delta_{\text{Reject}} = 0.001$. Case 3: $\epsilon_1 = 0.0001$, $\delta_{\text{Accept}} = 0.03$, $\delta_{\text{Reject}} = 0.001$.

Figure 1. Tolerance threshold convergence examples for various hyperparameter settings.

Birnbaum-Saunders distribution

The PDF and log-likelihood function of the Birnbaum-Saunders distribution with shape parameter α and scale parameter β ($\alpha > 0, \beta > 0$) are, respectively, given by

$$f_{\text{BS}}(x\,|\alpha,\beta) = \frac{1}{2\sqrt{2\pi}\alpha\beta}\left(\left(\frac{\beta}{x}\right)^{1/2} + \left(\frac{\beta}{x}\right)^{3/2}\right)\exp\left(-\frac{1}{2\alpha^2}\left(\frac{x}{\beta} + \frac{\beta}{x} - 2\right)\right), \qquad x > 0,$$

and

$$\mathscr{L}_{\text{BS}} = -n\ln\left(2\sqrt{2\pi}\alpha\beta\right) + \sum_{i=1}^{n}\ln\left(\left(\frac{\beta}{x_i}\right)^{1/2} + \left(\frac{\beta}{x_i}\right)^{3/2}\right) - \frac{1}{2\alpha^2}\sum_{i=1}^{n}\left(\frac{x}{\beta} + \frac{\beta}{x} - 2\right).$$

Gamma distribution

The PDF of the gamma distribution with shape parameter α and scale parameter β ($\alpha > 0, \beta > 0$) is given by

$$f_{\text{G}}(x\,|\alpha,\beta) = \frac{1}{\Gamma(\alpha)\,\beta^{\alpha}}x^{\alpha-1}\exp\left(-\frac{x}{\beta}\right), \qquad x > 0,$$

where the log-likelihood function is defined as

$$\mathscr{L}_{\text{G}} = -n\alpha\ln(\beta) - n\ln(\Gamma(\alpha)) + (\alpha - 1)\sum_{i=1}^{n}\ln(x_i) - \sum_{i=1}^{n}\frac{x_i}{\beta}.$$

Log-normal distribution

The PDF of the log-normal distribution with parameters μ and σ ($\mu \in \mathbb{R}, \sigma > 0$) is given by

$$f_{\text{LN}}(x\,|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma x}\exp\left(-\frac{1}{2\sigma^2}(\ln(x) - \mu)^2\right), \qquad x > 0.$$

The log-likelihood function of the log-normal distribution is given by

$$\mathscr{L}_{\text{LN}} = -n\ln\left(\sqrt{2\pi}\sigma\right) - \sum_{i=1}^{n}\ln(x_i) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(\ln(x_i) - \mu)^2.$$

4. Simulation results

In this simulation study, parameter estimation is considered for the five distributions defined in Section 3. Arbitrary true parameter values are selected for each distribution, given in Table 1, where these parameters are estimated using MLE and the ABC-TS algorithm. Sample sizes of $n = \{10, 20, 30, 50, 100, 200\}$ are considered and $M = 100$ Monte Carlo iterations are performed for each sample size. For each Monte Carlo iteration and life distribution, a sample is generated from the specific life distributions and then used to perform parameter estimation. The limited-memory Broyden-Fletcher-Goldfarb-Shanno with box constraints (L-BFGS-B) algorithm is used to perform MLE, since this algorithm allows for the specification of bound constraints on parameters (see, Byrd et al., 1995).

For the ABC-TS algorithm, the initial tolerance threshold is set to $\epsilon_1 = 0.01$ to prevent the acceptance of undesirable candidate particles due to an unnecessary large value for ϵ_1. The number

of accepted particles is set to $N = 200$ and the three distance functions defined in Section 2 are investigated. Considering the optimal tuning parameter choices and allowing for some relaxation to decrease the computational burden, the tolerance threshold adjustment values are chosen as $\delta_{\text{Accept}} = 0.01$ and $\delta_{\text{Reject}} = 0.000005$.

For each parameter of the life distributions under consideration, independent uniform priors are constructed around some initial parameter estimates. These uniform priors are chosen wide enough to allow for the exploration of a range of acceptable values for the distributional parameters. Given an initial parameter estimate $\widetilde{\theta}$ for a parameter θ of some life distribution, the uniform prior on the interval $[\widetilde{\theta}/3\ ;\ \widetilde{\theta} \times 3]$ is imposed on the parameter θ. Using the ABC-TS algorithm to generate an approximate marginal posterior for a parameter θ, the ABC estimate of θ under a squared error loss is given by the posterior mean, i.e.,

$$\hat{\theta}_{\text{ABC}} = \frac{1}{N} \sum_{i=1}^{N} \theta^{(i)},$$

which is the Monte Carlo average of the accepted particle values for the parameter θ.

To compare the performance of MLE and the ABC-TS algorithm, the relative root mean squared error (RRMSE) and relative absolute bias (RAB) for each parameter, given the respective life distributions, is calculated over the 100 Monte Carlo iterations. The RRMSE and RAB for some estimator $\hat{\theta}$ of the parameter θ are, respectively, computed as

$$\text{RRMSE}\left(\hat{\theta}\right) = \frac{1}{|\theta|} \sqrt{\frac{1}{M} \sum_{i=1}^{M} \left(\hat{\theta}_i - \theta\right)^2}$$

and

$$\text{RAB}\left(\hat{\theta}\right) = \frac{\left|\bar{\theta} - \theta\right|}{|\theta|},$$

where $\bar{\theta} = M^{-1} \sum_{i=1}^{M} \hat{\theta}_i$.

The results for the different life distributions are given in Tables 2 to 6. For each simulation setting in the tables, the lowest RRMSE and RAB are highlighted. The following general observations can be made from the results provided:

- While MLE performs the best in terms of RRMSE in most cases, the ABC-TS algorithm, specifically using the AD distance function, has comparable performance.
- In cases where parameter estimation via MLE is more complicated, such as for the Weibull and gamma distributions, the ABC-TS often outperforms MLE.
- The ABC-TS algorithm outperforms MLE in terms of RAB, although this is not consistent in terms of the distance function used.
- For the ABC-TS algorithm, the AD distance function has the best overall performance when considering both RRMSE and RAB.

Table 1. Arbitrary parameter values for the simulation study.

Exponential	Weibull	Birnbaum-Saunders	Gamma	Log-normal
$\lambda = 0.05$	$\alpha = 60, \beta = 2$	$\alpha = 0.5, \beta = 25$	$\alpha = 2, \beta = 15$	$\mu = 3.5, \sigma = 0.5$

Table 2. RRMSE and RAB for the exponential distribution.

		Metric results							
		RRMSE				RAB			
n	Parameter	MLE	KS	CvM	AD	MLE	KS	CvM	AD
10	λ	**0.3362**	0.3754	0.3888	0.3402	0.0655	0.0510	0.0651	**0.0325**
20	λ	**0.1965**	0.2121	0.2134	0.1998	0.0102	0.0049	**0.0009**	0.0084
30	λ	**0.1722**	0.2080	0.2070	0.1801	0.0327	0.0177	0.0188	**0.0097**
50	λ	**0.1505**	0.1962	0.1604	0.1539	0.0046	0.0115	0.0036	**0.0008**
100	λ	**0.1059**	0.1283	0.1248	0.1166	**0.0009**	0.0019	0.0044	0.0059
200	λ	**0.0686**	0.0790	0.0745	0.0720	0.0064	**0.0022**	0.0048	0.0025

Table 3. RRMSE and RAB for the Weibull distribution.

		Metrics results							
		RRMSE				RAB			
n	Parameter	MLE	KS	CvM	AD	MLE	KS	CvM	AD
10	α	**0.1580**	0.1697	0.1696	0.1634	**0.0019**	0.0064	0.0070	0.0150
	β	0.3421	0.3980	0.4793	**0.2967**	0.1190	0.1192	0.1477	**0.0054**
20	α	**0.1233**	0.1333	0.1343	0.1286	**0.0065**	0.0086	0.0087	0.0136
	β	0.2199	0.2572	0.2623	**0.2040**	0.0924	0.0863	0.0980	**0.0363**
30	α	**0.0931**	0.0995	0.0987	0.0948	0.0018	**0.0015**	0.0022	0.0050
	β	**0.1637**	0.2335	0.2320	0.1711	0.0387	0.0634	0.0596	**0.0137**
50	α	0.0684	0.0685	0.0689	**0.0683**	0.0117	**0.0092**	0.0105	**0.0092**
	β	**0.1366**	0.1811	0.1759	0.1387	0.0559	0.0552	0.0585	**0.0338**
100	α	0.0581	0.0583	0.0580	**0.0573**	**0.0002**	**0.0002**	0.0003	0.0011
	β	**0.0919**	0.1142	0.1113	0.0969	0.0226	0.0209	0.0201	**0.0085**
200	α	0.0382	0.0378	0.0378	**0.0373**	0.0029	**0.0019**	0.0021	**0.0019**
	β	**0.0579**	0.0736	0.0726	0.0642	0.0016	0.0017	**0.0004**	0.0038

Table 4. RRMSE and RAB for the Birnbaum-Saunders distribution.

		Metrics results							
		RRMSE				RAB			
n	Parameter	MLE	KS	CvM	AD	MLE	KS	CvM	AD
10	α	**0.2357**	0.2858	0.2985	0.2394	0.1205	0.1152	0.1242	**0.0340**
	β	0.1409	0.1421	0.1390	**0.1385**	0.0025	0.0042	0.0017	**0.0015**
20	α	**0.1655**	0.2247	0.2238	0.1830	0.0403	0.0378	0.0433	**0.0069**
	β	**0.1161**	0.1310	0.1299	0.1222	**0.0188**	0.0259	0.0263	0.0223
30	α	**0.1382**	0.1727	0.1695	0.1506	0.0291	0.0202	0.0306	**0.0004**
	β	**0.0791**	0.0844	0.0841	0.0814	**0.0022**	0.0032	0.0052	0.0037
50	α	**0.0986**	0.1161	0.1178	0.1022	0.0277	0.0184	0.0249	**0.0056**
	β	**0.0684**	0.0757	0.0754	0.0723	0.0141	**0.0114**	0.0124	0.0128
100	α	**0.0719**	0.0897	0.0898	0.0782	**0.0002**	0.0066	0.0050	0.0138
	β	**0.0541**	0.0560	0.0556	0.0545	**0.0004**	0.0030	0.0024	0.0012
200	α	**0.0491**	0.0568	0.0554	0.0498	0.0041	0.0006	**0.0004**	0.0034
	β	**0.0336**	0.0360	0.0362	0.0347	0.0026	**0.0020**	0.0029	0.0027

Table 5. RRMSE and RAB for the gamma distribution.

