Spatial Operations on Uncertain Positional Data

Welder B. Oliveira[1], Sávio S. T. Oliveira[2], Vagner J. S. Rodrigues[2], Helton Saulo[3], Kleber V. Cardoso[1]

[1] Institute of Informatics, Universidade Federal do Goiás, Goiânia, GO, Brazil
weldermat@gmail.com, kleber@inf.ufg.br
[2] GEOMAIS Serviços de Informática LTDA - ME, Goiânia, GO, Brazil
{savioteles, vsacramento}@gmail.com
[3] Department of Statistics, Universidade de Brasília, Brasília, DF, Brazil
heltonsaulo@gmail.com

Abstract. Positional errors on spatial data affect spatial join accuracy in an unexpected and undesirable way. In general, current probabilistic solutions barely achieve reasonable computational performance, unless they are employed in special cases such as when the errors follow a Circular Normal distribution.In this article, we present a general framework for spatial operations which is robust to positional imprecision in geographic coordinates. The framework is designed to be general in terms of the positional error distribution and provides parametric options for users to control efficiency and accuracy. Furthermore, we develop two new spatial join procedures: an adaptation of the Monte Carlo method to be used as a probabilistic step and a probabilistic efficient alternative to Minimum Bounding Rectangles (MBRs), which we call Confidence Rectangles.Empirical evidence suggests that our proposed methodologies significantly outperform current solutions in at least one of the three dimensions: generalism, efficiency and accuracy. In the worst case scenario, the proposed methodology is not significantly outperformed by any alternative solution in more than one of the three dimensions. Moreover, the user has the power to choose via parameter specification which dimension will be priorized instead of depending on the inherent advantages of each current solutions.

Categories and Subject Descriptors: J. [**Computer Applications**]: Miscellaneous; I.6 [**Simulation and Modeling**]: Miscellaneous; G.3 [**Probability and Statistics**]: Miscellaneous

Keywords: Uncertainty in positional data, Spatial join, Skyline query, Pareto efficiency, Monte Carlo method

1. INTRODUCTION

Spatial data are subject to several forms of uncertainty. In special, two forms stand out: existential and positional. The existential uncertainty is related to the confidence that the spatial object represented in the data actually exists. This can occur when extracting objects of a satellite image with low resolution or color definition. Therefore, a specific pixel may not be associated with a given object with a 100% confidence [Dai et al. 2005]. On the other hand, the positional uncertainty refers to the confidence in the object position. In this case, the dataset coordinates of the objects differ from their real locations. This type of uncertainty is also called positional error, or simply error. The error for a given object is defined as the distance between its actual coordinates and the coordinates that represent it in the dataset. The distance used may be of various types, such as geodesic, Euclidean, Manhattan, among others. The magnitude of the error is associated with the method used to produce the data. In such scenarios, computational operations may lack precision and accuracy. Operations like spatial joins - which look for intersections between two spatial objects - may return a false positive or false negative due to object position error. That may lead managers, whose decisions rely on these computational operations, to take wrong decisions.

Traditional spatial operations assume an absolute precision in the coordinates of the represented

objects, despite the fact that spatial data from most diverse sources possess different error patterns due to a number of factors. For example, [Hughes 2002] points out that the horizontal accuracy (with 95% confidence interval – CI) of the GPS (Global Positioning System) with WAAS (Wide Area Augmentation System) standard for the city of Los Angeles was 0.922 meters, while the vertical accuracy for Miami was 1.373 meters. Geocoding is a less precise georeferencing procedure than the GPS in which the geographical coordinates are obtained from addresses written in natural language. Typically, geocoding presents errors of at least 100 meters in about 20% to 30% of cases, according to [Faure et al. 2017]. In turn, methods based on satellite images can be affected, for example, by the scale used when generating the data. Some methods have been proposed in the literature to deal with both existential and positional uncertainty. [Pei et al. 2007] show how to compute a Skyline when multiple instances of the same object is provided. [Openshaw 1989] proposed the use of the Monte Carlo Method to estimate probabilities of intersection in spatial joins and [Ni et al. 2003] adapted the steps of a traditional spatial join solution assuming that the errors in the coordinates follow a Circular Normal distribution. However, the methodology discussed in [Pei et al. 2007] is only applicable if multiple instances of the objects are available, which usually is not the case; the Monte Carlo procedure suggested in [Openshaw 1989] is computationally too expensive and [Ni et al. 2003] approach requires errors to be Circular Normal. In this context, we propose a solution whose main contributions for the area of uncertain spatial operations are:

(a) to be applicable for any error distribution (generality, by accepting the probabilistic distribution provided by the solution's user);
(b) to be computationally practical (efficiency, as specified by the maximum number of Monte Carlo simulations); and
(c) to control false positives (accuracy, as specified by a parameter p which is the cut probability for assuming a match between two objects).

It is important to highlight that our contribution is not to provide a solution which satisfies (a) or (b) or (c), for which the above-cited authors have provided a solution, but (a) and (b) and (c) simultaneously. The solution user specifies its tolerance or requirements concerning each of these three dimensions. Thus, the user can apply our solution framework even if she/he

(1) does not possess multiple instances of each object as required by [Pei et al. 2007];
(2) possesses a dataset that is too large to be processed by a traditional Monte Carlo approach in reasonable time as in [Openshaw 1989];
(3) is not sure that the Circular Normal assumption for errors is reasonable as in [Ni et al. 2003];
(4) desires to control the balance between accuracy and efficiency in his/her applications (which the cited correlated solutions do not provide). That is the novelty that this article brings to the area.

In order to deliver those contributions, this article presents a framework for building spatial operations based on our proposed Progressive Monte Carlo Method (PMCM). To ilustrate its applicability, two operations - Spatial Skyline and Spatial Join are adapted with PMCM. PMCM helps mainly in providing a parametrization for the efficiency and accuracy of the solution, i.e., the balance between these two dimensions.

Empirical evidence provided by some tests (Section 4) points out that adopting the proposed framework may decrease the computational processing time by 71% when compared with a standard Monte Carlo approach. Furthermore, in the experiments the efficiency gain came with no loss in accuracy. Even if in some practical scenarios the developer faces a significant loss in accuracy, she/he may overcome that by adjusting the accuracy parameters. Tests performed in Section 4 showed that when that is done, the processing time does not increase significantly. Thus, the accuracy-efficiency trade-off is such that the solution still remains relatively efficient while accuracy standards significantly increase. Finally, the developer has the benefit of having a solution that does not require a specific probability

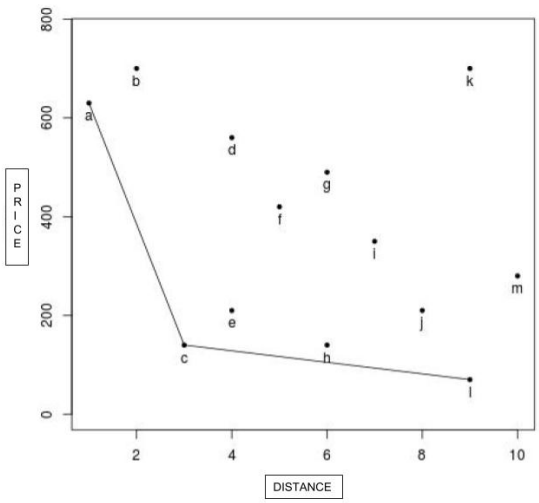

Fig. 1. Skyline Query: scenario 1.

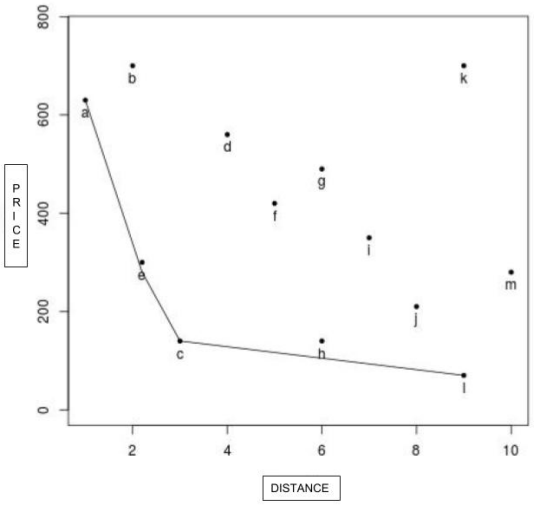

Fig. 2. Skyline Query: scenario 2.

Fig. 3. Skyline Query. Adapted from [de Oliveira et al. 2015].

distribution.

The rest of the article proceeds as follows. Section 2 presents the theoretical basis of the work and the main correlated works. Section 3 presents the two proposals for spatial operations with imprecision of spatial data: pSkyMCM (p-Skyline Monte Carlo Method) and PMCSJ (Progressive Monte Carlo Spatial Join). Lemmas and theorems are presented to prove the efficacy of the solutions presented. Section 4 presents and discusses the experimental results obtained. A conclusion section ends the article.

2. BACKGROUND

This section presents the main concepts relating the developed solutions in this article, namely spatial skyline and spatial joins. Furthermore, the literature in these two areas are discussed.

