Wildfire burn scar encapsulation

Subsetting common spatial domains for post-wildfire debris flow predictions over the United States.

Jorge A. Duarte¹ · Andrés D. González² · Jonathan J. Gourley³

Received: 2 September 2020 / Accepted: 25 August 2021 / Published online: 30 October 2021
© The Author(s) 2021

Abstract

Wildfires burn annually across the United States (US), which threaten those in close proximity to them. Due to drastic alterations of soil properties and to the land surfaces by these fires, risks of flash floods, debris flows, and severe erosion increases for these areas, which can have catastrophic consequences for biota, people and property. Computational tools, such as the WildfireRain algorithm, have been designed and implemented to assess the potential occurrence of debris flows over burn scars. However, in order to efficiently operate these tools, they require independent, non-overlapping buffers around burned areas to be defined, which is not a trivial task. In this paper we consider the problem of efficiently subsetting the conterminous US (CONUS) domain into optimal subdomains around burn scars, aiming to enable domain-wide WildfireRain product outputs to be used for operations by the National Weather Service (NWS). To achieve this, we define the Object Encapsulation Problem, where burn scars are represented by single-cell objects in a gridded domain, and circular buffers must be constructed around them. We propose a Linear Programming (LP) model that solves this problem efficiently. Optimal results produced using this model are presented for both a simplified synthetic data set, as well as for a subset of burn scars produced by severe wildfires in 2012 over the CONUS.

✉ Jorge A. Duarte
jduarte@ou.edu

Andrés D. González
andres.gonzalez@ou.edu

Jonathan J. Gourley
jj.gourley@noaa.gov

- ¹ Cooperative Institute for Mesoscale Meteorological Studies, National Weather Center, 120 David L. Boren Blvd., Norman, OK 73072, USA
- ² School of Industrial and Systems Engineering, University of Oklahoma, 202 W. Boyd St., Norman, OK 73019, USA
- ³ NOAA National Severe Storms Laboratory, National Weather Center, 120 David L. Boren Blvd., Norman, OK 73072, USA

Keywords Wildfires · Debris Flows · Domain Subsetting · Object Encapsulation Problem

1 Introduction

Dozens of large wildfires burn across the United States (US) every year leaving behind burned landscapes where fire has consumed most of the vegetation. Aside from the immediate threat these fires present for natural ecosystems, animals, property and people, the soil properties in these burned areas can be significantly altered. Particularly, soils can become more repellent to water and the removal of vegetation increases soil erosion and sediment delivery [6,9,22,24,25]. These processes lead to increasing risks of flash floods, severe erosion, and debris flows for local communities downstream from the affected areas [5]. Given that wildfires are increasing in size, duration, and intensity, and so has the wildland-urban interface [31], risk for wildfire effects on society can be expected to rise as well [12].

Tools to assess the potential danger and likelihood of debris flows over wildfire-affected areas have been designed and implemented in operational weather forecasting environments. In particular, the *WildfireRain* algorithm was developed as a part of the Flooded Locations and Simulated Hydrographs (FLASH) project [11], to assess whether incoming rainfall is likely to initiate debris flows over wildfire burn scars (areas of burned land and vegetation). This algorithm automatically ingests weather radar-based rainfall intensities, accumulates them, and compares them to debris flow initiation thresholds, which are specific for a given burn scar. These debris flow thresholds are estimated by the U.S. Geological Survey (USGS) using a logistic regression approach based on the surveyed soil's burn severity, upstream slope, and soil erodibility factors [26]. Because of this, debris flow initiation thresholds are unique to every burn scar. These thresholds represent the estimated amount of accumulated rainfall that would be needed to produce at least a 50% probability of debris flow for a given burn scar, over the past 15, 30 and 60min of precipitation. Lower thresholds are associated with more recent, larger, more intense fires, and the opposite is true for higher thresholds. Normally, this logistic regression is run immediately following the containment and assessment of a wildfire, and then thresholds are made available by the USGS to the stakeholders. Given that burn scars tend to recover as time passes, burn scars are surveyed periodically by the USGS, and their thresholds are usually recalculated as they *heal* (or burn again) over time.