		Metric results							
		RRMSE				RAB			
n	Parameter	MLE	KS	CvM	AD	MLE	KS	CvM	AD
10	α	0.7146	0.9499	1.0477	**0.5784**	0.3514	0.4461	0.4698	**0.1636**
	β	**0.4257**	0.5863	0.5801	0.5464	0.1169	0.0585	**0.0505**	0.0777
20	α	0.3668	0.5306	0.5210	**0.3220**	0.1263	0.1960	0.1830	**0.0488**
	β	**0.3585**	0.4655	0.4498	0.4151	0.0387	0.0193	**0.0114**	0.0579
30	α	0.2990	0.3923	0.3790	**0.2809**	0.1057	0.1245	0.1308	**0.0434**
	β	**0.2818**	0.3268	0.3390	0.3142	0.0441	**0.0207**	0.0256	0.0347
50	α	0.2187	0.2184	0.2201	**0.1988**	0.0543	0.0293	0.0352	**0.0118**
	β	**0.2117**	0.2517	0.2427	0.2149	0.0334	0.0096	**0.0015**	0.0127
100	α	**0.1364**	0.1946	0.1867	0.1488	0.0153	0.0253	0.0254	**0.0028**
	β	**0.1576**	0.2122	0.2010	0.1762	**0.0056**	0.0187	0.0133	0.0267
200	α	**0.0919**	0.1177	0.1117	0.0945	**0.0038**	0.0156	0.0051	0.0054
	β	**0.0969**	0.1314	0.1272	0.1079	0.0076	**0.0040**	0.0124	0.0196

Table 6. RRMSE and RAB for the log-normal distribution.

		Metric results							
		RRMSE				RAB			
n	Parameter	MLE	KS	CvM	AD	MLE	KS	CvM	AD
10	μ	**0.0460**	0.0498	0.0494	0.0472	**0.0054**	0.0064	0.0072	0.0060
	σ	**0.2020**	0.2595	0.2641	0.2219	0.0777	0.0453	0.0546	**0.0277**
20	μ	**0.0295**	0.0314	0.0315	0.0302	0.0013	**0.0002**	0.0009	0.0011
	σ	**0.1550**	0.1998	0.2006	0.1655	0.0562	0.0587	0.0640	**0.0126**
30	μ	**0.0300**	0.0320	0.0312	0.0304	0.0018	**0.0016**	**0.0016**	0.0017
	σ	**0.1270**	0.1702	0.1624	0.1420	0.0224	0.0138	**0.0037**	0.0218
50	μ	**0.0232**	0.0241	0.0239	0.0236	0.0014	0.0014	**0.0013**	0.0014
	σ	**0.1036**	0.1255	0.1222	0.1087	0.0179	0.0075	**0.0007**	0.0119
100	μ	**0.0154**	0.0156	0.0156	0.0155	0.0006	0.0003	**0.0001**	0.0003
	σ	**0.0803**	0.0906	0.0921	0.0831	0.0177	0.0038	0.0076	**0.0012**
200	μ	0.0104	0.0106	0.0106	**0.0103**	0.0012	**0.0008**	**0.0008**	0.0010
	σ	**0.0492**	0.0635	0.0603	0.0535	**0.0034**	0.0055	0.0006	0.0054

5. Conclusion

The ABC-TS algorithm introduced in this paper is a modification of the standard ABC-RS algorithm, where the tolerance threshold is adaptively adjusted as candidate particles are accepted or rejected. The threshold-search property of the ABC-TS algorithm is illustrated, which can be used to determine a suitable fixed tolerance threshold for the ABC-RS algorithm. Optimal choices for the hyperparameters of the ABC-TS algorithm are also discussed.

A simulation study on distributional parameter estimation is performed, where the results indicate that the ABC-TS algorithm has comparable performance to that of MLE. The ABC-TS algorithm often outperforms MLE in cases where there are no closed-form solutions for the maximum likelihood estimators. This warrants further investigation into the performance of the ABC-TS algorithm, specifically for parameter estimation in cases where the likelihood function becomes challenging to work with. For example, in accelerated life testing models, distribution parameters are expanded via a time transformation function, resulting complex likelihood functions. For these models, approximation techniques are often required to perform parameter estimation in both the frequentist and Bayesian paradigms.

References

Abdessalem, A. B., Dervilis, N., Wagg, D., and Worden, K. (2019). Model selection and parameter estimation of dynamic systems using a novel variant of approximate Bayesian computation. *Mechanical Systems and Signal Processing*, **122**, 364–386.

Beaumont, M. A., Zhang, W., and Balding, D. J. (2002). Approximate Bayesian computation in population genetics. *Genetics*, **162** (4), 2025–2035.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for

statisticians. *Journal of the American Statistical Association*, **112** (518), 859–877.

Brooks, S., Gelman, A., Jones, G., and Meng, X. (Editors) (2011). *Handbook of Markov Chain Monte Carlo*. Chapman and Hall, New York, NY.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal of Scientific Computing*, **16** (5), 1190–1208.

Lintusaari, J., Gutmann, M. U., Dutta, R., Kaski, S., and Corander, J. (2017). Fundamentals and recent developments in approximate Bayesian computation. *Systematic Biology*, **66** (1), e66–e82.

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., and Feldman, M. W. (1999). Population growth of human Y chromosomes: A study of Y chromosome microsatellites. *Molecular Biology and Evolution*, **16** (12), 1791–1798.

Sisson, S. A., Fan, Y., and Beaumont, M. A. (Editors) (2019). *Handbook of Approximate Bayesian Computation*. CRC Press, Boca Raton, FL.

Stephens, M. A. (1974). EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, **69** (347), 730–737.

Tavaré, S., Balding, D. J., Griffiths, R. C., and Donnelly, P. (1997). Inferring coalescence times from DNA sequence data. *Genetics*, **145**, 505–518.

Submitted to the Proceedings of the 65th Annual Conference of the South African Statistical Association

An improved similarity test for comparing spatial point patterns

René Stander[1], *Inger Fabris-Rotelli*[1], *Gregory Breetzke*[2] *and Jean-Pierre Stander*[1]

[1]Department of Statistics, University of Pretoria
[2]Department of Geography, Geoinformatics and Meteorology, University of Pretoria

In this paper we re-examine the similarity threshold of Andresen's S-Index for spatial point patterns. Andresen's S-Index is used widely by geographers, specifically in criminology literature, to determine the similarity between two spatial point patterns. A spatial point pattern consists of the locations where an event of interest occurred. The S-Index represents the proportion of spatial units that have similar spatial patterns in both point patterns and ranges from 0 to 1. The test is subjective in that it delineates spatial similarity and dissimilarity at a threshold of $S = 0.8$ without statistical motivation. We propose a technique to remove this subjectivity by considering the second-order nature of the spatial data. An improved, more robust test is thus set up providing more informative thresholds for the similarity test for the second-order nature of the point pattern as well as the chosen grid size. This approach is applied to road networks in the city centres of Pretoria and Johannesburg, South Africa. The road network is represented as a point pattern, and the similarity of the road structure is determined with the new methodology.

Keywords: Point pattern, Road network, S-Index, Similarity test.

1. Introduction

To determine whether two spatial data sets originate from the same spatial process, spatial similarity tests are used (Borrajo et al., 2020). Spatial data are of three main types, namely point patterns, lattice data and geostatistical data (Cressie, 2015). In this paper, the focus is only on the spatial similarity between spatial point patterns, for which the point location is modelled.

Only a handful of spatial similarity tests have been developed that can be divided into distance-based and area-based methods. Recent spatial similarity tests are developed by Alba-Fernández et al. (2016), Feuntes-Santos et al. (2017), Wheeler et al. (2018) and Kirsten and Fabris-Rotelli (2021). However, a commonly used method of statistically comparing point patterns is the spatial point pattern test developed by Andresen (2009). This area-based test is used to compare the similarity between two different spatial point patterns over the same domain. The final result of this test is the S-Index which represents the proportion of spatial units that have similar spatial patterns for both point patterns, ranging from zero to one. One of the main appeals of Andresen's test is that the output can be mapped, showing the user where local differences are present in the point patterns.

Corresponding author: René Stander (rene.stander@up.ac.za)
MSC2020 subject classifications: 62G10, 62M30, 62P25

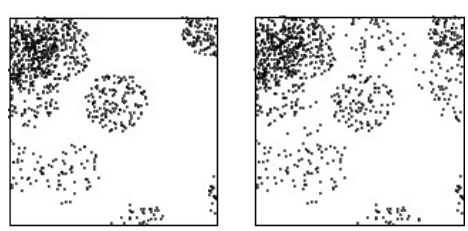
Unsurprisingly, Andresen's spatial point pattern test has been extensively used over the past decade mainly in the geographic analysis of crime with topics ranging from crime seasonality and its variation across space (Andresen and Malleson, 2013; Linning, 2015), to spatial crime displacement analysis (Andresen and Malleson, 2014; Vandeviver and Steenbeek, 2019) to studies examining the stability of crime patterns across various levels of aggregation (Andresen and Malleson, 2011). Moreover, the test has been used to examine crime concentrations in a variety of countries including Brazil (de Melo et al., 2015; Pereira et al., 2017), the Netherlands (Vandeviver and Steenbeek, 2019), Canada (Andresen et al., 2017), New Zealand (Breetske and Andresen, 2018), South Africa (Schutte and Breetske, 2018), and the United States (Wheeler et al., 2018).

Despite its relative success as a method, a number of limitations have been identified. These include the fact that the selection of the base and the test data set is arbitrary with the value of the *S*-Index being, to some extent, dependent on the choice made by the user. Another limitation is the fact that areas with no points in them in the test data set will always have a confidence interval of 0-0% (Wang, 2013). Another issue that has received very little attention in the extant literature is the arbitrariness of the threshold values delineated by Andresen (2016) as signifying whether the resultant *S*-Index value (ranging from 0 to 1) signifies whether the two point patterns are similar or dissimilar. According to Andresen and Linning (2012) the 'rule of thumb' is that an *S*-Index value of 0.8 indicates that the two point patterns being compared are similar. The similarity threshold value in particular appears to be arbitrarily based on prior threshold values identified for variance inflation factors (O'Brien, 2007) and correlation coefficients (Cohen, 1988) rather than on empirical proof.

Figure 1 shows two point patterns that are 90% identical, i.e. 90% of their points are in exactly the same location. When these point patterns are compared using the spatial point pattern test by Andresen (2009), the *S*-Index is 0.7, meaning they would fail a similarity test with the current threshold value of 0.8, even though they are 90% identical.

The aim of this research is to remove much of the subjectivity involved in identifying whether two spatial point patterns are similar or dissimilar as determined by Andresen's *S*-Index. Clearer similarity thresholds are proposed based on the second-order nature of the data, namely how the two original point patterns are distributed (regular or clustered). We run a series of simulations by simulating different spatial point patterns and calculating the re-evaluated thresholds.

In Section 2, Andresen's spatial point pattern test is explained in detail as well as the proposed improvement of the thresholds. A simulation study is conducted in Section 3 to determine the re-evaluated thresholds. Andresen's spatial point pattern test is applied to road networks in the city



(a) (b)

Figure 1. Illustration of spatial point patterns being 90% similar and yielding an *S*-Index of 0.7 when the spatial point pattern test by Andresen (2009) is applied.

centres of Pretoria and Johannesburg in Section 4. This is followed by a more detailed discussion in Section 5 and Section 6 concludes.