2.1 Skyline query

Skyline queries are one way to perform preference queries from a database by providing only the ordering direction of the attribute values [Chomicki et al. 2013]. This query type returns the efficient Pareto tuples of a dataset according to a number d of attributes and their ordering direction (maximization or minimization). Pareto efficiency is the property of objects that are not dominated by any other in a certain dataset. On the condition of minimization, point p_i dominates point p_j if and only if the coordinates of p_i in any dimension are not larger than the corresponding coordinates in p_j [Papadias et al. 2003].

As an example, a classic illustrative problem would be: "find hotels that are cheap and close to the beach". Thus, there are two objectives to be achieved, which can be mutually exclusive, i.e., the hotels closest to the beach may tend to be the most expensive. Naturally, if there is a hotel A that is cheaper and closer to the beach than other hotel B, then A is preferable to B concerning these two attributes.

Figures 1 and 2 (adapted from [de Oliveira et al. 2015]) show a set of hotels in relation to the two variables of interest: x-axis: "distance to the beach in meters"; y-axis: "daily price (R$)". The solid line represents the skyline S for the whole set of hotels concerning the two attributes. In Scenario 1, a is the closest hotel to the beach and l the cheapest one. Thus, these two hotels belong to S, since, by definition, they cannot be dominated by any other hotel. In addition to these two hotels, c also belongs to S, as it is not dominated by any another in this data set. All others are dominated by any of these three points. Therefore, as highlighted in Figure 1, $S = \{a, c, l\}$. However, a possible inaccuracy in the data can impact the query and change the solution. Scenario 2, Figure 2, shows how a position error of 150 meters from hotel e would include it in S. In this case, it makes sense to ask what would be the probability of e being Pareto efficient in this set concerning the two dimensions (or attributes). To deal with this sort of situations and incorporate the imprecision of attributes in the query, the concept of the p-skyline is introduced.

Given a set of spatial data D (such as the example with 13 hotels), we define p-skyline in D as the subset $S_p \in D$ formed by the points for which the probability of not being dominated by any of the other points of D is at least p. According to this definition and considering positional errors whose modular values can assume any positive real value, we have $S_0 = D$ and $S_1 = \phi$, since any point of D would have a probability of at least zero of belonging to S, just as no point would have 100% probability of being in S.

Since its introduction in [Borzsony et al. 2001], the execution of skyline queries has received considerable attention in the multidimensional database area. Several algorithms for obtaining skylines have been proposed. [Tan et al. 2001] use auxiliary structures for progressive skyline execution, [Kossmann et al. 2002] present a nearest neighbor algorithm, [Papadias et al. 2003] introduce the branch and bound algorithm for skyline (BBS) and [Chomicki et al. 2003] and [Chomicki et al. 2005] propose the sort-filter-skyline (SFS) algorithm, which acts by leveraging preordered lists, as well as an ordering by linear elimination.

The concept of space in the skyline query was introduced by [Sharifzadeh and Shahabi 2006]. Given a set of points P and a set of queries Q, at each point $p \in P$ is derived a number of spatial attributes corresponding to their distances to the query points. [You et al. 2013] propose the algorithm branch and bound farthest search (BBFS), comparing and demonstrating its superiority over the threshold farthest spatial skyline (TFSS). Efficient algorithms for TFSS using the Euclidean distance were proposed by [Son et al. 2009; Lee et al. 2011]. [Son et al. 2014] developed an algorithm using the Manhattan distance (also known as the taxi driver's distance). This metric is closely related to the real traveled distance between two points in a city than euclidean distance.

[Khalefa et al. 2008] present a solution for incomplete data, i.e., when there are missing data in some of the dimensions considered in the query. The authors generalized the dominance criterion as follows. Given any two points P and Q, which may have incomplete dimensions, point P dominates Q if the following two conditions are valid: 1) There is at least one dimension i where both $P[i]$ and $Q[i]$ are known, and $P[i] < Q[i]$; 2) For all other dimensions j, either $P[j]$ or $Q[j]$ are unknown or $P[j] \leq Q[j]$. Here, the symbols $<$ and $\leq$ denote the preferred sense of optimally, which may even be maximization. The traditional definition for complete data becomes a particular case of that proposed by [Lofi et al. 2013], who address the problem of incomplete data from a different perspective. The authors propose that the tuples with the greatest potential to degenerate the overall quality of the solution be shared with collaborators in time to resolve the missing data problem. A new probabilistic skyline model is proposed by [Ding et al. 2014], where an uncertain object may assume a skyline probability at a certain point in time.

[Pei et al. 2007] calculate skylines for uncertain data. According to the authors, uncertainty means that more than one instance is available for each attribute for the several objects under evaluation. The authors cite, as an example, the performance data of NBA players. For each player, statistics such as the number of assists and rebounds are considered. The higher the value reported in each of

these statistics, the better the player. As the players' performance varies from game to game, each of them possesses different values for the same statistic. To solve the problem, an alternative mentioned by the authors is to replace the several values of each attribute by their averages for each player. In this way, an A player would present a single value for rebound, which would be the average of their rebounds in the several games reported in the data set. Therefore, a traditional skyline solution would be sufficient to solve the problem of setting the best players in the championship. However, that would not allow the computation of probabilities that a given player is in fact among the best. When considering all the instances of each attribute, [Pei et al. 2007] derive such probabilities and introduce the concept of probabilistic skyline.

2.2 Spatial join

Before defining the spatial join, we introduce the concept of the join operation. Given two datasets A and B, it is called a join in relation to an attribute x common to both sets, the subset of Cartesian product $(A \times B)$ denoted by $(A \times B)_x$, containing all tuples (a, b) with a belonging to A and b belonging to B, such that there is a match between a and b with respect to x. The rule used to define whether there is a match between the two tuples is called the join predicate.

In many applications, the equality predicate is used, i.e., it is said that a and b match if $a = b$. For example, a given query to a database may have a goal of returning a table with the names and phone numbers of customers who are registered in a store, using two sets of data: $A = \{id, phone\_numbers\}$ and $B = \{id, people\_names\}$. In this case, a join of the tables by the id attribute meets the objective.

In the presence of spatial attributes, it is common to consider different predicates than those used to evaluate matching between textual or numeric attributes. Examples of typical predicates applied in join of spatial attributes are: the intersection and the property of being at a maximum of x meters away. Join operations that apply spatial predicates are called spatial joins. The scope of the present work is restricted to the predicate of the intersection. Figure 0**??** shows five case in which there is intersection in four of them: B, C, D, and E. Therefore, it is said that there is a match relating this predicate in these cases.

When there is uncertainty in spatial data, both in relation to existence and to the position of the objects, the concept of probabilistic spatial join (PSJ) is necessary. Probabilistic spatial join with cut probability p for two datasets A and B and spatial attribute x is the subset of the Cartesian (AxB) containing all tuples (a, b) with $a \in A$ and $b \in B$, such that the match probability between a and b concerning the spatial attribute x is at least equal to p.

Unlike the deterministic versions of the spatial join, where the verdict in predicate evaluation is boolean, i.e., true or false, in the probabilistic version one can only achieve a specific confidence in such a verdict. Thus, one cannot assure that the probability of the intersection is greater than or equal to p, but that there exists a known confidence γ in that statement. Thus, assuming there is a positional error in the coordinates of the polygons from Figure 4, one cannot decide with 100% confidence if there is a match in any of the five cases presented. However, assuming that such errors can be modeled with a probability distribution, it is reasonable to expect that the match probability in scenario B is greater than in scenario A. A PSJ should be able to assess such probability and return match only in cases where this exceeds the established p threshold.

The inherent probabilistic nature of a PSJ solution leads to both false positives and false negatives. For example, taking $p = 0.90$, there will be a false negative when the true probability of intersection is at least 90%, but PSJ does not return the match. Conversely, PSJ will incur a false positive when returning the match for two spatial objects whose true intersection probability is less than 90%. In a PSJ solution, accuracy can be measured as the proportion of correct assessments of the join predicate for all pairs of geometries or for a sample of that set. In Section 4, the accuracy of our and concurrent solutions are evaluated for pairs of geometries whose true probability of intersection is located at a

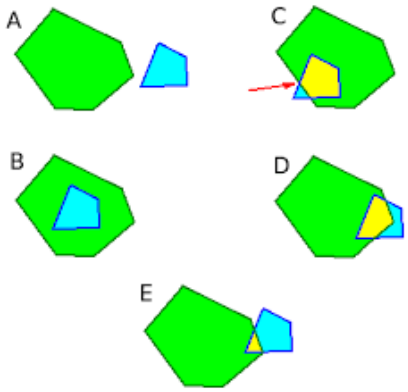

Fig. 4. Polygon intersection cases.

radius R of p.