The WildfireRain algorithm ingests and accumulates real-time radar-estimated rainfall rates, for (past) durations of 15, 30, and 60 min, in concordance with how the debris flow thresholds have been defined. For every time step, it also applies a steady-state assumption for assumed *future* rainfall rates, which allows it to provide short-term forecast accumulations for rainfall intensity. These forecast and estimated rainfall accumulations are then compared to the thresholds for debris flow initiation on a pixel-by-pixel basis for a specific burn scar. Ultimately, each rainfall pixel gets classified as light rain (below threshold), debris flow danger (estimated rainfall rates exceed debris flow thresholds), or debris flow likely (forecast rainfall rates exceed debris flow thresholds). These classifications are aimed to provide weather forecasters insight

Fig. 1 Example of WildfireRain algorithm outputs. Notice the color-coded rainfall (entering over the coast), the burn area (inland) presented in blue, and the subdomain (buffer) around it (square window where data is present)

regarding how potentially hazardous neighboring or incoming precipitation is for a given fire-affected area. An example of WildfireRain output is presented in Fig. 1.

As can be seen from Figure 1, the comparison of precipitation accumulations to debris flow thresholds is performed within a limited subdomain surrounding the area of interest in proximity to each burn area. This is due to the fact that only neighboring rainfall poses an imminent threat. Therefore, this *buffer* enclosing a burned area should *ideally* extend beyond the burn area's perimeter, in order to observe and assess nearby incoming precipitation from all directions (up to a certain distance). Also, by using these buffers to only evaluate small subdomains of precipitation in close proximity to burned areas, the computational complexity of running WildfireRain remains relatively low (compared to the over 10 million pixels needed to cover the entire US domain, using 1 km grid cells), and would allow the algorithm to operate over multiple burn scars simultaneously. A diagram is presented in Fig. 2, exemplifying how the WildfireRain algorithm relates to burn scars, buffers and rainfall.

These buffers effectively represent the *output space* where WildfireRain can classify precipitation for each corresponding burn scar (each enclosed within a buffer). For a single burned area, or for burn scars that are well distanced from one another, it is trivial to devise subdomains (buffers) that: 1) fully contain a given burned area without interfering with another neighboring one's; and 2) provide enough distance around each burn scar for forecasters to have operationally feasible "lead times". This is the case for the current experimental implementation of WildfireRain, where

Fig. 2 Diagram of how wild fires, burn scars, precipitation and buffers come together and relate to practical implementations of the WildfireRain algorithm. WildfireRain, as a post-wildfire hydrology product, ingests and accumulates real-time radar precipitation which is compared to burn scar-specific thresholds, in order to classify neighboring rainfall according to its potential to trigger a debris flow for a given burn scar

currently a handful of burn scars have been included in a real-time forecast system [23]. This small selection of burn scars are not in close proximity to one another, and each have a sufficiently ample buffer to assess incoming rainfall from all directions. However, taking into account that between 2003 and 2012 alone there were over 1,200 wildfires, it is clear that a large number of these occurred nearby one another. A map showing these burn scars is presented in Fig. 3. Therefore, the need for a solution to optimally assign available buffer space to every burn scar arises. By doing so, we strive to achieve our ultimate operational goal of including all major active burn scars within WildfireRain's outputs.