2. Methodology

A spatial point pattern is a realisation from a stochastic mechanism called a spatial point process (Baddeley et al., 2015). A spatial point pattern consists of the locations of a certain event denoted by coordinates. The arrangement of the points in a spatial point pattern can be classified into three groups as shown in Figure 2 (Baddeley et al., 2015). Figure 2(a) is an example of a regular spatial point pattern where the points within the spatial point pattern repel each other and are spaced throughout the domain. A completely spatially random point pattern is given in Figure 2(b) which is a point pattern where the points are independently distributed from a Poisson point process and form no distinct pattern. Lastly, a clustered point pattern is shown in Figure 2(c) which is a point pattern where some of the points tend to attract each other to form groups at certain locations within the point pattern.

2.1 Andresen's S-Index

Consider two spatial point patterns X_j, $j = 1, 2$, observed on a spatial domain. Let each spatial point pattern consist of n_j number of points, so that the patterns are denoted by

$$X_j = \{p_{1j}, p_{2j}, \dots p_{n_j j}\},$$

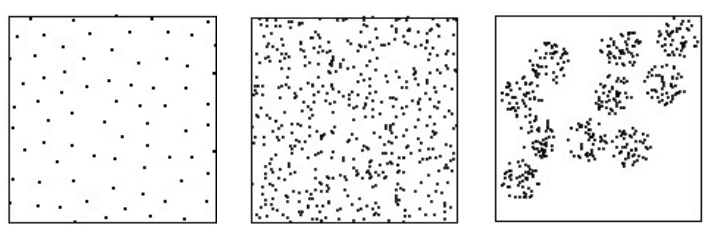
where p_{kj} is the location of the k^{th} point in pattern X_j.

The spatial domain of X_j, $j = 1, 2$ is divided into regions, A_i, $i = 1, 2, \dots, m$. These regions can either be regularly (grid-like) or irregularly (for example, administration boundaries) shaped. In the absence of pre-defined areas within the spatial domain, a regular grid is the popular choice.

The spatial point pattern similarity test, proposed by Andresen (2009), is performed on X_1 (base dataset) and X_2 (test dataset) by the following algorithm:

1. Using X_1, calculate the proportion of points within each $A_i, i = 1, 2, \dots, m$:

$$t_i = \frac{\sum_{k=1}^{n_1} I(p_{k1} \in A_i)}{n_1}.$$



(a) Regular (b) CSR (c) Clustered

Figure 2. Illustration of how the points within a spatial point pattern can be arranged. (a) Regular spatial point pattern where the points are fairly evenly spaced. (b) Completely spatially random point pattern where the points from no particular pattern. (c) Clustered point pattern where the points form groups at different locations in the point pattern.

2. Repeat the following 200 times:

   2.1. Using X_2, sample 85% of the points randomly. Denote the r^{th} sample as $B_r = \{q_{1r}, q_{2r}, \ldots, q_{n_b r}\}$ where $n_b = 0.85 \times n_2$ and q_{kr} is the location of the k^{th} sampled point for the r^{th} sample.

   2.2. Calculate the proportion of points of B_r in each A_i:

$$b_{ir} = \frac{\sum_{k=1}^{n_b} I(q_{kr} \in A_i)}{n_b}.$$

Step 2 is repeated 200 times for the sake of being conservative (Andresen, 2009). A sample of 85% is taken following the research done by Ratcliffe (2004) that if a random sample is taken on a spatial point pattern, the spatial structure will be preserved if at least 85% of the points in the spatial point pattern are sampled.

3. Let $\mathbf{b_i} = (b_{i1}, b_{i2}, \ldots, b_{i200})'$ be the vector of all the percentages of points in region A_i. Let c_i be the non-parametric confidence interval for region A_i by taking the 2.5^{th} and the 97.5^{th} percentiles of $\mathbf{b_i}$ as the lower and the upper limits respectively.

4. Determine the local similarities for each A_i:

$$s_i = I(t_i \in c_i).$$

Thus if the proportion of points within each A_i for pattern X_1 is contained in the confidence interval, c_i, calculated in step 3, then the patterns are said to exhibit a similar proportion of points in A_i (i.e. $s_i = 1$). Otherwise they are significantly different (i.e. $s_i = 0$).

4. The S-Index is a global similarity parameter that ranges from 0 (no similarity) to 1 (perfect similarity) and is calculated as

$$S = \frac{\sum_{i=1}^{m} s_i}{m},$$

   where m is the number of spatial regions considered within the domain.

The test is not concerned with the statistical distribution of the points in the spatial point pattern but only if the points in the different patterns are similarly located; this makes it a non-parametric test.

2.2 Improved Similarity Test

In this study, we aim to provide clearer, empirically-based cut-off values as the threshold for the similarity value S. We propose a threshold based on the second-order nature of the data; namely, how strongly the data is clustered. This addressed using the Index of Clumping (ICS) (David and Moore, 1954) which provides an indication of the degree of clustering in the spatial pattern. The index of clumping is calculated as

$$ICS = \frac{s^2}{\bar{x}} - 1,$$

where $\bar{x}$ and s^2 are the sample mean and variance of the number of points in each grid cell. If $ICS < 1$, the point pattern is classified as regular and if $ICS > 1$, the point pattern is clustered.

A series of simulations are performed to propose these new similarity thresholds for the S-Index in a simulation study. Pairs of spatial point patterns will be simulated to be either 80% or 90% identical. That is, 80% or 90% of the points in the two spatial point patterns are located in exactly the same location. Andresen's S-Index will then be applied to these simulated pairs.

We create new similarity thresholds for Andresen's S-Index calculated as the lower outlier limit of the S-Index values generated from the simulation study since outliers in the data comes from a different distribution than the rest of the data (Schwertman and de Silva, 2007). The inner fences method to identify outliers will be employed to provide the threshold values. The lower limit is given by

$$f_1 = Q_1 - 1.5(Q_3 - Q_1).$$

In cases where there are no outliers for those S-Index values, the lower outlier limit will be below all the S-Index values. In such circumstances, the new similarity threshold will be the minimum of all the S-Index values. Accordingly, the following equation will be used to calculate the new threshold value,

$$f = \max\{\min(S), f_1\}. \tag{1}$$

3. Simulation Study

The aim of the simulation study is to calculate and propose new similarity thresholds for Andresen's S-Index, based on the second-order nature of the point pattern. The second-order nature of point patterns is classified as either regular, clustered or completely spatially random (CSR). A regular point pattern has points with an inhibition distance between them, a clustered point pattern exhibits points closer together than expected, and a CSR point pattern indicates points are randomly placed without any spatial dependency.

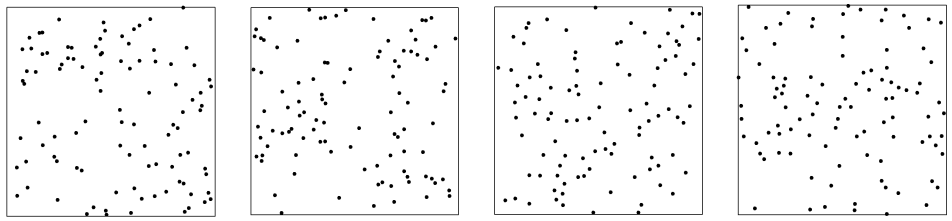
3.1 Simulation Design

Regular and clustered patterns are simulated, over two different windows with an area of 100 square units - a rectangular window and a convex hull polygonal window (see Figure 4). These windows are commonly used in the absence of lattice data. The number of points are varied as small (±100), medium (±500) and large (±1000).

In the simulation study, two types of point pattern simulations are considered as simulated as follows:

1. Regular point patterns: The `rSSI` function in `R` (Baddeley et al., 2015) was used with four different inhibition distances ($r = 0.1, 0.15, 0.2, 0.3$) to simulate regular point patterns. Examples of regular point pattern simulations for the different inhibition distances are shown in Figure 5(a)-(d).

2. Clustered point patterns: The `rMatClust` function in `R` (Baddeley et al., 2015) was used by varying the parameter values of `kappa` (number of clusters divided by 100), `scale` $(1, 1.5, 2)$ and `mu` (size of the point pattern divided by the number of clusters). The number of clusters in each simulation has been randomly chosen at each iteration as between 8 and 12. Examples of



(a) $r = 0.1$ (b) $r = 0.15$ (c) $r = 0.2$ (d) $r = 0.3$

(a) $\lambda_1(s) = 5(x^2 + y^2)$ (b) $\lambda_2(s) = 13(x + y)$ (c) $\lambda_3(s) = 25y$

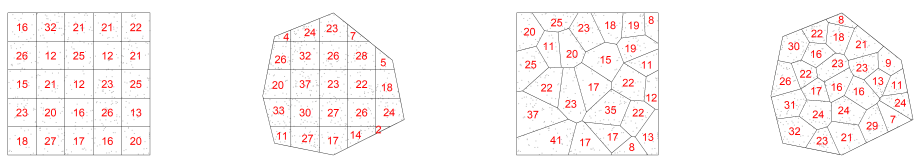
Figure 3. Illustrations of the non-constant intensity functions in the simulation study.

clustered point pattern simulations for the different values of the `scale` parameter are shown in Figure 5(e)-(g).

To create the other pattern to use in the comparison that is either 80% or 90% identical, 20% or 10% of the points in the point pattern are replaced with other random points simulated using a Poisson point process with the `rpoispp` function in R.

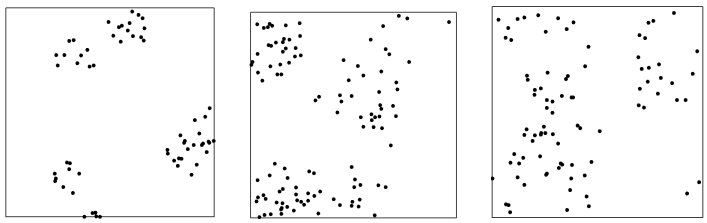
The point patterns are simulated at four different intensities, one constant and the other three non-constant. The three different functions of the non-constant intensities are shown in Figure 3. Inhomogeneity of the clustered spatial point patterns was incorporated with the `kappa` parameter.

Andresen's spatial point pattern test was applied to all pairs of simulations. The point patterns were divided into different areas, and then the S-Index was calculated. As the S-Index utilises pre-defined regions within the domain for comparison of the two spatial point patterns, it is important that the two point patterns are divided in the same manner. We use grids of different sizes to divide the patterns in order to see what influence this has on the consistency of the test result. Regular grids



(a) (b) (c) (d)

Figure 4. Examples of how a 5×5 grid appears on the (a) rectangular window and the (b) convex polygonal window. Examples of a (c) rectangular window and a (d) convex polygonal window with 25 irregular Voronoi cells.



(e) scale = 1 (f) scale = 1.5 (g) scale = 2

Figure 5. Examples of some of the simulations considered. (a), (b), (c), (d) are examples of the simulated regular point patterns for the different inhibition distances considered. (e), (f), (g) are examples of the simulated clustered point patterns for different `scale` parameter values.

as well as irregular grids are considered. Figure 4 shows an example of the areas overlaid on the two different windows considered. Figure 4(a) is an example of a regular 5×5 grid on a rectangular window and Figure 4(b) is an example of a regular 5×5 grid on a convex polygonal window. Figure 4(c) and Figure 4(d) are examples of an irregular grid used on the rectangular window and the convex polygonal window, respectively. Irregular grids were simulated as Voronoi cells. In this simulation study, the following grids and resolutions are considered:

- Regular grid: 10×10, 15×15, 20×20.
- Irregular grid: 25 areas, 100 areas, 200 areas.