There are several methods for performing spatial join using or not indexed datasets. If no dataset is indexed, a nested-loop join, [Mishra and Eich 1992], plane-sweep algorithm [Arge et al. 1998] and [Jacox and Samet 2003], or spatial join based on partitioning, [Luo et al. 2002] and [Patel and DeWitt 2000], can be used. If only one dataset is indexed, then the spatial join can be performed using a nested loop index, [Lo and Ravishankar 1994], which requires an index I_A for the dataset A, in addition to a loop for the dataset B and I_A queries for each object. If both (A and B) are indexed with R-Trees, [Guttman 1984], a synchronous path, [Brinkhoff et al. 1993] and [Huang et al. 2006] can be used to perform the operation. This process recursively traverses the trees to the level of the leaves where the objects are compared. However, these approaches do not deal with probabilistic spatial join.

Spatial join in data with existential uncertainty were explored by [Dai et al. 2005]. Moreover, [Ljosa and Singh 2008] present an approach capable of handling both existential and positional uncertainty, making use of a score function in which the two types of uncertainty are both taken into account. [Openshaw 1989] propose using the MMC to calculate the probabilities of intersection of geometries with positional uncertainty. We will call this solution Random Spatial Join (RSJ) and it will be evaluated and compared to our approach in Section 4.

In the PSJ proposed by [Ni et al. 2003], the error is modeled using a Circular Normal distribution. Due to some well-known specificities and properties of Normal distribution, the authors were able to adapt the filtering and refinement steps to achieve optimal accuracy and efficiency. We will call this spatial join approach Circular Normal Spatial Join (CNSJ) and we will discuss how this solution can be compared to our approach in the Section 4.

As RSJ can handle errors from multiple probability density functions (PDFs) and CNSJ is designed to work properly only with the Circular Normal distribution for the errors, then RSJ is more generalist than CNSJ. As we are proposing a solution – PMCSJ – in which generality is one of the requirements, CNSJ is not a direct competitor, but RSJ is. Therefore, PMCSJ will only be directly compared against RSJ, also in the Section 4.

In this article we also present an adaptation of the Minimum Bounding Rectangle (MBR) used in filtering steps in several spatial join algorithms. That is done by developing the concept of confidence rectangle (CR). [Tu et al. 2012] proposed normalized cross correlation to create "motion bounding box" to be applied in the context of detecting moving objects. [Arcaini et al. 2013] proposed a buffer d - with length chosen by the solution's user - to be added in the MBR corners. However, none of these proposals are flexible to be applicable for different PDFs as the generality requirement of our framework demands. In order to achieve that requirement, we made use of Chebyshev Inequality - which holds for a large class of PDFs - when building the CR. In section 3, we mathematically prove

the validity of CR to provide the required level of confidence for intersection evaluation.

3. PROPOSED SOLUTIONS

As previously described, the solutions presented in this work are designed to meet three requirements: generality, accuracy and efficiency. In this section, we present the general framework of solutions for robust spatial operations with uncertain coordinates, as well as the adaptations of the main heuristics used in each of the developed operations, specifically. In the PMCSJ, the filtering step, common in spatial join algorithms and important for reducing the number of exhaustive geometric calculations, is adapted to the PSJ by introducing the concept of the CR, which is an extension of MBR built to guarantee the required level of confidence when evaluating intersection probabilities. To meet the generalism requirement, the Chebyshev Inequality is applied to its derivation, since this inequality is true for a large family of PDFs - more specificaly any integrable random variable with finite expected value μ and finite non-zero variance σ^2. The details of its constrution is provided in Subsection 3.6. In the refinement step, we propose an adaptation of the MCM, denoted by PMCM, which helps our framework to keep user's desired balance between accuracy and efficiency, since it does not run a fixed number of simulations, but a sufficient number of batches of simulations. After running each batch it stops to evaluate whether the decision concerning objects intersection can be made with the statistical significance required by the user. Therefore, it avoids extra computational cost by runing a large number of unecessary simulations - improving efficiency - and keeps accuracy in standards controlled by the user. The parameter associated with accuracy is called *gamma* (γ) and represents the confidence level for claiming a match between two spatial objects. As Monte Carlo simulations may be performed with any probability distribution, it provides the generality we are persuing. Thus, its inclusion benefits our framework in this particular requirement. However, as they are computationally expensive, we introduce the PMCM to improve its efficiency allowing the control by users of the desired level of confidence in the results (accuracy, as provided by γ).

In the pSkyMCM, the MCM is used to evaluate the probabilities of dominance for any PDF (generality) and the direct application of a lemma presented by [Pei et al. 2007] is used to save computation (efficiency).

Subsection 3.1 discusses the general structure of solutions for spatial operations with imprecise geographical coordinates. Subsection 3.2 shows how the positional errors are simulated and Subsection 3.3 presents the proposed adaptation to the MCM in order to efficiently and effectively deal with the evaluation of probabilistic conditions. More specifically, the solution is able to decide the true value of the condition $q \leq p$, where p is a cutoff probability and q is a probability of success. In Subsections 3.4 and 3.5, the PMCM algorithm is applied to develop pSkyMCM. In Subsection 3.6, another contribution of our work is also presented, which increases the efficiency of spatial solutions with uncertainty in the coordinates: the concept of CR, as well as a proposal for the development of a CR applicable to the broad variety of probabilistic distributions for the positional errors.

3.1 General framework for robust spatial operations

Figure 5 presents our framework. It is composed of two stages: 1) PMCM and 2) probabilistic adaptations (probabilistic heuristics) for each spatial operation. PMCM is applicable for all spatial operations when evaluating a predicate. For example, in the case of spatial join, that predicate is "do objects A and B intercept?". In case of spatial skyline the predicate is "is object A not dominated by any other object?".

The need for using PMCM and not other procedures in competing solutions is to provide both a competitive standard in terms of efficiency and accuracy - while also working with datasets which exhibit the most unusual error patterns (generalist regarding error distribuion). The PMCM also allows developers to set which dimension is more important, efficiency or accuracy.

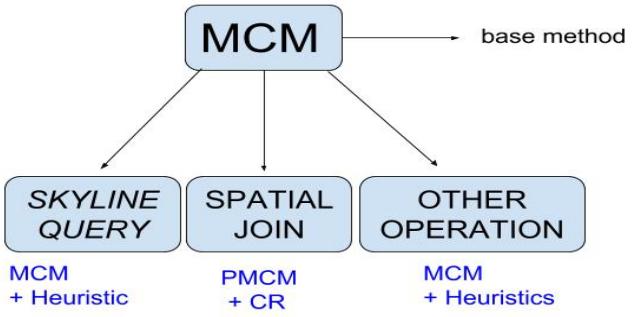

Fig. 5. General Framework Solution.

While each spatial operation has a specific predicate to be evaluated, it has also other specificities which need to be adapted as well in order to turn the operation probabilistic. Here we call those adaptations probabilistic heuristic. In the case of spatial join, it is the traditional filtering step applied in most algorithms for that operation. In this article, the operation becomes probabilistic with the introduction of the CR, where its formulation is consonant with the dimensions of generality, efficiency and accuracy. A theorem is proved to demostrate its applicability to the filtering stage of a probabilistic spatial join. Furthermore, empirical resuts in Section 4 indicate that it collaborates to the efficiency and accuracy results achieved by our proposed spatial join. In the case of spatial skyline, we use a lemma as probabilistic heuristic for that operation. With its use, some predicate evaluations are avoided, saving processing time (efficiency). The combined application of these mechanisms allows our solutions to reach satisfactory levels of generality, accuracy, and efficiency, as it will be demonstrated by the theoretical and empirical results in this section and in the next one.

Standard MCM provides generality for spatial solutions, since it allows them to be performed by any PDF associated with the errors in the coordinates. On the other hand, the MCM also considerably increases the computational cost of the algorithms. Thus, it is imperative to create procedures to avoid that the maximum cost associated with the MCM is reached. In this context, this article presents the PMCM. Along with PMCM, for each spatial operation, some heuristics need to be applied to provide efficiency for that particular computational task. In the case of the proposed PSJ, it is used the CR. PMCM and CR are responsible for decreasing the total number of simulations to be performed by our solutions and so save computation time.

3.2 Monte Carlo simulations

MCM refers to any method in statistical inference or numerical analysis that uses simulations, [Rizzo 2007]. They are known in the literature for their different contexts, such as obtaining estimates for defined integrals, mean of a function, calculation of probabilities, among others.

In this article, the MCM is used in two contexts: 1) to estimate the intersection probabilities between two geometries; 2) to estimate the dominance probability between two points. In order for the MCM to be properly applied, the simulations must represent the studied phenomenon well. The procedure applied to perform the simulations is described as follows. Each simulation consists of replacing a point $P = (x, y)$ of a geometry g with another coordinate Q that simulates its real position. After that, Q is obtained from the displacement of P in the direction of a vector v generated with an angle θ, such that $\theta \sim Uniform(0, 2\pi)$, and with a norm r generated according to the PDF that models the positional error.