Close proximity between burn scars, each having unique debris flow thresholds (as explained before), presents complications when attempting to develop a single gridded product (2D *image* file) encompassing all burned areas. As burn scars are close to one another our practical ability to produce buffers around each burn scar, that provide *visibility* for neighboring rainfall in all directions and that don't interfere with other burn scar's buffer, is reduced. In theory, several individual product outputs specific to each burn scar (one per burned area) could be generated, but this would be infeasible in practice due to bandwidth limitations (transferring per-burn scar output files every two minutes) and the limited number of product identifiers that can be assigned within the National Weather Service's (NWS) operational display tool (unique to each operational product). Distributing a single WildfireRain output file (2D *image*) for the whole conterminous United States (CONUS) domain would be ideal for operations. Thus, it is pertinent to develop a singular, gridded product that encompasses all active burn scars (of interest) at once; and in order to do so, the issue of assigning buffers to *nearby* burn scars must be solved.

Fig. 3 A map of the Conterminous United States, showing the locations of 1,270 active burn scars left from wildfires between 2003 and 2012

Let us hypothetically assume that buffers that comply with desired visibility and lead time constraints are arbitrarily assigned to some of the neighboring burn scars. If two burn areas are too close to one another, conflicts arise in how the WildfireRain algorithm should assess pixel classification, if rainfall pixels are present where their respective buffers overlap. Figure 4 exemplifies this type of conflict. An argument could be made for the case of two overlapping subdomains, to classify the conflicting pixels in association with the burn area that has the higher vulnerability (i.e., lower rainfall rate thresholds). However, these rainfall rate thresholds could potentially be drastically different between nearby burned areas, as they could correspond to fires that occurred in different years, burned with very different intensities, and/or are located over vastly different terrain. Furthermore, it is possible to have more than two neighboring burn scars (considering the vast amounts of wildfires that occur every year), which complicates these decisions even further. Figure 5 illustrates how this can be the case.

Another appealing alternative could be resizing neighboring domains such that no conflicting pixels are present over the areas of interest, as shown in Fig. 6. However, it is clear that this could have drastic effects on event forecasting, as the observable distance at which upstream rainfall observations can occur becomes limited. This would entail a reduction in warning assessment times available to forecasters, and could geometrically favor certain directions leaving other cardinalities vulnerable to incoming precipitation. Practically speaking, this is not a completely infeasible alternative, but simply a perceptually undesired one. Nonetheless, for our current approach

Fig. 4 Subset of two burn scars (A and B) that burned between 2003 and 2012; shown with overlapping mock-up buffers, illustrating conflict when the same rainfall has to be evaluated for two distinct sets of thresholds

Fig. 5 Subset of burn areas that burned between 2003 and 2012; shown with overlapping mock-up buffers, illustrating conflict between more than two burn area subdomains

to this problem, we would like to favor all cardinalities equally, to determine an overall feasible observable distance for every burn scar.

Fig. 6 Subset of burn scars that burned between 2003 and 2012; shown with overlapping mock-up buffers, illustrating resized neighboring buffers to accommodate the influence of nearby burn scars

Additional efforts could be made to subset all burn areas into several different sets, which would yield different complementary output files (2D images) corresponding to potentially larger, non-overlapping subdomains for all burn areas (*i.e.* applying the four-color theorem [3]). However, given that WildfireRain is aimed at being used operationally by NWS offices, (as stated before) it would be ideal to present all outputs in a single file (single 2D georeferenced image file) for the CONUS domain. Also, as mentioned earlier, in an operational scenario output files would be generated and distributed to NWS offices every 2 minutes. Therefore, generating more output files would increase the data distribution throughput rates considerably, which is not favorable. WildfireRain is expected to ultimately comply with reasonable and efficient operational forecasting requirements.