A regular as well as irregular grid are considered as in some applications administrative boundaries will be available and these might be preferable to use instead of a regular grid. This type of division will also likely consist of fewer areas compared to the regular grid, hence, the lower resolution considered in the simulation study.

In practice, when comparing the similarity of two point patterns, a user does not know their degree of similarity (if this were known, the test would not be needed in the first place). Final thresholds are thus calculated using bootstrap sampling. Simple random samples from the S-Index values were taken 999 times for the 80% and 90% similar patterns together. Threshold values were then calculated for each sample using Equation (1) and their rounded mean used as the final threshold.

For all the simulations, a division is made into two groups according to the ICS values. If $ICS < 1$, the point patterns are classified as regular and if $ICS > 1$, the point patterns are classified as clustered.

Figure 6 shows the ICS values for spatial point patterns X_1 and X_2 used in the simulation study. From this figure it can be seen that the spread for the regular patterns (with ICS values less than 1) is less than the spread for the clustered patterns (with ICS values greater than 1). In some cases, the point patterns will be simulated as either regular or clustered but according to the ICS value will be classified as the other. When calculating the thresholds, the ICS values are purely used in the classification as in real applications it is the measurement used to distinguish quantitatively between regular and clustered patterns.

3.2 Simulation Results

Figure 7 shows the final S-Index values for the simulation. Each density curve represents the distribution of the S-Index values for each grid type and resolution. The blue curves are the

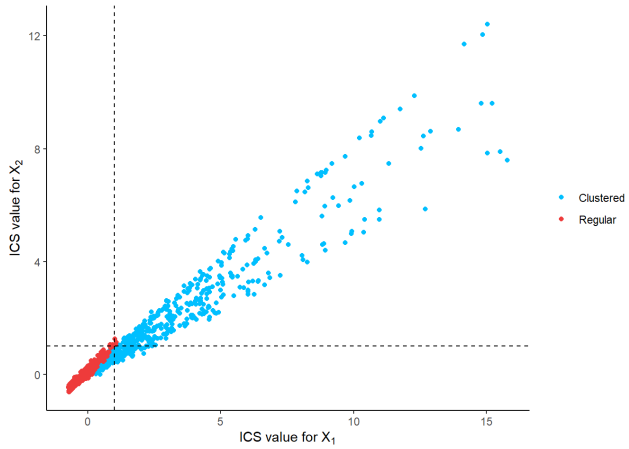
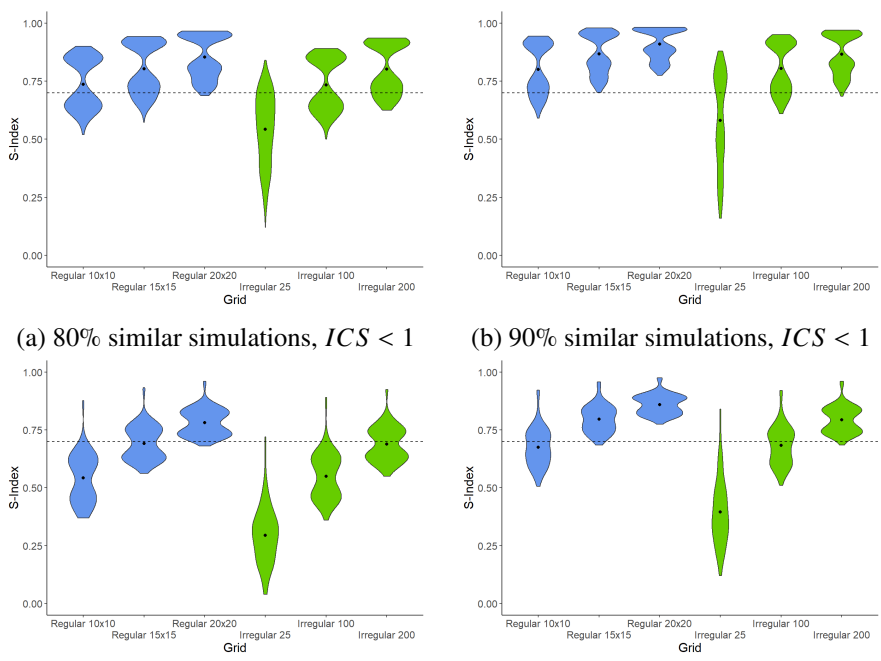


Figure 6. Plot of the ICS values for X_1 and X_2 in the simulation study. The black dotted lines indicate the cut-off point of one.



(a) 80% similar simulations, $ICS < 1$ (b) 90% similar simulations, $ICS < 1$

(c) 80% similar simulations, $ICS > 1$ (d) 90% similar simulations, $ICS > 1$

Figure 7. Visual representation of the results from the simulation study. The dotted lines indicate the similarity percentage at which the pairs of simulations are simulated at.

Table 1. Re-evaluated similarity thresholds for each grid size and ICS category.

		$ICS < 1$	$ICS > 1$
Regular	10×10	0.436 ± 0.012	0.400 ± 0.035
	15×15	0.561 ± 0.009	0.574 ± 0.026
	20×20	0.675 ± 0.007	0.693 ± 0.021
Irregular	25	0.134 ± 0.028	0.053 ± 0.036
	100	0.444 ± 0.014	0.382 ± 0.030
	200	0.556 ± 0.010	0.557 ± 0.032

distributions of the regular grid, while the green curves indicate the distributions of the irregular grid. Figures 7(a) and 7(b) are the simulation results of the spatial similarity test applied between two regular point patterns, where the first are patterns simulated to be 80% similar and the latter 90% similar. Similarly, Figures 7(c) and 7(d) are the simulation results of the spatial similarity test applied to two clustered point patterns.

As can be seen from Figure 7 the finer the resolution of the grid, the more accurate the results of the proposed spatial similarity test. A finer resolution grid results in more stable results in the case of clustered point patterns compared to the regular point patterns. When making use of irregular grids, a higher resolution results in better performance in estimating the similarity between the point patterns. Even more so in the clustered point patterns compared to the regular point patterns. In general, the regular grids do perform better than the irregular grids.

In Table 1 the results of the thresholds derived from the simulation study are shown. These thresholds were obtained using bootstrapping methods as discussed above.

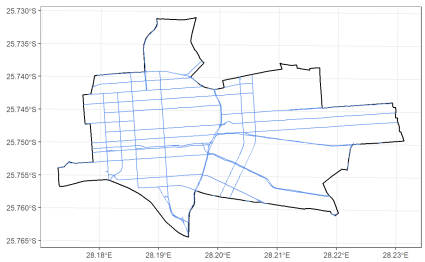
Different thresholds are derived from the different grid types and sizes as well as the different types of point patterns. The finer the grid, the higher the threshold is with a smaller standard deviation. This supports that a higher resolution results in more stable and accurate results.

4. Application

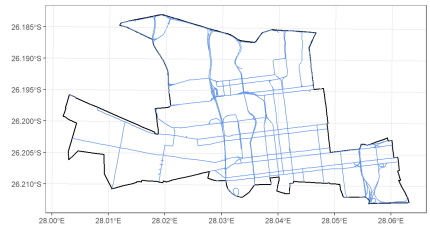
This similarity test is applied to road networks in Pretoria and Johannesburg city centres in South Africa to determine whether the road structures are analogous. The road networks are given in Figures 8(a) and 8(b) respectively. The road networks can be represented as a spatial point pattern by placing equal numbers of equidistant points along each linear segment of the network (see Figures 8(c) and 8(d)). When the road networks are represented as such, the spatial point pattern similarity test can be applied to assess the similarity between the patterns.

In most cases, the similarity of spatial point patterns is assessed considering the data observed over the same domain. However, in this application, different domains necessitate a slightly different approach. This situation will often arise in real examples. When applying the spatial similarity test, the union of the two domains are considered before dividing the region into a grid. This is to ensure that similarity is tested over a domain that covers both regions. This is illustrated in Figure 9(a).

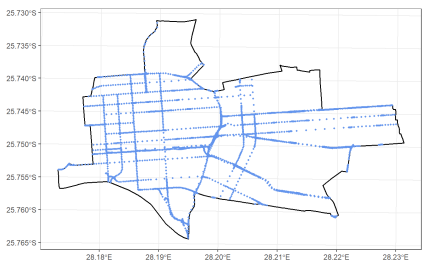
Because the two road networks are located in different domains, the angle at which they should be overlayed to obtain the best estimate of similarity should also be carefully considered. An objective solution is to consider the similarity of the domains at a number of different rotations. One of the



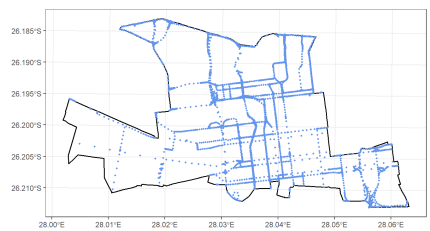
(a) Pretoria roads



(b) Johannesburg roads



(c) Pretoria roads represented as points



(d) Johannesburg roads represented as points

Figure 8. Road networks in the city centres of Pretoria and Johannesburg

domains is rotated before testing the similarity between the point patterns. Herein, we consider rotations at 10 degree intervals. Figure 9(b) is an illustration of where the Pretoria region is rotated at 10 degrees. The grid is determined considering the union of the two domains. In Figure 9(c) the Pretoria region is rotated at 20 degrees.

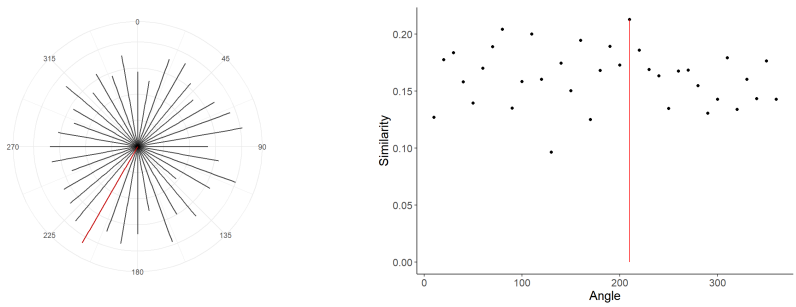
Figure 10 shows the results for all the rotations considered. In each case, a regular grid of size 20×20 was considered. Non-overlapping grid cells, i.e. the grid cells included in only one of the domains, are still considered in the similarity test but are classified as dissimilar ($s_i = 0$). All similarities are relatively low, indicating that the road networks in the two city centres are not similar when compared to the thresholds above. At a rotation of 210 degrees, the road networks are the most similar. The results, as presented in Figure 10, provide a useful visualisation of the similarity of the data on different domains. This approach has not been suggested before. The comparison of different cities is complex, as the roads arise due to a number of different factors. However, road structure in a area talks to accessibility and the use of this similarity test could be harnessed in future for accessibility modelling.

5. Discussion

From the simulation study conducted, it is clear that a re-examination of the threshold is necessary. Even patterns simulated to be 80% similar have traditional thresholds less than 80%. The same is observed for the simulated 90% similarity cases. The proposed approach provides an objective decision on similarity and considers the second-order nature of point patterns.