Figure 6 illustrates the procedure used in the simulation. The choice of $Uniform(0, 2\pi)$ ensures that

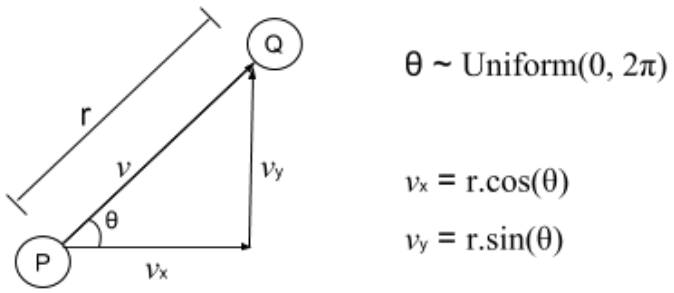

Fig. 6. Positional error simulation.

the error is evenly distributed in all directions. To compute the displacement vector v, its components v_x and v_y are obtained from the trigonometric relations: $v_x = r.cos(\theta)$ and $v_y = r.sen(\theta)$. Finally Q is obtained with the equation $Q = (x + v_x, y + v_y)$. This ensures that each point is shifted in a random direction to another point whose distance from the original is given by the chosen PDF.

The Monte Carlo estimate $\hat{q}$ of the probability of intersection q between two geometries - g and h - is obtained by the simulation of n displacements in g and h by means of formula $q = \frac{x}{n}$, where x is the number of simulations in which g and h intersect (that is an unbiased estimate for minimum variance for q). From inference theory, we know that $E(\hat{q}) = q$ and $Var(\hat{q}) = \frac{q}{1-q}$. In addition, $Var(q) \to 0$ when $n \to \inf$.

The estimate obtained is used to evaluate the condition: $q \leq p$. If this condition is true, the spatial join returns the match between the geometries involved. Otherwise, there is no match. However, since $\hat{q}$ is an estimate, there can be an error in the evaluation of this condition. The Central Limit Theorem guarantees that when n tends to infinity, $\hat{q}$ tends to q. Therefore, the assessment of the condition may be as accurate as desired, by simply applying a sufficiently large number of simulations. Yet, this number may be computationally impractical. The next subsection shows how the PMCM solves the practical problem of finding a sufficient number of simulations so that the condition can be evaluated accurately and in a computationally efficient way.

3.3 Monte Carlo method

The PMCM aims to set a procedure that applies the MCM in an accurate and efficient way. It consists of carrying out just the right number of simulations to evaluate the condition of interest satisfying the required accuracy constraint - in our case $q \leq p$, with the required level of confidence.

After performing the simulations, we can obtain the estimate $\hat{q}$ of the probability of intersection q between two geometries a and b, $a \in A$, $b \in B$. By fixing a number n of simulations, we obtain the margin of error ϵ for $\hat{q}$, i.e., the maximum distance $\hat{q}$ is from q with a confidence level γ. Therefore, it is known that the probability that q belongs to the interval $(q - \epsilon,\ q + \epsilon)$ is equal to γ. Therefore, based on this confidence interval, one can adopt the following rule:

(1) If $p < \hat{q} - \epsilon$, it is decided with confidence level γ that there is no match.
(2) If $p \leq \hat{q} + \epsilon$, it is decided with confidence level γ that there is a match.
(3) Otherwise, neither alternative is ensured with a level of confidence γ. Nevertheless, a decision can be made based on the true value of $(q \leq p)$ with a confidence level less than γ.

For reasons of efficiency, the number of simulations should not be too high. However, due to the accuracy constraint, it must not be too small either. For example, $n = 50$ may be sufficient to

guarantee a computation time below a threshold acceptable to the user, but insufficient to ensure that the match verdict is evaluated at the desired level of confidence. The reverse may also occur: $n = 50$ be more than necessary to ensure the desired accuracy, which would constitute a computation time waste.

PMCM solves this problem by controlling the number of Monte Carlo simulations used to evaluate if $(q \leq p)$. Instead of executing the maximum number allowed for the simulations for each pair of candidates, PMCM executes a smaller number $m < n$ in each step. Thus, if $n = 1000$, m can be 40, for example. A single batch of $m = 40$ may be sufficient to decide with the desired confidence level the true value of the condition $(q \leq p)$. In case that occurs, the procedure would save 960 simulations without affecting the solution's desired accuracy level. If not, a new batch of $m = 40$ would be executed and the resulting 80 simulations (40 in the first lot plus 40 in the second) would be used again to evaluate $(q \leq p)$. The procedure is repeated until either the condition can be evaluated with the specified level of confidence or the limit for the number of simulations is reached. The PMCM algorithm is presented below.

Algorithm 1: Progressive Monte Carlo method (PMCM).

Data: a e b – two geometries
p – cut probability
m – batch size
n_{max} – maximum number of simulations
Result: TRUE if the success proportion is greater than p, otherwise FALSE

1 Initialize the counter n with zero.
2 Shift the geometries a and b m times.
3 Update n in m units $(n \leftarrow n + m)$.
4 Compute the **success** proportion $\hat{q}$ in the n simulations.
5 Compute the CI for q, i.e,

$$CI(q, \gamma) = \left[\hat{q} - t_c\sqrt{\frac{\hat{q}(1-\hat{q})}{n}}, \hat{q} + t_c\sqrt{\frac{\hat{q}(1-\hat{q})}{n}}\right],$$

where t_c is the $(1+\gamma)/2$ quantile of t-Student distribution with $n-1$ degrees of freedom.
6 If $p \in CI(q, \gamma)$ and $n \leq n_{max}$, repeat steps 2-5.
7 If $\hat{q} \geq p$ return **TRUE**, else return **FALSE**.

PMCM executes batches of m simulations (line 2) whenever the maximum number of iterations is not reached and the desired accuracy guarantees it is not achieved. At the end of each batch, the CI for q with the confidence level γ is calculated using the formula for the confidence interval for a proportion provided by the theory of statistical inference (line 5). If $p \in CI(q, \gamma)$ (line 6), it is not possible to decide with a confidence γ the true value of the condition $q \leq p$. In this case, another batch of simulations is needed to reduce the CI size (go to line 2). Otherwise, the CI is already sufficient to decide the true value of the condition with the confidence level γ. Thus, the decision can be taken with the accuracy required by the user without the need for further simulations, saving computation time.

The CI presented in line 5 presents a margin of error $E = t_c\sqrt{\frac{\hat{q}(1-\hat{q})}{n}}$ for the estimated success rate, where t_c is the $(1+\gamma)/n$ quantile of the t-Student distribution. Each time new simulations are performed, the size of CI decreases - since the denominator n increases - to a point where it will be small enough to decide the true value of the condition $q \leq p$ with confidence level γ or reach the maximum number of iterations defined. The efficacy of the method in correctly assessing condition

($q \leq p$) is related to the value of E. Once a confidence level γ has been set, the smaller E is the greater the chance of the probability of cut p not belonging to the CI and, therefore, the greater the chance the decision by the true value of the condition $q \leq p$ will be accurately assessed.

The computation time for the evaluation of condition $q \leq p$ for a given pair of geometries increases linearly as a function of the number of simulations n, while the margin of error decreases proportionally to $\sqrt{n}$. Therefore, the computational cost in evaluating the condition $q \leq p$ grows quadratically relating the decrease factor k in which the margin of error is reduced. However, we emphasized that the need of a relatively high value k is directly associated with a smaller difference between p and q. In several practical scenarios, a difference low enough to requiring a too high value for k can be ignored by the user.

3.4 Probabilities in the skyline

Let $U = \{A_1, ..., A_r\}$ be a collection of georeferenced objects and $L = \{L1, ..., L_d\}$ be a collection of reference points. We wish to obtain the subset S_p of U, such that for every $B_i \in S_p$ the probability of B_i not being dominated by any of the points of U is at least equal to p. This subsection shows how to calculate the probability of a point A_i belonging to S_p, formally $Pr(A_i \in S_p)$. We have

$$Pr(A_i \in S_p) = Pr\left[\bigcup_{i=1;i\neq j}^{r} (A_i \prec A_j)^c\right], \tag{1}$$

where

—the symbol "$\prec$" denotes dominance, i.e., $A_i \prec A_j$ means that A_i is dominated by A_j.
—A_i^c represents the complement of A_i.

Assuming independence between events, one can write

$$Pr(A_i \in S_p) = \bigcup_{i=1;i\neq j}^{r} (A_i \prec A_j)^c, \tag{2}$$

in which $Pr(A_i \prec A_j)$ can be calculated by

$$Pr(A_i \prec A_j) = Pr\left[\bigcup_{k=1}^{d} A_{ik} \prec A_{jk}\right] = \bigcup_{k=1}^{d} Pr[A_{ik} \prec A_{jk}], \tag{3}$$

assuming independence between the dimensions and with $k = 1, ..., d$ going through each one of the distances for the reference points. For a given k in particular, $(A_{ik} \prec A_{jk})$ is equivalent to $(A_{jk} < A_{ik})$, since optimization occurs in the sense of minimization points of reference. Finally, $Pr(A_{jk} < A_{ik})$ is evaluated using the MMC for a number m of simulations, according to Equation 4.

$$Pr(A_{jk} < A_{ik}) = \#(A_{jk} < A_{ik})/m. \tag{4}$$

Lemma 1 below, presented and proven by [Pei et al. 2007], can be used to avoid high computational costs involved in Equation (2), mainly due to simulations required in Equation (4).