Consequently, the primary goal of this study is to explore and propose an efficient initial solution for subsetting the CONUS domain into optimal subdomains around wildfire burn scars, which would enable the WildfireRain algorithm to produce a single output file for all burn scars in the domain. To this end, we address our problem from a mathematical optimization perspective, by defining the *Object Encapsulation Problem* (OEP), in which each object's (burn scars) subdomain (buffer) is maximized, while not interfering with other neighboring objects' buffers. A synthetic scenario for the OEP was proposed, over which we developed a straightforward Linear Programming (LP) model that provides an optimal solution. This LP model is then used to provide optimal solutions over a real-world subset of burn scars across the CONUS. It must be made clear that our current efforts to tackle this problem deal with a somewhat relaxed version of reality: we assume that burn scars are independent, contiguous multi-grid elements over our domain, which do not overlap with one another, and are also not contiguous in any way. This means that every pixel in our domain corresponds to only

one burn scar, and no two burn scars in our domain *touch*. Even though cases that violate these relaxations are feasible in reality: 1) different wildfires throughout the years can have burned the same pixel; and 2) differently catalogued wildfires can burn alongside one another (both in the same or different years), we believe that allowing the aforementioned relaxations are a reasonable compromise which allowed us to begin addressing this problem.

The rest of this article is organized as follows: Section 2 will present a literature review of relevant works which applied a variety of models to address problems pertinent or comparable to our domain subsetting problem; Section 3 introduces the Object Encapsulation Problem, as well as the LP model to solve it optimally for a synthetic scenario; Section 4 demonstrates how the LP model is used to provide optimal solutions over a real-world burn scar data set, leading to conclusions and discussion of future works in Sect. 5.

2 Literature review

The problem of subsetting a subdomain around a given location or set of points, appears to be similar to several canonical problems treated in the realm of optimization. Among the relevant applications we found the topics of *set covering problems* [1], *shape and object packing problems* (both regular and irregular) [2,7,15,19,20,27], *location problems* [8,16,21,28] and *districting problems* [10,17,29] to relate very closely to the general aspects of our specific domain subsetting needs. Even though a wide variety of literature was found within the last 7 years concerning the aforementioned applications, in this section we cover only the most relevant papers that explore diverse methods for solving problems directly comparable or applicable to our domain subsetting problem.

2.1 Shape and object packing problems

The problem of packing shapes in a compact way, or within a pre-defined container is commonly referred to as the shape or object packing problem. Applications for solutions to this problem typically pertain a wide arrange of industries, where optimizing the grouping and packaging of products (similarly-shaped objects) for transport may be needed [27]. Both the shape of the objects and the container can be fixed or variable, as well as regularly or irregularly shaped.

The Circular Open Dimension Problem (CODP), also known as the strip cutting/packing problem, can be approached in different ways, always assuming a constant height for the strip: 1) assuming a fixed length of strip, over which a given number of circles of incrementally variable radii must be placed without intersecting one another (local maximum, strip cutting); 2) assuming a maximum length of strip, over which a given number of circles of incrementally variable radii must be placed as compactly as possible, minimizing the length of strip needed to *pack* them all (local minimum, strip packing). In both cases, the radius for each circle must be maximized, and the width of the strip is held constant. Examples of optimal solutions for both the strip cutting and the circle packing problem are shown in Figs. 7, 8.

Fig. 7 Strip cutting solution corresponding to a local maximum problem, [27]

Fig. 8 Circle packing solution corresponding to a local minimum problem, [27]

The work by Stoyan and Yaskov [27] explored a novel approach towards solving the CODP, by implementing a Jumping Algorithm (JA) methodology. This JA methodology allowed their model to jump from one local extremum to another, and was preceded by a particular way of initializing circles on the strip, using a local maximum solution as a starting point. Additionally, the radii of all circles are modelled as variables simultaneously. By jumping from one local extremum to another once a local minimum (position of circles on the strip) is found for the circle packing arrangement, the radii of all circles is adjusted iteratively, until the area of each circle is maximized with respect to the remaining length of strip. This leads to completely contiguous circles packed into the strictly minimum amount of strip necessary (absolute minimum). This is exemplified in Fig. 9