(a) (b) (c)

Figure 9. Illustration of how the union of the two domains are divided into a grid. (a) Both regions are at their original rotations, (b) The Pretoria city centre region is rotated 10 degrees, (c) The Pretoria city centre region is rotated 20 degrees.



(a) (b)

Figure 10. Resulting similarities at the various rotations with the angle of rotation leading to the highest similarity indicated in red.

The choice of the grid has an impact on the results, with higher thresholds observed with finer grid sizes, both regular and irregular. This is, of course, a general observation in spatial statistics; the grid size choice is always a contentious discussion. Since the threshold will plateau at some point as the grid size is further reduced, the consideration of multiple sizes in a use case is suggested. The study has shown that for higher resolution grids, the spatial similarity test proposed produces more stable results.

Irregular regions in real applications, for example, municipal demarcations, are more common than imposing a grid, in the past use of the traditional S-Index. Such lattice data type regions usually have fewer areas and are pre-defined. The simulations with the least areas, 25 irregular regions, performed the worst of all the grid sizes, indicating that the use of these pre-defined demarcations may skew similarity results. In such cases, imposing a grid and investigating a re-aggregation of points may be a more objective approach.

6. Conclusion

The proposed improved similarity test provides objectivity in decisions, as well as a data-driven angle to this objectivity. The methodology proposed re-examines the similarity threshold of a well-used spatial similarity test by taking the second-order nature of the spatial data into account. An empirical,

data-driven approach with a simulation study was used to calculate new threshold values.

The improved test was applied to the road networks in the city centres of Pretoria and Johannesburg in South Africa. The application to road networks on different domains shows a real data complication. The approach considering various rotations added further objectivity to similarity understanding. Research of spatial linear networks is still growing and this application can be investigated in more depth in future. This work can inform the design of transportation infrastructure projects that prioritise accessibility, efficiency, and environmental sustainability. The similarity of road networks could further be used to eliminate the dependency of the road network when considering the analysis of events occurring in the proximity of a road network such as in Modiba et al. (2022).

There are many avenues for future work based on this first objective re-examination of the S-index. The simulation of clustered point patterns is not robust due to the complex nature of clustered structures, for example, inhomogeneous cluster sizes and shapes. Real clustered data requires more understanding and warrants further investigation in testing for similarity in such complex situations. Additionally, the simulation study considered well-chosen hyperparameters but could be extended to fully understand the impact of these, especially related to the cluster structure complexities. The ICS classification is binary, and due to the abundance of variation in cluster structures, more ICS categories for clustered point patterns may provide a better understanding of the patterns in general, as well as further improvement in similarity quantification. Determination of an optimal grid size, and the assessment of its impact, would add strength to the proposed methodology and warrants future investigation. Concrete guidelines for the grid choice would be hugely beneficial.

References

Alba-Fernández, M., Ariza-López, F., Jiménez-Gamero, M. D., and Rodríguez-Avi, J. (2016). On the similarity analysis of spatial patterns. *Spatial Statistics*, **18**, 352–362.

Andresen, M. (2009). Testing for similarity in area-based spatial patterns: A nonparametric Monte Carlo approach. *Applied Geography*, **29** (3), 333–345.

Andresen, M. (2016). An area-based nonparametric spatial point pattern test: The test, its applications and the future. *Methodological Innovations*, **9**.

Andresen, M. and Linning, S. (2012). The (in)appropriateness of aggregating across crime types. *Applied Geography*, **35** (1-2), 275–282.

Andresen, M., Linning, S., and Malleson, N. (2017). Crime at places and spatial concentrations: Exploring the spatial stability of property crime in Vancouver, BC, 2003-2013. *Journal of Quantitative Criminology*, **33** (2), 255–275.

Andresen, M. and Malleson, N. (2011). Testing the stability of crime patterns: Implications for theory and policy. *Journal of Research in Crime and Delinquency*, **48** (1), 58–82.

Andresen, M. and Malleson, N. (2013). Crime seasonality and its variations across space. *Applied Geography*, **43**, 25–35.

Andresen, M. and Malleson, N. (2014). Police foot patrol and crime displacement: A local analysis. *Journal of Contemporary Criminal Justice*, **30** (2), 186–199.

Baddeley, A., Rubak, E., and Turner, R. (2015). *Spatial Point Patterns: Methodology and*

Applications with R. CRC Press.

Borrajo, M., González-Manteiga, W., and Martínez-Miranda, M. (2020). Testing for significant differences between two spatial patterns using covariates. *Spatial Statistics*, **40**, 100379.

Breetske, G. and Andresen, M. (2018). The spatial stability of alcohol outlets and crime in post-disaster Christchurch, New Zealand. *New Zealand Geographer*, **74** (1), 36–47.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioural Sciences*. 2nd edition. Hillsdale, NJ, Erlbaum.

Cressie, N. (2015). *Statistics for Spatial Data*. John Wiley & Sons.

David, F. and Moore, P. (1954). Notes on contagious distributions in plant populations. *Annals of Botany*, **18** (1), 47–53.

de Melo, S., Matias, L., and Andresen, M. (2015). Crime concentrations and similarities in spatial crime patterns in a Brazilian context. *Applied Geography*, **62**, 314–324.

Feuntes-Santos, I., González-Manteiga, W., and Mateu, J. (2017). A nonparametric test for the comparison of first-order structures of spatial point processes. *Spatial Statistics*, **22** (2), 240–260.

Kirsten, R. and Fabris-Rotelli, I. N. (2021). A generic test for the similarity of spatial data. *South African Statistical Journal*, **55** (1), 55–71.

Linning, S. (2015). Crime seasonality and the micro-spatial patterns of property crime in Vancouver, BC and Ottawa, ON. *Journal of Criminal Justice*, **43** (6), 544–555.

Modiba, J., Fabris-Rotelli, I., Stein, A., and Breetzke, G. (2022). Linear hotspot detection for a point pattern in the vicinity of a linear network. *Spatial Statistics*, **51**, 100693.

O'Brien, R. (2007). A caution regarding rules of thumb for variance inflation factors. *Quality & Quantity*, **41** (5), 673–690.

Pereira, D. V., Mota, C. M., and Andresen, M. A. (2017). The homicide drop in Recife, Brazil: A study of crime concentrations and spatial patterns. *Homicide Studies*, **21** (1), 21–38.

Ratcliffe, J. (2004). Geocoding crime and a first estimate of a minimum acceptable hit rate. *International Journal of Geographical Information Science*, **18** (1), 61–72.

Schutte, C. and Breetske, G. (2018). The influence of extreme weather conditions on the magnitude and spatial distribution of crime in Tshwane (2001 - 2006). *South African Geographical Journal*, **100** (3), 364–377.

Schwertman, N. and de Silva, R. (2007). Identifying outliers with sequential fences. *Computational Statistics & Data Analysis*, **51** (8), 3800–3810.

Vandeviver, C. and Steenbeek, W. (2019). The (in)stability of residential burglary patterns on street segments: The case of Antwerp, Belgium 2005–2016. *Journal of Quantitative Criminology*, **35** (1), 111–133.

Wang, W. (2013). A note on the bootstrap confidence interval for proportions. *Statistics & Probability Letters*, **83**, 2699–2702.

Wheeler, A., Steenbeek, W., and Andresen, M. (2018). Testing for similarity in area-based spatial point patterns: Alternative methods to Andresen's spatial point pattern test. *Transactions in GIS*, **22** (3), 760–774.

Submitted to the Proceedings of the 65th Annual Conference of the South African Statistical Association

Spatial network analysis for predicting future densification within South African informal settlements

R. van der Walt, C. van Zyl, R. N. Thiede and I. N. Fabris-Rotelli

Department of Statistics, University of Pretoria, Pretoria, South Africa

An informal road network is a system of roads which develop without formal planning or design. These networks can be modelled as spatial linear networks. Predicting how informal settlements will densify in future is important, since it is a metric that can be used to make decisions regarding infrastructure. The existing informal road network, as well as accessibility to points of interest on the road network, will inform where this densification will occur. Accessibility to points of interest are influenced by many factors, such as distance. In this paper, topography is included in addition to distance to assess accessibility. This is achieved by developing a novel routing algorithm which calculates shortest paths considering both distance and topography. Using these routing calculations, which may differ from traditional routing calculations using only distance, accessibility patterns are analysed, and statistically significant hot- and coldspots are identified, which can be used to make predictions on which areas might densify in future.

Keywords: Accessibility, Coldspots, Hotspots, Informal settlements, Spatial linear networks.

1. Introduction

The United Nations Sustainable Development Goals (SDGs) outline universal targets for improving global quality of life. Goal 9[1] of these SDGs targets the development of accessible infrastructure and Goal 11[2] focuses on making settlements inclusive. Together, these goals aim to provide access to road networks and essential facilities, promote sustainable urban development, and ensure that infrastructure and services are accessible to all, especially in vulnerable communities.

Melusi is a burgeoning settlement located in Western Pretoria[3]. The Melusi settlement is subdivided by formal and informal roads. An informal road network is a system of roads which develop without formal planning or design. The surrounding landscape also varies, with sloped hills and dam structures likely influencing the locations of dwellings. Kekana et al. (2023) provide various challenges that informal settlements face, such as limited availability of clean water and transport. This research analyses the road network in Melusi as a spatial linear network.

Corresponding author: R.N. Thiede (renate.thiede@up.ac.za)

MSC2020 subject classifications: 62H11, 62P12, 62P25

[1] https://sdgs.un.org/goals/goal9#targets_and_indicators

[2] https://sdgs.un.org/goals/goal11#targets_and_indicators

[3] https://saprin.mrc.ac.za/grtinspired.html

A spatial linear network can be defined as the collection of line segments on a plane, where each line segment consists of a line with a vertex at each respective endpoint (Ang et al., 2012). Vertices on this plane are therefore connected by these line segments, and each line segment has an associated length in Euclidean space. Consequently, there is a cost associated with the traversal of these linear networks when they are situated in geographic space (Barthélemy, 2011).

This research explores spatial linear network optimisation in the Melusi informal settlement. Factors that were considered included in the optimisation process are topography and distance. When optimising a spatial linear network, distance is a crucial factor to be considered. Finding the shortest path between two points in a network can be done using one of many well-established shortest path algorithms. The Floyd-Warshall algorithm (Floyd, 1962; Warshall, 1962), Dijkstra's algorithm (Dijkstra, 1959) and the Bellman-Ford algorithm (Bellman, 1958; Ford, 1956) are prime examples (Magzhan and Jani, 2013). These algorithms structure a shortest path problem as a graph, with nodes and edges. Dijkstra's algorithm and the Bellman-Ford algorithm find the shortest path from the input node to every other node. The Floyd-Warshall algorithm will determine the shortest path for all pairs of nodes. Dijkstra's algorithm (Dijkstra, 1959) will be extended in this paper, to account for topography in the calculation of shortest paths. This is to account for the relevance of changes in topography for pedestrians and cyclists, since an individual will need to exert more physical effort to traverse extreme topographical changes. It is also an especially relevant consideration in informal settlements, since many individuals do not have access to private motor vehicles.