Lemma 1. Let $U = \{A_1, ..., A_r\}$ be a collection of objects with imprecise coordinates and S_p the p-skyline for the set U relative to any set of spatial attributes. If A_i dominates A_j, then $Pr(A_i \in S_p) > Pr(A_j \in S_p)$.

3.5 The pSkyMCM solution

The p-Skyline proposed in this work - pSkyMCM - uses in its algorithm the Lemma 1, Equations (1), (2), and the Monte Carlo method performed on the PDF used to model positional errors. The algorithm is presented in the following.

Algorithm 2: pSkyMCM

Data: S – a set of n georeferenced objects
R – a set of reference coordinates
p – cut probability
μ – mean error
σ – standard deviation of errors.
Result: S_p – a subset of S of the objects whose probabilities of being Pareto efficient concerning the derived attributes from R is at least p

```
Compute the distance from each point to the reference coordinates, resulting in a matrix n × d,
  where n is the number of points, d is the number of reference coordinates.
Initialize the arrays P and Q, which will, respectively, store the points already known to be
  Pareto efficient and those already known to be not.
for each point i do
    Verify if i belongs to either P or Q. If yes, increment i and repeat the verification. If not,
      follow the next steps.
    Compute the probability of i being dominated by j for each j = 1, ..., n.
    Estimate the probability q of i not being dominated by any other point.
    if q ≥ p then
        include i in P and apply Lemma 1 to also include in P all points j such that j
          dominates i in a deterministic way.
    end
    else
        include i in Q and apply Lemma 1 to also include in Q all j points dominated by i in a
          deterministic way.
    end
end
Return P.
```

3.6 The PMCSJ solution

As the main proposals in the spatial deterministic join, PMCSJ also presents two classic steps: filtering and refinement. However, to meet the three requirements for which the solution is designed, these steps need to be adapted. The filtering step is run in a probabilistic and extended version of the MBR that we call the CR, commonly used to index spatial data. The refinement phase is performed by the PMCM algorithm presented in Subsection 3.3.

Definition 1: Given a geometry g and a cut probability p, a CR for g with a cut probability p is the one containing the MBR of g and, in addition, the probability of containing the real object represented by g is at least $\sqrt{1-p}$.

A first consequence of Definition 1 is that a CR is not unique as there are infinite rectangles satisfying the definition. However, we build one that is valid for a large family of probability distributions and

has the smallest size necessary to ensure the effectiveness of the filtering step. A smaller CR provides greater filtering power, since it will prevent a larger number of pairs from being evaluated in the refinement stage using PMCM, reducing the computational cost. Figure 7 shows the geometry g, its smallest surrounding rectangle (SSR) and the displacement value d. Another consequence of the definition is expressed in the following theorem.

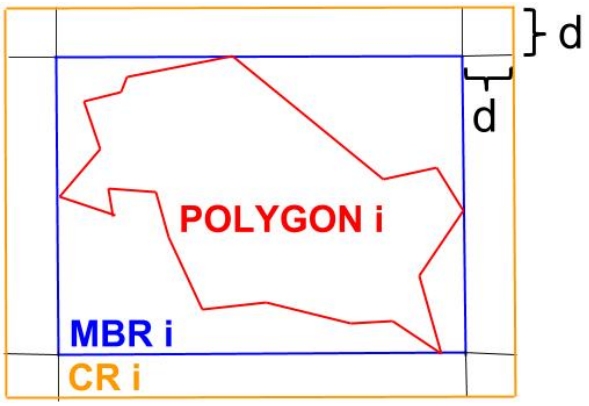

Fig. 7. Confidence Rectangle.

Theorem 1: Let G and H be confidence rectangles for geometries g and h, respectively. If G and H do not intersect, the probability of intersection between g and h is less than p.

Proof: Given that G and H are CRs for g and h then it follows from the definition that $Pr(g \subset G) \leq \sqrt{1-p}$ and $Pr(h \subset H) \leq \sqrt{1-p}$. Thus, assuming the independence concerning the error direction in g and h coordinates, $Pr(g \subset G, h \subset H) = Pr(g \subset G).Pr(h \subset H) \leq \sqrt{1-p}\sqrt{1-p} = 1-p$. Therefore, $Pr(g \subset G \cup h \subset H) < p$. Since G and H do not intersect, g and h only have a chance of intersection if at least one of them is not contained in its respective CR. However, the last equation shows that this probability is less than p. Thus, the probability of intersection between g and h is less than p as we wanted to demonstrate.

To construct a CR for a given geometry g, its MBR is expanded by a factor d in both vertical and horizontal directions. To meet the definition of CR, the value of d will be provided by Chebyshev's inequality. This inequality is valid for any integrable PDF, [Ross 2014], which meets the generality requirement for the present work. Chebyshev's inequality states that: if X is a random integer with finite mean $\mu = E(X)$ and standard deviation σ, then for any $k > 0$,

$$Pr(|X - E(X)| > k\sigma) < 1/k^2.$$

A consequence of this expression is that, for $X > 0$,

$$Pr(X - E(X) > k\sigma) < 1/k^2 \Rightarrow Pr(X \leq E(X) + k\sigma) \geq \frac{k^2-1}{k^2}.$$

In PMCSJ, X is the positional error (positive). Consequently, the probability of the error be at most $d = E(X) + k\sigma$ is at least $\frac{k^2-1}{k}$. By the Definition 1, in order to build a confidence rectangle, it is sufficient to take d such that $Pr(X \leq E(X) + k\sigma) > \sqrt{1-p}$. Now we can set

$$\sqrt{1-p} = \frac{k^2-1}{k^2} \Rightarrow k = \frac{1}{1-\sqrt{1-p}}.$$

Therefore, for a specific p,

$$d = E(X) + \sigma \frac{1}{\sqrt{1 - \sqrt{1-p}}}.$$

The CR coordinates are given by the following coordinates: $P_{min} = (x_{min}, y_{min})$ and $P_{max} = (x_{max}, y_{max})$, with $x_{min} = x_{SSR_{min}} - d$, $y_{min} = y_{SSR_{min}} - d$, $x_{max} = x_{SSR_{max}} + d$ and $y_{max} = y_{SSR_{max}} + d$.

The two datasets are indexed using an R-tree with the CRs assuming the same role as MBRs in a traditional spatial join solution (see [Patel and DeWitt 2000] for more details on indexing spatial objects). For two pairs of geometries a and b, $a \in A$, $b \in B$, the tree is traversed and in its descent, if the CRs do not intersect then the pair (a, b) is already discarded from the join result. The pairs in which CRs intersect are evaluated using the proposed refinement, i.e., PMCM.

4. EXPERIMENTAL EVALUATION

4.1 Probabilistic skyline

PSkyMCM was evaluated with a 10% cut probability for data of 36 schools from the city of Goiânia/Brazil, which the locations are shown in Figures 8 and 9. The schools' names were omitted and are identified by numbers for the sake of anonymity. PSkyMCM is performed in order to find schools that are closest to three locations of interest, indicated with an asterisk in these figures. These locations may correspond, for example, to the residence, place of work or university of a given person. PSkyMCM returns schools that are as close as possible to the three places and discards those that are dominated in that dataset. This scenario illustrates how PSkyMCM can be applied to facility recommendation based on multiple criteria.

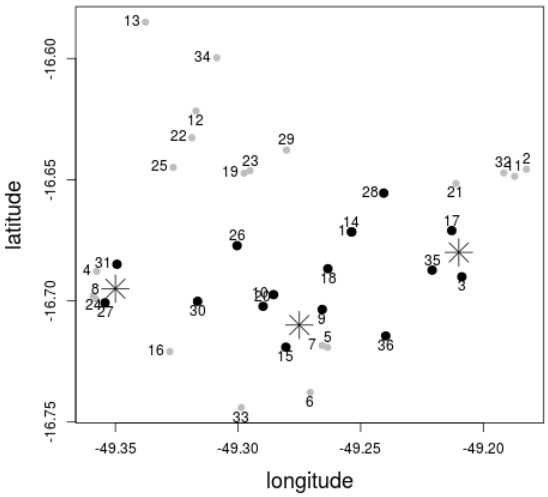

Fig. 8. Deterministic skyline query result.

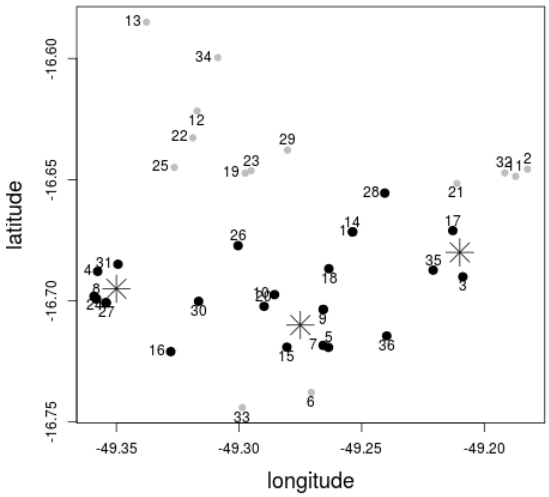

Fig. 9. *p-skyline* result with $p = 0.10$.