Similarly, Kampas *et al.* [15] explore solving Object Packing Problems (OPP) by packing ellipses of arbitrary size without overlaps, within a variety of regular optimal polygons. The authors present a methodology based on minimizing the apothem (line segment from the center of a the polygon, to the midpoint of one of its sides) of the polygonal container, solved by using Lagrange Multipliers. Their method encompassed two crucial steps: 1) determining the maximum distance from the center of all polygon faces to each ellipse's boundary, and 2) finding the minimal distance between all pairs of ellipses. For the first part, their work describes explicit analytical formulas, but for the latter, Lagrange multipliers were embedded into the overall optimization strategy (which ultimately prevent the ellipses from overlapping). Even though this

Fig. 9 Circle packing solution corresponding to a local minimum problem, [27]

methodology was designed using regular polygons, the authors state it could be easily extended to irregular convex polygons. In contrast to Stoyan and Yaskov [27], as instead of adjusting the size of the enclosed objects (just their positions), the enclosing domain's (regular polygons) size is adjusted (in two dimensions) optimally to pack objects as compactly as possible.

The domain subsetting problem treated in the present work could be modeled similarly to a CODP or OPP, but with fixed strip length and width corresponding to the whole CONUS domain. Having circular buffers around burn scars would guarantee equal rain observation capabilities in all directions. However, given that the centers for each circle would correspond to centroids for each burn scar, circle locations would have a fixed, immovable origin. This would leave each circle's radius as the sole set of variables needed to be solved for, while guaranteeing circles will not overlap or extend beyond the strip's bounds. It should also be guaranteed that circles extend beyond the perimeter of each burn scar.

2.2 Set covering problems

One could draw similarities between the problem at hand, and applications of the Set Cover Problem. Given the set of pixels on the domain, one could imagine a greedy approach to divide such domain among all burn scars, such that: the maximization of their joined area could provide an optimal solution, while enforcing constraints that meet minimum requirements for buffer extent and distance between burn scars.

Alizadeh and Nishi [1] proposed to solve a hybrid location problem derived from disaster or emergency relief scenarios, which is a meld between the set covering problem (SCP) and the maximal covering location problem (MCLP). In the SCP, a minimum number of facilities is found such that coverage is maximized to meet the demand; alternatively MCLP tries to maximize the covered demands, using a fixed, given set of facilities. Their work developed a novel formulation to solve this hybrid problem, which allowed to model the incremental introduction of facilities as a means to provide full demand coverage across different planning periods. This model reflected the need for providing optimal demand coverage in disaster scenarios, but also a flexible (operational) resource allocation scheme (modularity) for the distinct

facilities set in place, responding to a given emergency. Their work provided a solid conceptualization of their multi-period hybrid model, which was then tested using various numerical scenarios. Results showed that their model was able to provide accurate locations for facilities, as well as providing optimal resource allocation plans for multiple periods. However, the authors note that a more efficient solving method must be developed, before this hybrid model can effectively solve *real-sized* problems.

Set covering problem applications can be conceptually similar to the domain subsetting problem as well. However, these models deal with both the allocation of centers or facilities across the domain and how they associate to the entirety of their surrounding population or resources. In our case, burn scar allocation is not a concern, as these are immovable objects that make up certain portions of the domain. And even though over time these areas may change in shape as they recover (altering their geometry, and therefore their centroids), their initial location is not under our control. Also, for our specific case, not all the domain must necessarily be assigned to a given burn scar, or covered by their buffers.

2.3 Location analysis and districting problems

Location analysis problems primarily revolve around locating facilities to cover customer's demands, however, these can be approached from the inverse perspective: allocating customer to a given set of facilities. This is known as the *districting* problem, which has been applied in various fields such as politics, home health, and school zoning problems [29] [17]. This is relevant for the domain subsetting problem, as one could model the burn scar locations as said facilities, and all neighboring (non-burn scar) pixels could then be treated as customer locations that could be ascribed to a given burn scar.