Furthermore, statistically significant hot-and-coldspots within Melusi will be identified to predict areas most likely to attract new dwellings, emphasizing the relationship between accessibility and settlement growth.

In the sections to follow, a novel routing algorithm will be described which considers topography in addition to distance when calculating shortest paths. The identification of hotspots and coldspots can finally be used for prediction of future densification.

2. Methodology

In order to determine accessibility within informal settlements, a novel routing algorithm is described in which topography and distance are both considered during shortest path calculations. Accessibility between two locations is measured by distance, with shorter distances indicating better access. Evaluating the connection between a location and a point of interest enables determine whether the area is well-served or under-served. However, accounting for topography is equally crucial, particularly in informal settlements where foot traffic and informal transit systems dominate. Steep terrain, road types, and overall road quality can significantly impact access to essential facilities, creating mobility barriers for residents. While proximity to services is a key factor in accessibility assessments, neglecting the effects of topography may further hinder infrastructure development and service delivery in these communities.

2.1 Shortest path algorithms

Shortest path solutions are usually modelled by graph structures (Magzhan and Jani, 2013). These graph structures can be used to represent geographical data. A **graph** $G = (N, E)$ (Zhang and Chartrand, 2006) can be defined as a mathematical system consisting of two sets of elements: N,

nodes and E, edges. Nodes in the graph are connected by edges. The point where an edge is connected to a node is called an endpoint. A **weighted graph** (Zhang and Chartrand, 2006) is a graph where each edge has an associated weight. Furthermore, graphs used to determine shortest paths may be undirected or directed. In an **undirected graph**, edges do not have a specified direction. An edge connecting nodes A and B is bidirectional, meaning that it could be traversed from A to B, or from B to A. A **directed graph**, on the other hand, is a graph where edges have a specified direction. An edge that connects node A to node B could then be traversed from A to B, but vice versa. Herein, we will make use of undirected graphs.

A **spatial linear network** is a collection of line segments, where each line segment consists of a line (or an edge) with a vertex (or a node) at each endpoint (Barthélemy, 2011). These line segments are situated in space and have associated costs of travel.

In this research, weighted, undirected graph structures are used to represent spatial linear networks. These graph structures capture the cost and direction of traversal associated with the linear network, as well as topographical information associated with each node and edge. To optimise these networks, different shortest path algorithms are considered and compared. A slope estimation method will also be described. The `sfnetworks` data structure in `R` will be used to model spatial networks. Such a network is simply called an `sfnetwork` (van der Meer et al., 2023).

2.1.1 Dijkstra's shortest path algorithm

Dijkstra's algorithm (Dijkstra, 1959) finds the shortest path from an input node to every other node. This algorithm was selected as the basis for the proposed custom routing algorithm to come. This is because Dijkstra's algorithm is more time efficient on large and complex networks compared to Floyd-Warshall and Bellman-Ford, and since no negative weights will be present, it does not matter that Dijkstra cannot handle negative weights. It was also only necessary to compare the shortest paths from a small set of nodes to a small set of other nodes, therefore the entire set of shortest paths did not need to be calculated. Pseudocode for Dijkstra's algorithm can be found in Algorithm 1 in the Appendix.

2.1.2 Slope estimation

In their paper on evaluating methods of slope determination, Warren et al. (2004) confirm the use of Digital Elevation Models (DEMs) as an established tool for slope determination. They discuss a basic approach in which a slope percentage is calculated,

$$\text{Slope } \% = \left(\frac{\Delta z}{\Delta s}\right) \times 100,$$

where Δz = change in elevation between two points, and Δs = distance between the same two points.

2.1.3 Proposed custom routing algorithm

In order to take topography as well as distance into account when calculating shortest paths on spatial linear networks, it was decided that the existing Dijkstra's shortest path algorithm could be modified to achieve this, as previously noted.

The first step was to calculate an adjacency list. This adjacency list was calculated from the full set of edges in the `sfnetwork`, where the full set of edges contained each initial edge (e.g., the edge between node A to node B which contains its distance and elevation change) as well as each reversed

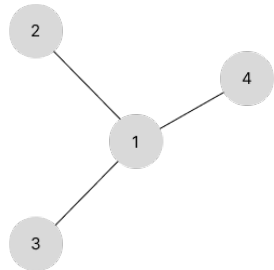


Figure 1. Node 1 with three neighbouring nodes.

Table 1. Information in the adjacency list for an example node 1.

Neighbouring node	Distance to neighbouring node (m)	Elevation change (m)
2	0.118	0
3	2.60	0
4	61.0	-2

edge (e.g. the edge between node B to node A would be explicitly added, also containing its distance and elevation change). This approach ensures that the algorithm will pick up the reverse traversal that was possible on each edge, i.e. that every edge in the network should be bidirectional. The adjacency list finally contains a table per node with outward edges, where each table contains the neighbours of that node, as well as the applicable information (the distance to the neighbour and the elevation change to the neighbour). Consider a node called node 1, with 3 neighbouring nodes, illustrated in Figure 1. An example of the adjacency list entry for node 1 can be seen in Table 1.

After the adjacency list is created, the elevation penalty function is defined as the absolute elevation of the edge, divided by the distance of the edge, multiplied by 100. This elevation penalty function can be altered without altering the core algorithm, to customise the application.

Lastly, the custom routing algorithm based on Dijkstra's shortest path algorithm could be implemented. This algorithm computes the shortest path between a given start node and end node, taking distance and topography into account. Pseudocode for the algorithm is given in Algorithm 2 in the Appendix. Key steps of the algorithm are as follows:

1. Initialise a distance table with ∞ as the initial distance between the starting node and every other node, and 0 is the distance between the starting node and itself. The distance table will track the shortest known distance between the start node and every other node.

2. Initialise a priority queue. Priority queues form an essential part in efficiently implementing Dijkstra's algorithm (Chen et al., 2007). In R, the tibble data structure was used to implement priority queues in this algorithm. This queue is used to efficiently process the next node with the smallest distance.

3. Initialise the list that is used to construct the shortest path between the start node and the end node.

4. While the priority queue is not empty, the following logic is looped over:

(a) Get the node v_i in the priority queue with the smallest distance $d(v_s, v_i)$.

(b) Retrieve all the neighbours v_j of the current node v_i. Compute the alternative distance to the neighbours by routing through the current node v_i. In this step, the alternative distance $d_{alt}(v_s, v_j)$ is calculated as

$$d_{alt}(v_s, v_j) = d(v_s, v_i) + w(v_i, v_j) + \text{elevation penalty},$$

where $w(v_i, v_j)$ is the weight of the edge between nodes v_i and v_j. Since we are using distance as the weight, $w(v_i, v_j)$ is thus the length of the edge from node v_i to v_j. The elevation penalty inflates the distance of edges with a high slope percentage, making them less likely to be selected in the shortest path calculation.

(c) If the newly calculated distance $d_{alt}(v_s, v_j)$ is smaller than the previously known distance $d(v_s, v_j)$ to the neighbour v_j, the algorithm updates the distance in the distance table, records the previous node $p[v_j] = v_i$, and adds the neighbour v_j to the priority queue.

(d) Once the end node v_t is reached, the shortest path from the start node v_s to the end node v_t is reconstructed using the predecessor data structure p.

Results from the custom routing algorithm are then compared to those of built-in algorithms in R, which consider only distance.

2.2 Hot route methodology

Chakravorty (1995) defines a hotspot as a smaller area within a broader region that contains an above-average concentration of points as relative to its surrounding areas. A linear hotspot is defined as a line segment that indicates an above-average density of spatial point patterns compared to line segments in its vicinity (Modiba et al., 2022). To identify linear hot- and coldspots, an intensity measure is required. Diggle (2013) proposed measuring the intensity of a point pattern as the average amount of points per specified area. This intensity function, denoted as $\lambda(x)$ represents the expected density of points at location x and is defined as:

$$\lambda(x) = \lim_{|dx| \to 0} \left(\frac{\mathbb{E}[N(dx)]}{|dx|} \right).$$

In the equation, x represents a location in the specified are, while dx represents an infinitesimal region around x. $|dx|$ represents the area of the region dx. $E[N(dx)]$ represents the expected number of points in a specified area. The limit $|dx| \to 0$, ensures that the intensity function captures local variations in point density rather than averaging over a larger region.

Tompson et al. (2009) developed the hot route methodology specifically for points situated directly on a linear network. Modiba et al. (2022) extended the idea to points situated in the surrounding area of the network. In Modiba et al. (2022), a point can be assigned to a line segment using one of two methods. In the first method, a point can be assigned to a single line segment by mapping it onto the nearest line segment. In the second method, a point can be assigned to multiple line segments by making use of weights. The weights are determined by how far a point is from each line segment. The weighted counts for a given radius r are computed using the equation $c^w_{ik}(r) = \sum_{j=1}^{m} w^j_{\ell_{ik}} 1^j_{\ell_{ik}}(r)$, where the indicator function

$$1^{j}_{\ell_{ik}} = \begin{cases} 1 & \text{if } ||p^{j}_{\ell_{ik}} - x_j|| \leq r, \\ 0 & \text{otherwise,} \end{cases}$$

determines whether a point x_j lies within a specified distance r from a given line segment ℓ_{ik}. If the distance between the point's projected location $p^{j}_{\ell_{ik}}$ on the segment and the point itself is less than or equal to r the function takes the value of 1, otherwise 0. In this equation $i = 1, 2, ..., n$ indexes different line segments, while $j = 1, 2, ..., m$ indexes the points in the study area. The term $w^{j}_{\ell_{ik}}$ represents the weight between point x_j and line segment ℓ_{jk}, which is determined by the inverse distance between them. The weight is given by $w_{ij} = ||p^{j}_{\ell_{ik}} - x_j||^{-1}$ as proposed by (Suryowati et al., 2018).

Using the weighted approach, the weight assigned to a point increases if the point is closer to a particular line segment compared to nearby line segments. Weights are assigned only to points that fall inside a radius r of the line segment of interest, where r is chosen appropriately. We use a weight count of points for each line segment. A small weight will indicate that a point is further away from a certain line segment. The event rate per distance for each line segment ℓ_{ik} is,

$$r_{ik} = \frac{c_{ik}}{|\ell_{ik}|} \text{ or } \frac{c^{w}_{\ell_{ik}}}{|\ell_{ik}|}, \tag{1}$$

where c_{ik} or $c^{w}_{\ell_{ik}}$ are the counts and $|\ell_{ik}|$ denotes the standardised length of the line segment. The r_{ik}'s visualises the concentration of points along each line segment.

To identify statistically significant line segments (hot- or coldspots), we compare the amount of points in the vicinity of each line segment to that of its neighboring segments, then identify those with above-average and below-average counts. Modiba et al. (2022) proposed the following definition for neighbouring line segments. A weight matrix $E = [e_{st}]$ represents the arrangement of the neighbours, where line segments ℓ_s and ℓ_t are labelled neighbours if $e_{st} = 1$. Here s and t are indices in $I = \{i, k : i = 1, 2, ..., n, k = 1, 2, ..., k_i\}$.