The error considered in the experiment follows a Normal distribution with a mean of 100 meters and a standard deviation of 1000 meters. These values are commonly found in real data obtained through a geocoding process. The three points considered are in the coordinates: $P_1 = (-16.68; -49.21)$, $P_2 = (-16.71; -49.275)$ and $P_3 = (-16.695; -49.35)$. Figure 8 shows the result obtained by the

deterministic skyline query and Figure 9 presents the result of PSkyMCM with $p = 0.10$ for the 36 schools. The obtained results were $S = \{1, 3, 9, 10, 14, 15, 17, 18, 20, 26, 27, 28, 30, 31, 35, 36\}$ for the deterministic skyline and $S_p = \{1, 3, 4, 5, 7, 8, 9, 10, 14, 15, 16, 17, 18, 20, 24, 26, 27, 28, 30, 31, 35, 36\}$ for the PSkyMCM. Thus, S_p contains all elements in S and also the points 4, 5, 7, 8, 16, and 24. This implies an increase of 37.5% in the number of available options. Since there is no strong probability guarantee (with confidence level of γ) that those extra points should be removed, the presented solution keep them as options. This avoid potentially good candidates to be wrongly eliminated for a specific query. In addition, if we assume that the 22 schools returned in S_p have about the same chance of being the most useful to the user, then the 6 additional schools in S_p (not in S) would represent a risk of $\frac{6}{22} = 27,27\%$ of the deterministic skyline not returning the best option. PSkyMCM prevents potentially useful options to be unnecessarily discarded and is therefore suitable for facility recommendation for data with inaccurate geographical coordinates. The methods to perform a p-skyline, such as those proposed by [Pei et al. 2007] are not applicable to this context, since they do not model the positional error of spatial objects.

4.2 Probabilistic spatial join

To test the probabilistic spatial join, three sets of data will be used: 1) green areas, 2) deforestation, and 3) wildfire areas, all covering the territory of the State of Goias, in Brazil. Figures 10 and 11 show the data layers of green areas, deforestation, and wildfire areas in the State of Goias. In Figure 10, the deforestation and wildfire area geometries can hardly be perceived. This illustrates how big can be the scale differences between the spatial objects represented in the datasets. In Figure 11, the wildfire areas and deforestation are more visible, as are the intersections with the green areas. Some of these intersections may be a consequence of the positional error in the data layer. Conversely, due to the imprecision of the data, some geometries that are not being intercepted in the figure may correspond to spatial objects that intersect in the reality.

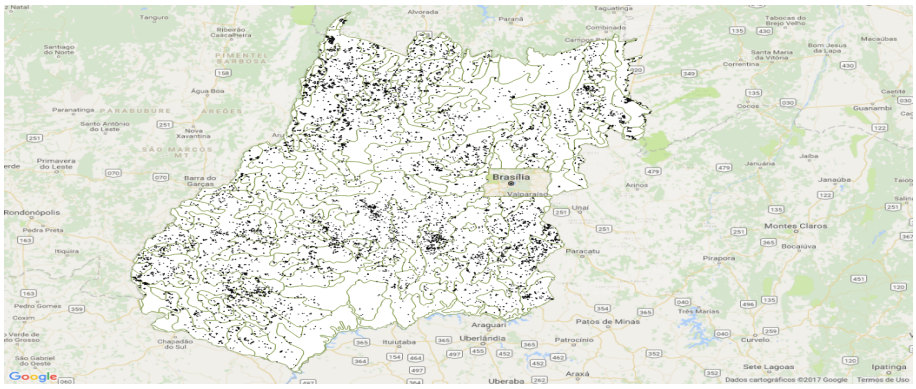

Fig. 10. Deforestation and wildfire at a province zoom level.

In order to judge the results obtained by the PSJs, a set of reference intersecting probabilities - we will call REF - will be used. The REF set is composed of the estimates of probability of intersecting each pair of geometries (a, b), $a \in A$ and $b \in B$, where A and B are the geometry pairs evaluated at the spatial join. In order to obtain such estimates, $N = 500$ Monte Carlo simulations were performed to evaluate the spatial join in each possible pair (a, b). The margin of error E_0 for the estimated intersection probability q_0 for a given pair of geometries (a, b) is given by $E_0 = z_c\sqrt{\frac{q_0(1-q_0)}{500}}$. Assuming a 95% level of confidence, the highest value that E_0 can assume is 0.0447 when q_0 is 0.5, and is as low as 0.023 when q_0 approaches more extreme values such as 0.1 or 0.9. The Subsection 4.3 discusses in greater detail how REF is used to evaluate the PSJs.

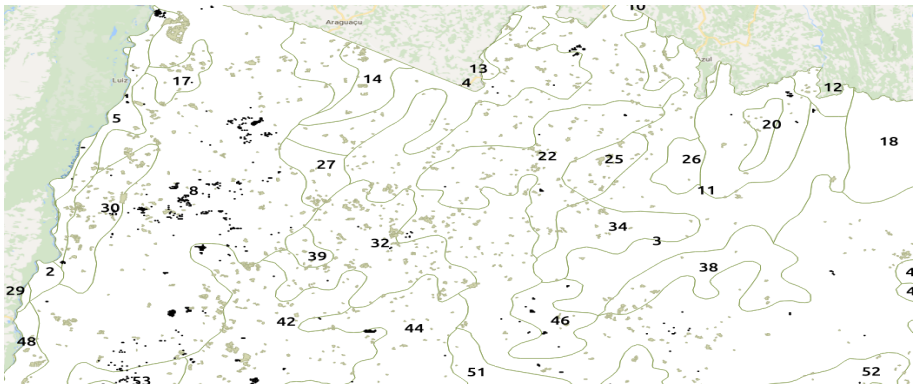

Fig. 11. Deforestation and wildfire at a county zoom level.

The solutions evaluated in this work are: Random Space Junction (RSJ) as proposed by [Openshaw 1989] with $m = 150$ simulations, and our approach (PMCSJ) that is executed with two configurations: 1) with $n_{max} = 150$, $m = 50$ simulations per step and confidence level $\gamma = 0.99$, which provides a confidence of 99% in the predicate evaluation; 2) $n_{max} = 1000, m = 50$. In the first configuration, we evaluate how PMCSJ compares to RSJ for the same number of simulations (in this case, 150). In the second configuration, we evaluate how PMCSJ compares to RSJ when we allow the number of PMCSJ simulations to be relatively high. In addition, we present a discussion in the end of the next subsection section on the Circular Normal Space Join (CNSJ) presented by [Ni et al. 2003].

4.3 Evaluation metrics

The solutions will be compared in a scenario with errors Y following a half-normal distribution, defined by $Y = |X|$, with $X \sim N(200, 100^2)$. The parameters of mean and standard deviation (required for RC computation) of this distribution are: $E(X) = \sigma\pi^2$ and $sd(X) = \sigma\sqrt{1 - \frac{2}{\pi}}$. The cutoff probabilities tested are: 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, and 0.90. The precision parameter is $\gamma = 0.99$, the size of the simulation batches is m = 50, and the maximum number of simulations is $n_{max} = 150$. Before presenting the metrics used in the comparison, some concepts must be defined.

First, it will be called the neighborhood of radius R around p and denoted by $VR(p)$, the interval $(p - R, p + R)$. Figure 12 shows the intervals used to compare the solutions, namely, for each value of p, the interval $(p - R, p + R)$ without the margin of error interval $(p - \epsilon, p + \epsilon)$ around p. The exclusion of the margin of error is applied to prevent the solutions from being evaluated within a radius of p for which there is not sufficient confidence in the sign of the expression $(q - p)$, where q is the probability of intersection between the two geometries evaluated. For example, the estimated probability $\hat{q}$ of intersection may be greater than p when the real probability q is less than p. In this scenario, a legitimate true positive can be considered a true negative in the set of reference adopted to evaluate the solutions. It is defined as the left neighborhood of p with radius R, the interval $(p-R, p)$. Similarly, $(p, p + R)$ is called the neighborhood on the right. The proportion of false negatives F_N in $VR(p)$ is defined as $FN = \frac{\#(-\infty,p)_{TEC}}{\#(p,p+r)_{REF}}$ and the proportion of false positives F_P in $VR(p)$ as $F_P = \frac{\#(p,\infty)_{TEC}}{\#(p-R,p)_{REF}}$, where

—$\#(a, b)_{TEC}$ is the number of geometry pairs whose intersection probabilities estimated by the PSJ technique lies in the interval (a, b).

—$\#(a, b)_{REF}$ is the number of geometry pairs whose intersection probabilities estimated by REF lies in the interval (a, b).