Tran *et al.* [29] take a metaheuristic approach to solving the districting problem, for the case of patient allocations to nurses that work at a given clinic, which in turn cover a given localized population within the territory covered by the clinic. The territory covered by the clinic was separated into districts, each nurse was responsible for a given district, and each district was assumed to be assigned to only one nurse (as a simplification of the problem). Each district represented a subset of the total list of patients. There was a workload balance to be maintained as new patients were added to the system and assigned to a nurse. Workload was defined in terms of the travel distance for the clinic to a patient's home, as well as the treatment time needed for each patient. It was also desirable that continuity of care was maintained, by keeping the same follow-up nurse of each patient as much as possible, or by avoiding changing a patient's nurse too often. Districts could have different number of nurses assigned to them, and the number of patients visited by every nurse could vary depending on the district.

Being in essence an assignment problem, this districting problem was formalized by using a mathematical model, and solved using both a Tabu search algorithm and a Simulated Annealing algorithm. To model this problem, Tran *et al.* implemented a knapsack problem to select neighboring districts, and using the *k*-means algorithm to reduce the traveling times within a district. Neighboring districts were defined by

Fig. 10 Example Voronoi diagram, corresponding to the assignment of 19 patients and 3 nurses, [29]

using Voronoi diagrams (See Fig. 10). Given that the list of patients was updated on a monthly basis, model assignment states were modelled iteratively, where at every step patients who did not require further follow-up care were removed from the patient list, and new patients in need of care were added.

Tran *et al.* contrasted results for their models by solving for synthetic and real data sets. Their results show that for small differences in workload ranges, the Tabu search-based algorithm was able to maintain workload equilibrium among nurses better than the Simulated Annealing approach. This held true for both synthetic and real data. Tabu search-based solutions also allowed for better quality assignment of nurses, leading to smaller workloads and better preserving continuity of care.

Mallozzi and Puerto [21] propose a dual problem solution to the location analysis problem (districting). Their work aimed to design the optimal partition of regions surrounding a distinct set of facilities, while complying with specific criteria (particularly distance). For their study, distance was assumed to be a proxy for costs to be minimized, resulting in the minimization of the average distance covered by customers to their assigned facilities. To address this problem, the authors proposed to find the best approximation, by discrete measures, of a density function on a compact subset of $\mathbb{R}^n$, by making use of the p-Wasserstein distance. Also, Voronoi diagrams were used to visualize results, as they exhibit closeness properties with respect to their centers, and these behave differently depending of which distance measure is used (See Fig. 11). To estimate the cost (in distance) between each customer location and a given facility, transport maps were established (based on Borel sets and functions); solutions that minimize the total cost were referred to as optimal transport maps.

The Voronoi diagrams in Fig. 11 show the impact of each distance metric had on the properties of the final solution. The use of squared Euclidean measurements relates to scenarios where a standard distribution of errors is assumed, while the use of polyhedral or block norms (ℓ_1 , ..., ℓ_∞) often relate to model realistic scenarios more accurately [21]. In addition to subsetting domains according to given center points, Mallozzi and Puerto [21] considered the case where facilities in the domain were represented by

Fig. 11 Optimal domain partitions using four different distance metrics for three points distributed in space, [21]

circles. This allowed to model scenarios where the area and shape of a given center weights the influence it has on its surrounding domain. In order to achieve this, the distance metrics were modified to express the distance from any given point to a circle's edge. Even though different results were obtained by implementing circular facilities, the authors note that results remain rather similar (shapewise) to the ones obtained with punctual facilities.

In Kong *et al.* [17], the authors focus on districting problem applications that require three basic criteria to be satisfied: balance, compactness and contiguity (*i.e.* political districting, or sales territory design). Balance refers to the desire of delineating districts that are very similar one to another (in terms of area, population, workers, sales potential, *etc.*); compactness refers to the geometry of a district, where a perfectly round district would be the ideal; and a district is contiguous if travel is possible between all areal units enclosed within it, without having to leave the district. Even though solving these types of problems are very computationally expensive, the authors remark on how these can be generalized to the "graph partitioning problem", where a graph is split into a given number of independently connected clusters.