Definition 1. Let ℓ_{k_1} and ℓ_{k_2} denote multiple line segments characterised as

$$\ell_{k_1} = \{p_{k_1} = (x_{k_1}, y_{k_1}) : y_{k_1} = m_{k_1} x_{k_1} + c_{k_1}\}$$

and

$$\ell_{k_2} = \{p_{k_2} = (x_{k_2}, y_{k_2}) : y_{k_2} = m_{k_2} x_{k_2} + c_{k_2}\}.$$

Let $d_{\ell_{k_1},\ell_{k_2}} = \min ||p_i - p_j : p_i \in \ell_{k_1}, p_j \in \ell_{k_2}||$ be the smallest distance between ℓ_{k_1} and ℓ_{k_2} and M_{k_1}, the midpoint of ℓ_{k_1}.

1. If $d_{\ell_{k_1},\ell_{k_2}} = 0$ then ℓ_{k_1} is a linear neighbour of ℓ_{k_2}.
2. If $d_{\ell_{k_1},\ell_{k_2}} \leq r$ then ℓ_{k_1} is a radial linear neighbour of ℓ_{k_2}.
3. If $d_{M_{k_1},\ell_{k_2}} = \min ||M_{k_1} - p_j : p_j \in \ell_{k_2}|| \leq r$, then ℓ_{k_1} is considered a radial midpoint linear neighbour of ℓ_{k_2}.

Considering all three spatial structures defined in Definition 1 are essential (Modiba et al., 2022). It allows for a detailed assessment over a chosen radius r, therefore ensuring accurate identification of significant hot- or coldspots.

2.3 Hotspot detection

Traditional hotspot detection employs the Getis−Ord statistic to locate statistically significant hotspots by evaluating the sum of features relative to their neighbours against the overall sum of the features (Getis and Ord, 1992). In this approach, the features refer to the event rates for each line segment. The adapted Getis−Ord statistic (Modiba et al., 2022) tests the null hypothesis of complete spatial randomness versus the alternative hypothesis of a pattern existing in space. The statistic is defined as

$$G_{ik} = \frac{\sum_{t \in I, s=ik} e_{st} r_{rk}}{\sum_{i=1}^{n} r_{ik}}. \tag{2}$$

In this equation, G_{ik} represents the Getis–Ord hotspot score for a given line segment ℓik. The numerator consists of a weighted sum of event rates, where e_{st} denotes the observed event rate for segment s at time t, and r_{rk} is a spatial weight reflecting the relationship between segment ℓik and its neighbouring segments. The denominator normalises this sum by the total weight assigned to the segment, ensuring that the statistic accounts for differences in local densities across the network.

To identify significant linear hotspots and coldspots, z-scores and p-values are used to assess the probability of observing the value under the null hypothesis of complete spatial randomness. A statistically significant hotspot is identified by $G^*_{ik} > 0$ and p-value $\leq \alpha$, whereas a statistically significant coldspot is identified by $G^*_{ik} < 0$ and p-value $\leq \alpha$.

3. Data

The South African Population Research Infrastructure Network (SAPRIN) has the potential to significantly improve health, social and economic well-being in disadvantaged and rapidly changing populations. Their promise entails generating accessible and up-to-date health and demographic data sourced from collaborative longitudinal studies conducted within diverse communities.

3.1 Boundary

The general area of Melusi is specified on the SAPRIN website. The western boundary of Melusi is the border between the Gauteng and North West provinces. The southern, northern and eastern borders were not explicitly stated in the data. Therefore, prominent roads were used to further demarcate the examined area. It was decided that the N4 would be the southern boundary of the area being considered in this application, and the R80 would specify the northern and eastern boundaries. These boundaries were used to draw a polygon around the borders of Melusi, using Google Earth software, shown in Figure 2.

3.2 Data points

Dwelling location data was obtained SAPRIN (ethics application: NAS256/2024). The data contains the latitude and longitude coordinates of approximately 930 residency locations. The data serve as a representative sample of the Melusi area. These dwellings are visualised in Figure 3.

3.3 Road network

OpenStreetMap is an open-source platform that provides geographic data. The OpenStreetMap data of the South African road network for 2021 was used, which corresponds to the time in which the

Figure 2. Melusi polygon drawn using road demarcations as general boundaries, Map Data (c) Google 2024.

Melusi dwelling coordinates were collected. The South African road network was filtered using QGIS software to obtain only a subset of the roads within the Melusi polygon, shown in Figure 3.

3.4 Points of Interest

The locations of points of interest (POI) were also obtained through OpenStreetMap. The locations of interest are filtered to contain schools, police stations and hospitals that fall within the Melusi boundary. These three essential POIs were selected as they provide fundamental services that directly affect health, safety, and educational opportunities, which are crucial for the sustainable development of informal settlements. These POIs are represented by polygons. In order to perform shortest path algorithms a point location of these polygons are required. The exact point is calculated by taking the center of the polygon, then snapping the centroid to the nearest node on the road network. This results in point pattern data for each of the points of interest on the road network. In total there were 37 schools, 2 hospitals and 3 police stations. The nearest existing nodes in the road network were identified to represent each POI. The house locations as well as POIs on the road network can be seen in Figure 3.

3.5 Topography

A Digital Elevation Model (DEM) provides topographical information of an area. Using the DEM for the Melusi area, the elevation information of each node in the spatial linear network could be obtained. The change in elevation associated with each edge was calculated by using this elevation information, where the change in elevation was calculated as the difference in elevation between two nodes in the spatial linear network. A positive change in elevation indicated a downward slope and a negative change in elevation indicated an uphill slope. These attributes, along with the distance of each road segment, were used to calculate the slope percentage of each road segment, where the slope percentage is related to the change in elevation in two points divided by the distance between the same two points, as discussed previously. The elevation model of Melusi can be seen in Figure 4, where elevation is measured in meters above sea level.

Figure 3. Schools, hospitals and police stations within Melusi boundary.

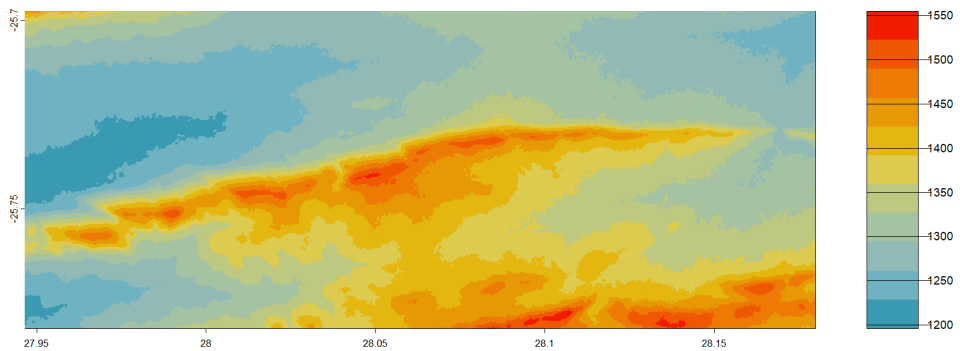


Figure 4. Elevation variation within Melusi landscape, measured in meters (m) above sea level.

4. Results

The objective of this research was to explore accessibility through spatial linear network optimisation in the Melusi informal settlement, specifically considering the accessibility from dwellings to specific points of interests, namely hospitals, schools and police stations. Factors to be considered were topography and distance. To implement this, a custom routing algorithm was developed in which both distance and topography were taken into account, and compared to established routing algorithms in which distance is the sole factor in route determination. In addition, hot and cold spot detection was performed to identify potential areas for future expansion. All implementation was done in R (R Core Team, 2024).

4.1 Custom routing algorithm results

This section compares results from the proposed custom routing algorithm considering both distance and topography to Dijkstra's algorithm, in which distance is the only factor considered. The shortest path between each cluster centre, and each point of interest was calculated. Results related to schools are given below. This can be applied to other points of interest in the area, such as hospitals and police stations, as well.

All schools within the Melusi boundary were considered, including primary schools and high

schools with varying languages of instruction. This totalled 37 schools, and therefore 74 routing solutions were calculated with the respective algorithms. Of these 74 routing solutions, 44 routing solutions differed when using the proposed custom routing algorithm compared to the built-in solution. Figure 5 shows all the routes as they were originally calculated using the built-in solution. The cluster centres are represented by red asterisks, and the schools are represented by yellow asterisks. Each route is represented by a unique colour, for ease of interpretation. Figure 6 shows all the routes calculated by the custom routing algorithm, similarly displaying each route in the same unique colour as its corresponding route in Figure 5. Some routes only differ by a small number of nodes.

To more clearly visualise the differences between the two routing approaches, Figure 7 shows all the built-in routing solutions plotted in orange, and all the custom routing solutions overlaid in blue. As before, red asterisks represent cluster centres, and yellow asterisks represent schools.

Where routes do differ between the built-in routing solutions and the custom routing solutions, some routes only differ by a few nodes. This might indicate that the informal road network that has naturally developed in Melusi, which is frequented by pedestrians, has already taken topography into account in the establishment of any informal road segments. The road network has been developed for convenience of the community, and ease of movement for pedestrians was likely a contributing factor in the development of the existing road structure.

4.2 Statistically significant hot-or-coldspots

The rate of houses per line segment is calculated using the weighted rate of events calculation in Equation 1. A radius $r = 50$ was chosen to ensure house locations are weighed to road segments nearby. The radius also ensures that the line segment neighbours given in Definition 1 are satisfied.

Statistically significant line segments are identified using the Getis-Ord statistic in Equation 2. All three neighbourhood structures are consider in the calculation as defined in Definition 1. Figure 8 visualises statistically significant hotspots. Three levels of significance are chosen, $\alpha = 1\%, 5\%$ and 10%. A statistically significant hotspot is identified by $G^*_{ik} > 0$ and p-value $\leq \alpha$. Figure 9 visualises statistically significant coldspots. A statistically significant coldspot is identified by $G^*_{ik} < 0$ and p-value $\leq \alpha$.

Formal settlements expand in the vertical and horizontal plane. These settlements face fewer restrictions regarding how far or high they can grow, allowing them to accommodate increasing populations more flexible. As the population grows in formal settlements hotspots, areas of high activity or density increase, but so do colder spots, as the expansion creates a more diverse spatial distribution of activity.

In contrast, informal settlements display distinct patterns of growth. Unlike formal settlements, which expand both upward and outward, informal settlements primarily expand outward until they encounter a boundary. These boundaries may be imposed by surrounding formal settlements, natural environmental characteristics, or infrastructural constraints. As a result, the expansion of informal settlements is often constrained, leading to a saturation point where hotspots reach a population limit. When this happens, the proportion of neutral and cold spots within the settlement increases as the population grows, reflecting the spatial and structural limitations inherent in informal urban expansion. Figure 8 and Figure 9 indicate the expansion patterns mentioned previously.