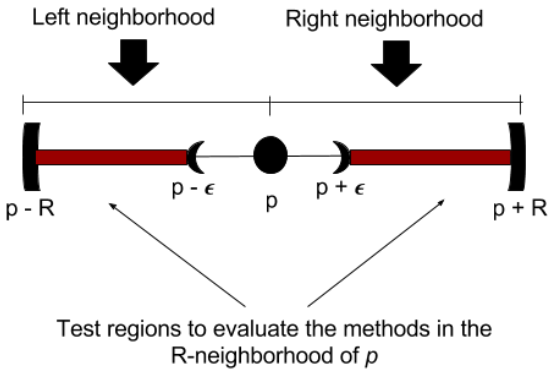

Fig. 12. Test regions.

As mentioned before, within the margin of error around p (see Figure 12), the probability that the REF estimate is on the wrong side with respect to the p, i.e., to the right when in fact it is on the left and vice versa - it is not negligible. For that reason, only those pairs whose probabilities are in the safe neighborhood region, shown with a stronger dash in Figure 12, are used to calculate the proportions of false negatives and positives - S_{FP} and S_{FN}, given by the expressions $S_{FN} = \frac{\#(-\infty,p)_{TEC}}{\#(p+\epsilon,p+R)_{REF}}$ and $S_{FP} = \frac{\#(p,\infty)_{TEC}}{\#(p-R,p-\epsilon)_{REF}}$.

4.4 Results and discussion

Table I presents results regarding the S_{FN} and S_{FP} metrics. The first column provides the metric and the cut probability (in parentheses) used. The RSJ-150, PMCSJ-150 and PMCSJ-1000 columns show the values obtained in each metric by RSJ with 150 simulations and PMCSJ with the limit of 150 and 1000 simulations respectively.

metric	RSJ-150	PMCSJ-150	PMCSJ-1000	signal 1	signal 2
$S_{FN}(0.10)$	0.200	0.000	0.000	+	+
$S_{FN}(0.20)$	0.000	0.056	0.000	-	
$S_{FN}(0.30)$	0.000	0.000	0.000		
$S_{FN}(0.40)$	0.000	0.000	0.000		
$S_{FN}(0.50)$	0.000	0.333	0.167	-	-
$S_{FN}(0.60)$	0.000	0.000	0.000		
$S_{FN}(0.70)$	0.000	0.111	0.000	-	
$S_{FN}(0.80)$	0.133	0.200	0.000	-	+
$S_{FN}(0.90)$	0.000	0.000	0.000		
$S_{FP}(0.10)$	0.000	0.000	0.000		
$S_{FP}(0.20)$	0.000	0.045	0.000	-	
$S_{FP}(0.30)$	0.000	0.077	0.000	-	
$S_{FP}(0.40)$	0.100	0.100	0.000		+
$S_{FP}(0.50)$	0.000	0.000	0.000		
$S_{FP}(0.60)$	0.000	0.000	0.000		
$S_{FP}(0.70)$	0.200	0.000	0.000	+	+
$S_{FP}(0.80)$	0.000	0.000	0.000		
$S_{FP}(0.90)$	0.100	0.000	0.000	+	+

Table I. Accuracy comparison of RSJ and PMCSJ with signals indicating whether PMCSJ was more accurate (or not) than RSJ for $n.max = 150$ and $n.max = 1000$, respectively.

PMCSJ benefits from not having to run all set simulations to be able to accurately assess the condition of interest. Thus, PMCSJ saves most of the time RSJ would require to evaluate the predicate for each pair of geometries. However, as the stopping criterion for simulations is stochastic, it may occur that in some cases the algorithm wrongly judges that the number of existing simulations is sufficient to evaluate the join predicate with confidence. This risk that can be minimized by increasing the value of the parameter γ at the price of potentially making the algorithm slower, but not that much since filtering step avoid most of the evaluations. Table I shows that for limit 150, RSJ won in six scenarios and PMCSJ in three. However, when compared to PMCSJ with a limit of 1000 RSJ loses in five out of six scenarios.

To test whether the difference observed in the RSJ-150 versus PMCSJ-150 is statistically significant or can be attributed to chance, a hypothesis test is performed – the signal test. The signal column of Table I shows the 18 observations paired with a signal: "+" if PMCSJ performed better than RSJ and "-" if performed worse. In the case of a tie, no signal is emitted. The two statistical hypotheses – null (H_0) and alternative (H_1) – which we consider in the test are:

—"H_0: PMCSJ is as effective as RSJ" (the probability of "+" is equal to "-";
—"H_1: PMCSJ is less effective than RSJ" (the probability of "+" is less than "-").

The statistic test t is: "number of +". Only untied cases are considered in this type of test, resulting in a total of $n = 9$ cases. Under the H_0 assumption, t follows a binomial distribution with $n = 9$ trials and probability of success $p = 0.5$. One way to assess whether the observed value t_{obs} of the t statistic corroborates for or against H_0 goes through the calculation of the p-value or descriptive value associated with t_{obs}. The p-value associated with the observed value of a statistic is the probability of obtaining an equal or more extreme value for it when H_0 is true. In other words, the p-value provides the probability that a value is as or more extreme than the observer could have been observed if H_0 is true.

A "low" p-value indicates that H_0 is probably false. Typically, a p-value less than 0.05 is considered low enough to reject the hypothesis H_0 in favor of H_1, because in this case the chances that such an extreme value could be simply the work of chance would be less than 1 in 20. Since $t \sim Binomial(9, 0, 5)$ under H_0, we have that p-value is given by $Pr(t \leq 3|H_0) = 0,254$. Thus, the scenario 6 against 3 does not provide sufficient evidence to reject H_0 even with a level of confidence of 0.80. In fact, if PMCSJ-150 and RSJ-150 are equally effective, it is expected that the observed difference occurs 1 in 4 experiments. Therefore, there is no sufficient evidence – even at the confidence level of 80% – that points to a significant difference between the efficacy of both solutions. This result is in line with the probabilistic assurance offered by Algorithm 1 of Section 3.

Figure 13 shows the difference in computation time for RSJ (c) with 150 simulations and PMCSJ with 150 (a) and 1000 (b) simulations. The hardware used in the execution of both solutions was an Intel Core i5-4200U, 1.6GHz CPU and with 4 threads in parallel. PMCSJ-150 spent an average of 64 seconds to perform deforestation/green areas join against 73 of PMCSJ-1000 and 257 of RSJ-150. Regarding the wildfire/green areas join, the computation time was 35 for PMCSJ-150, 40 for PMCSJ-1000, against 92 for RSJ-150. In both scenarios, PMCSJ presented lower computation times than RSJ. It is noted that the computation time increased by only 14% when increasing the simulation limit in PMCSJ from 150 to 1000. This happens because most predicate evaluations for a given pair of geometries can be revolved with the desired confidence level using either a few batches of m = 50 or the stochastic filtering step. RSJ, on the other hand, has its computation time impacted linearly with the increase in the number of simulations.

In addition, as Table I points out, PMCSJ-1000 presented five favorable results in six possible against RSJ-150. Under the null hypothesis of equivalent efficacy between PMCSJ-1000 and RSJ-150, we have a p-value of 0.11 relative to a unilateral test to test whether PMCSJ-1000 performed better than RSJ-150. Note that the p-value was lower for this test than the previous one. Given the

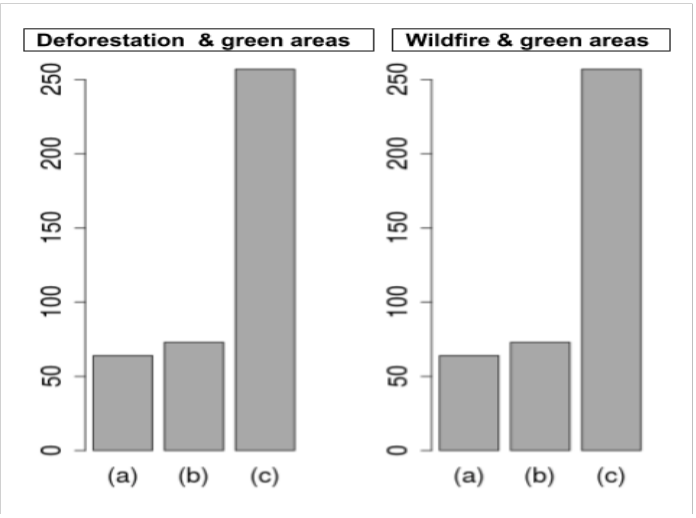

Fig. 13. Computation time: (a) PMCSJ-150; (b) PMCSJ-1000; (c) RSJ-150.

p-value obtained, we have that case PMCSJ-1000 and RSJ-150 possess the same accuracy, the chance of observing at least 5 of 6 cases in favor of PMSCJ would be 1 in 9 experiments. In summary, it is noticed that, concerning the obtained results: 1) one can increase the effectiveness of PMCSJ in order to make it as effective as RSJ by simply increasing the maximum number of simulations allowed by the program; 2) increasing the maximum number of simulations does not drastically impact performance of PMCSJ, allowing it to still perform significantly better than RSJ.

4.5 Qualitative analysis

The CNSJ solution, [Ni et al. 2003], was designed to be efficient, but it is effective just under the Circular Normal assumption for the errors and reasonably effective for error distributions whose behaviour does not diverge much from a Normal distribution. The PMCSJ proved to achieve satisfactory performance in the three requirements in such a way that it is so generalist as RSJ while being more efficient and effective than it. Furthermore, PMCSJ possess almost the same effectiveness that CNSJ does for Circular Normal distribution, with the advantage of being applicable to other PDFs.

The decision for the best solution to a given real case can be following way. If the errors exhibit a pattern that can be modeled by the Circular Normal distribution, choose CNSJ to obtain the best possible accuracy and efficiency (specialist solution). If the application requirements in terms of the computation time is very restrictive (perhaps due to the large volume of data) and error distribution is not too far from the Normal one, being at least symmetric, then the CNSJ is also a good option to meet the requirement imposed by the restriction of science, without impairing the accuracy. However, if the distribution is definitely not Circular Normal or at least one symmetric that does not present serious deviations from, then PMCSJ is the recommended solution. In addition, if the requirement for computational performance is not rigid, PMCSJ is also indicated to obtain a more accurate solution than CNSJ even for distributions similar to the Circular Normal distribution.

5. CONCLUSIONS

We presented a general framework of solutions for spatial operations on uncertain positional data and evaluated two operations as a case study: Spatial Skyline with Monte Carlo Method (PSkyMCM) and Progressive Monte Carlo Spatial Join (PMCSJ). In order for these solutions to perform accurately

and efficiently, it was necessary to adapt the heuristic used in the classical spatial operations to the probabilistic case. This required the development of two new techniques that are contributions of our work: the confidence rectangles and the Progressive Monte Carlo Method.

The results of PSkyMCM pointed out that there is a significant risk of not returning a potentially useful solution to the user if a deterministic skyline query is performed instead of a probabilistic one. The Monte Carlo simulations and heuristics used to avoid the massive use of them enabled skyline queries to be more effective for the scenario where imprecise coordinates are present. The experiments showed that the PMCSJ is: i) generalist in relation to the distribution of positional errors; ii) accurate; and iii) efficient. PMCSJ enables applications to be built to handle large datasets returning an accurate response for a variety of error patterns with a comparatively low computation time.

As future work, we envisage the opportunity for improving the filtering stage. As a positive collateral effect, the accuracy would also improve since the time saved could support a more expensive filtering step involving higher values for the accuracy parameter γ and the superior limit for simulations n_{max}. The two steps of the spatial join can also be adapted to the specificities of a probability distribution family, for example, the exponential family. This could improve efficiency and accuracy, with a some impact on the generality.

REFERENCES

Arcaini, P., Bordogna, G., and Sterlacchini, S. Flexible querying of volunteered geographic information for risk management. In *8th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-13)*. Atlantis Press, 2013.

Arge, L., Procopiuc, O., Ramaswamy, S., Suel, T., and Vitter, J. S. Scalable sweeping-based spatial join. In *VLDB*. Vol. 98. Citeseer, pp. 570–581, 1998.

Borzsony, S., Kossmann, D., and Stocker, K. The skyline operator. In *Data Engineering, 2001. Proceedings. 17th International Conference on*. IEEE, pp. 421–430, 2001.

Brinkhoff, T., Kriegel, H.-P., and Seeger, B. *Efficient processing of spatial joins using R-trees*. Vol. 22. ACM, 1993.

Chomicki, J., Ciaccia, P., and Meneghetti, N. Skyline queries, front and back. *ACM SIGMOD Record* 42 (3): 6–18, 2013.

Chomicki, J., Godfrey, P., Gryz, J., and Liang, D. Skyline with presorting. In *ICDE*. Vol. 3. pp. 717–719, 2003.

Chomicki, J., Godfrey, P., Gryz, J., and Liang, D. Skyline with presorting: Theory and optimizations. In *Intelligent Information Processing and Web Mining*. Springer, pp. 595–604, 2005.

Dai, X., Yiu, M. L., Mamoulis, N., Tao, Y., and Vaitis, M. Probabilistic spatial queries on existentially uncertain data. In *International Symposium on Spatial and Temporal Databases*. Springer, pp. 400–417, 2005.

de Oliveira, W. B., de Oliveira, S. S. T., do Sacramento Rodrigues, V. J., dos Santos, H. S. B., and Cardoso, K. V. A method for location recommendation via skyline query tolerant to noised georeferenced data. *Revista Brasileira de Cartografia* 68 (6), 2015.

Ding, X., Jin, H., Xu, H., and Song, W. Probabilistic skyline queries over uncertain moving objects. *Computing and Informatics* 32 (5): 987–1012, 2014.

Faure, E., Danjou, A. M., Clavel-Chapelon, F., Boutron-Ruault, M.-C., Dossus, L., and Fervers, B. Accuracy of two geocoding methods for geographic information system-based exposure assessment in epidemiological studies. *Environmental Health* 16 (1): 15, 2017.

Guttman, A. *R-trees: a dynamic index structure for spatial searching*. Vol. 14. ACM, 1984.

Huang, Z., Lu, H., Ooi, B. C., and Tung, A. K. Continuous skyline queries for moving objects. *Knowledge and Data Engineering, IEEE Transactions on* 18 (12): 1645–1658, 2006.

Hughes, W. Wide-area augmentation system performance analysis report. *Federal Aviation Administration WAAS Test Team, Atlantic City, NJ*, 2002.

Jacox, E. H. and Samet, H. Iterative spatial join. *ACM Transactions on Database Systems (TODS)* 28 (3): 230–256, 2003.

Khalefa, M. E., Mokbel, M. F., and Levandoski, J. J. Skyline query processing for incomplete data. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. IEEE, pp. 556–565, 2008.

Kossmann, D., Ramsak, F., and Rost, S. Shooting stars in the sky: An online algorithm for skyline queries. In *Proceedings of the 28th international conference on Very Large Data Bases*. VLDB Endowment, pp. 275–286, 2002.

Lee, M.-W., Son, W., Ahn, H.-K., and Hwang, S.-w. Spatial skyline queries: exact and approximation algorithms. *GeoInformatica* 15 (4): 665–697, 2011.

Ljosa, V. and Singh, A. K. Top-k spatial joins of probabilistic objects. In *2008 IEEE 24th International Conference on Data Engineering*. IEEE, pp. 566–575, 2008.

Lo, M.-L. and Ravishankar, C. V. Spatial joins using seeded trees. In *ACM SIGMOD Record*. Vol. 23. ACM, pp. 209–220, 1994.

Lofi, C., El Maarry, K., and Balke, W.-T. Skyline queries in crowd-enabled databases. In *Proceedings of the 16th International Conference on Extending Database Technology*. ACM, pp. 465–476, 2013.

Luo, G., Naughton, J. F., and Ellmann, C. J. A non-blocking parallel spatial join algorithm. In *Data Engineering, 2002. Proceedings. 18th International Conference on*. IEEE, pp. 697–705, 2002.

Mishra, P. and Eich, M. H. Join processing in relational databases. *ACM Computing Surveys (CSUR)* 24 (1): 63–113, 1992.

Ni, J., Ravishankar, C. V., and Bhanu, B. Probabilistic spatial database operations. In *International Symposium on Spatial and Temporal Databases*. Springer, pp. 140–158, 2003.

Openshaw, S. Learning to live with errors in spatial databases. *Accuracy of spatial databases*, 1989.

Papadias, D., Tao, Y., Fu, G., and Seeger, B. An optimal and progressive algorithm for skyline queries. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM, pp. 467–478, 2003.

Patel, J. M. and DeWitt, D. J. Clone join and shadow join: two parallel spatial join algorithms. In *Proceedings of the 8th ACM international symposium on Advances in geographic information systems*. ACM, pp. 54–61, 2000.

Pei, J., Jiang, B., Lin, X., and Yuan, Y. Probabilistic skylines on uncertain data. In *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, pp. 15–26, 2007.

Rizzo, M. L. *Statistical computing with R*. CRC Press, 2007.

Ross, S. *A first course in probability*. Pearson, 2014.

Sharifzadeh, M. and Shahabi, C. The spatial skyline queries. In *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, pp. 751–762, 2006.

Son, W., Hwang, S.-w., and Ahn, H.-K. Mssq: Manhattan spatial skyline queries. *Information Systems* vol. 40, pp. 67–83, 2014.

Son, W., Lee, M.-W., Ahn, H.-K., and Hwang, S.-W. Spatial skyline queries: an efficient geometric algorithm. In *Advances in Spatial and Temporal Databases*. Springer, pp. 247–264, 2009.

Tan, K.-L., Eng, P.-K., Ooi, B. C., et al. Efficient progressive skyline computation. In *VLDB*. Vol. 1. pp. 301–310, 2001.

Tu, J., Del Amo, A., Xu, Y., Guari, L., Chang, M., and Sebastian, T. A fuzzy bounding box merging technique for moving object detection. In *2012 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*. IEEE, pp. 1–6, 2012.

You, G.-w., Lee, M.-W., Im, H., and Hwang, S.-w. The farthest spatial skyline queries. *Information Systems* 38 (3): 286–301, 2013.