This understanding allows us to predict where residents will settle next. Coldspots near essen-

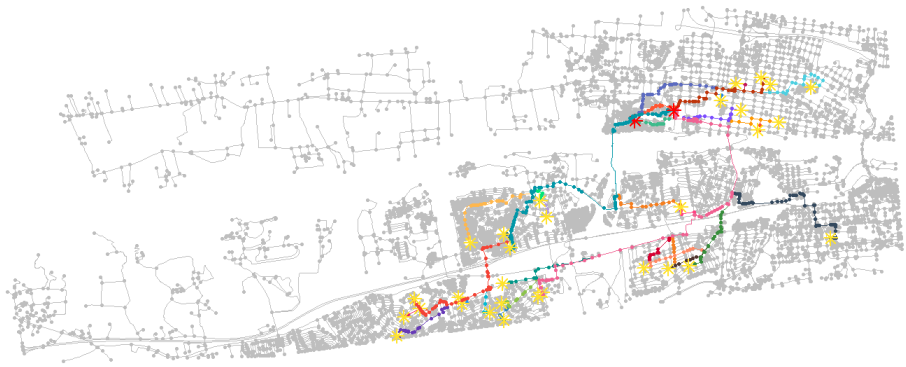


Figure 5. Routing solutions yielded by built-in shortest path algorithm from cluster centres (red asterisks) to schools (yellow asterisks). Each route is represented by a unique colour.

Figure 6. Routing solutions yielded by custom routing algorithm from cluster centres (red asterisks) to schools (yellow asterisks). Each route is represented by the same unique colour as its corresponding route in Figure 5.

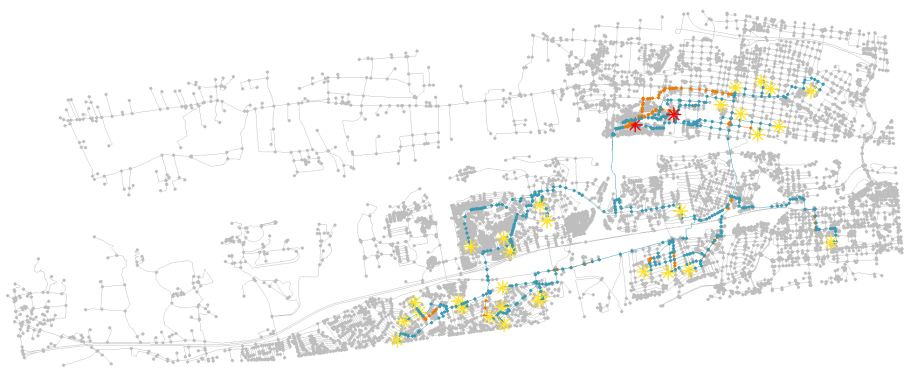


Figure 7. Built-in routing solutions in orange and custom routing solutions in blue from cluster centres (red asterisks) to schools (yellow asterisks). Orange routes represent built-in routing solutions, and blue routes represent custom routing solutions.

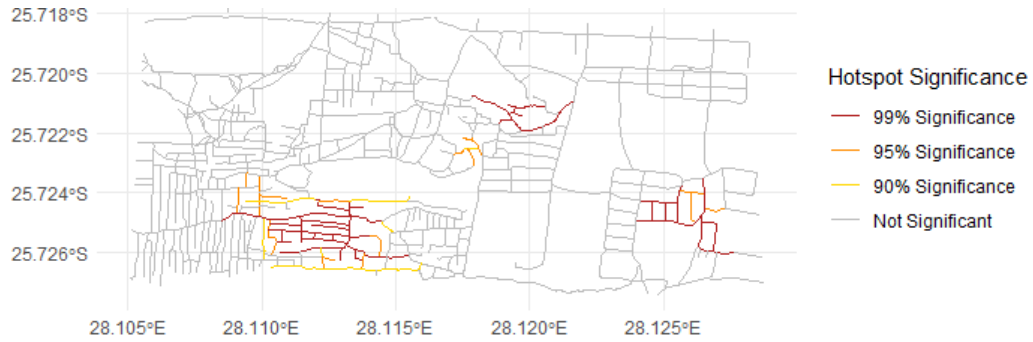


Figure 8. Statistically significant hotspots.

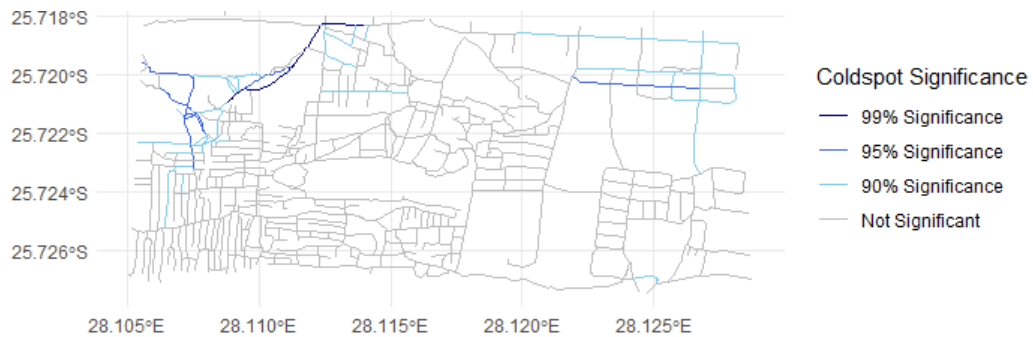


Figure 9. Statistically significant coldpots.

tial facilities are likely to experience the most growth, as accessibility drives settlement patterns. Similarly, neutral areas adjacent to existing hotspots can also attract residents, as these areas offer proximity to activity without the saturation of established hotspots.

5. Conclusion

This research introduced a novel routing algorithm that integrates topography in addition to distance, and shows that different routing solutions can arise when topography is considered. Statistically significant hot-and coldspots are identified to evaluate future densification within informal settlements.

Many aspects of this study can be expanded on in future work. The elevation penalty used throughout can be fine-tuned, since this application penalised all slope changes. It might be beneficial to penalise uphill slope changes more, when multiple paths diverge at a point with similar slope percentages. It might also be useful to promote some downhill slope changes, e.g. if the slope change is very slight, it might be an easier route to take.

Future work can also investigate how routing solutions differ when using a custom routing algorithm taking topography as well as distance into account, compared to routing algorithms which consider only distance, when exploring formal settlements. This is relevant since this research was only applied to an informal settlement where convenient movement of pedestrians was likely already considered as a factor in the development of the road structure.

Furthermore, future research could explore the temporal evolution of hot- and coldspots within

informal settlements. By analysing how hotspots shift over time, it would be possible to predict future settlement growth, and assess how accessibility influences long-term spatial patterns. This could provide valuable insights for urban planning and infrastructure development in rapidly changing environments.

The identification of hot- and coldspots within the settlement provides further insight into accessibility and densification. Coldspots and neutral regions, particularly those near essential facilities, present opportunities for future growth as residents seek areas with better access to key services. Understanding these spatial dynamics helps anticipate informal settlement expansion and informs potential infrastructure planning.

6. Acknowledgements

This work is partially based upon research supported by the South Africa National Research Foundation (NRF) Grant number 137785. Opinions expressed and conclusions arrived at are those of the authors and are not necessarily to be attributed to the NRF.

Appendix

Algorithm 1 Dijkstra's Algorithm

while u is not empty **do**
  $currentNode \leftarrow$ node in u with minimum $d[node]$
  for each $neighbour$ of $currentNode$ that is in u **do**
    $neighbourEdge \leftarrow$ edge connecting $currentNode$ and $neighbour$
    $currentLength \leftarrow d[currentNode] + neighbourEdge$
    if $currentLength < d[neighbour]$ **then**
      $d[neighbour] \leftarrow currentLength$
      $p[neighbour] \leftarrow currentNode$
    end if
  end for
  Remove $currentNode$ from u
end while

Algorithm 2 Modified Dijkstra's Algorithm with Elevation Penalty

Initialise priority queue u with all nodes
while u is not empty **do**
  $currentNode \leftarrow$ node in u with minimum $d[node]$
  for each $neighbour$ of $currentNode$ that is in u **do**
    $neighbourEdge \leftarrow$ edge connecting $currentNode$ and $neighbour$
    $distance \leftarrow$ distance of $neighbourEdge$
    $elevationChange \leftarrow$ elevation change of $neighbourEdge$
    $elevationPenalty \leftarrow \left|\frac{elevationChange}{distance}\right| \times 100$
    $adjustedLength \leftarrow d[currentNode] + distance + elevationPenalty$
    if $adjustedLength < d[neighbour]$ **then**
      $d[neighbour] \leftarrow adjustedLength$
      $p[neighbour] \leftarrow currentNode$
    end if
  end for
  Remove $currentNode$ from u
end while

References

Ang, Q. W., Baddeley, A., and Nair, G. (2012). Geometrically corrected second order analysis of events on a linear network, with applications to ecology and criminology. *Scandinavian Journal of Statistics*, **39** (4), 591–617.

Barthélemy, M. (2011). Spatial networks. *Physics Reports*, **499** (1-3), 1–101.

Bellman, R. (1958). On a routing problem. *Quarterly of Applied Mathematics*, **16** (1), 87–90.

Chakravorty, S. (1995). Identifying crime clusters: The spatial principles. *Middle States Geographer*, **28**, 53–58.

Chen, M., Chowdhury, R. A., Ramachandran, V., Roche, D. L., and Tong, L. (2007). Priority queues and Dijkstra's algorithm. Technical report, The University of Texas at Austin.

Diggle, P. J. (2013). *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns*. CRC press.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, **1** (1), 269–271.

Floyd, R. (1962). Algorithm 97: Shortest path. *Communications of the ACM*, **5** (6), 345–345.

Ford, L. R. (1956). *Network Flow Theory*. Rand Corporation, Santa Monica, CA.

Getis, A. and Ord, J. K. (1992). The analysis of spatial association by use of distance statistics. *Geographical Analysis*, **24** (3), 189–206.

Kekana, H., Ruhiiga, T., Ndou, N., and Palamuleni, L. (2023). Environmental justice in South Africa: The dilemma of informal settlement residents. *GeoJournal*, **88** (4), 3709–3725.

Magzhan, K. and Jani, H. M. (2013). A review and evaluations of shortest path algorithms. *International Journal of Scientific and Technology Research*, **2** (6), 99–104.

Modiba, J., Fabris-Rotelli, I., Stein, A., and Breetzke, G. (2022). Linear hotspot detection for

a point pattern in the vicinity of a linear network. *Spatial Statistics*, **51**, 100693.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Suryowati, K., Bekti, R., and Faradila, A. (2018). A comparison of weights matrices on computation of dengue spatial autocorrelation. *In IOP Conference Series: Materials Science and Engineering*, volume 335. IOP Publishing, p. 012052.

Tompson, L., Partridge, H., and Shepherd, N. (2009). Hot routes: Developing a new technique for the spatial analysis of crime. *Crime Mapping: A Journal of Research and Practice*, **1** (1), 77–96.

van der Meer, L., Abad, L., Gilardi, A., and Lovelace, R. (2023). *sfnetworks: Tidy Geospatial Networks*. https://github.com/luukvdmeer/sfnetworks.

Warren, S. D., Hohmann, M. G., Auerswald, K., and Mitasova, H. (2004). An evaluation of methods to determine slope using digital elevation data. *Catena*, **58** (3), 215–233.

Warshall, S. (1962). A theorem on Boolean matrices. *Journal of the ACM (JACM)*, **9** (1), 11–12.

Zhang, P. and Chartrand, G. (2006). Introduction to Graph Theory. *Tata McGraw-Hill*, **2**.