Kong *et al.* propose a center-based (district centers are known) mixed-integer linear programming (MILP) model to solve the districting problem, optimizing weighted

Fig. 12 Example of a p -center problem solution: a set of demand points and service points. Notices how the demand points have been placed and assigned a radius which result in the coverage of all demand points. All radii are smaller or equal to a pre-established maximum distance [16]

objectives of district balance and compactness, while ensuring contiguous districts. Their approach was a product of combining the network flow model (which ensured district contiguity) and the general location-allocation model (which served as a mechanism to promote compactness). In order to calculate the district centers, a multi-start weighted k -medoids (most centrally located point) algorithm was used on three different scenarios corresponding to 297, 324 and 1297 areal units to be grouped into 10 districts. Their center-based approach was shown to outperform local-search and metaheuristic-based approaches. Final results showed that their approach was able to solve all model instances optimally or near-optimally.

Another family of problems related to location analysis is the p -center problem (PCP). It comes in several variations, but the general idea is to minimize the maximum distance between a demand point and a closest point belonging to a set of p supply points [4].

Suzuki and Drezner [28] approached a PCP to locate p facilities within a given area to provide service to demand points within it. Their objective was to minimize the maximal distance for all demand points, and thus to cover every point of a given area (a square) by p circles with the smallest possible radius. They proposed several heuristic procedures and upper bounds on the optimal solution, and presented several computational results. Among the heuristics explored by these authors, the Voronoi heuristic was explored and proposed to approximate solutions to the PCP, paired with a Non-Linear Programming (NLP) algorithm to *finalize* an optimal solution. The work by Suzuki and Drezner expanded previous work, which explored PCP applications on graphs, trees and discrete Euclidean planes, onto continuously distributed demands over an area.

Kaveh and Nasr [16] approached the solution to both the conditional PCP (where q facilities are already in place in the search space) and the unconditional PCP using a modified harmony search algorithm, a metaheuristic approach. Their results for the modified harmony metaheuristic are compared to the classic (unmodified) harmony search approach, as well as other metaheuristics like Tabu search and variable neighborhood search. The authors indicate that their approach is applicable to both discrete and continuous search spaces. However, only discrete problems were solved

and benchmarked. Lastly, a real world application is presented, where their modified harmony location model is applied to locate bicycle stations in a major city in Iran: the first time as an unconditional PCP (no existing stations in the first year), and anticipating an increment of the available stations for next year, implying a conditional PCP to be solved in the near future. Their study concluded that their modified version of the harmony search algorithm is faster and more efficient than the other benchmarked methods.

Du *et al.* [8] propose a two-stage robust model to reliably locate facilities to provide a service when existing facilities are disrupted (i.e. by natural disasters). Their approach is to design reliable networks "proactively" during a planning phase (first step), and once a disruption occurs the affected demand can be reallocated to another viable facility (second step). Their proposed solution is based on the PCP, as an attempt to optimize the worst case performance of the service network, and taking into account the reliability for every demand node (client). These authors presented three distinct solution methods for their model, which were then compared and analyzed in terms of performance. Results yielded favorable outcomes for two out of the three methods, one being favored over the other, depending on the size of the scenario to be solved.

The same assessment concerning set covering problem applications is true for location problem applications. One of the closest matches to the problem at hand is the districting problem, as in these scenarios, it is feasible to deal with *a priori* district centers dictated by external factors (geography, population density, *etc.*). Nonetheless, special consideration is given to aspects that are not an issue in our case, such as district contiguity and balance. For the domain subsetting problem, we assume contiguous districts (buffers, subdomains enclosing individual burn scars), and as burn scars vary widely in characteristics (area for example), balance is not of interest as districts should be sized independently for each case. District compactness should be guaranteed by definition in our problem, as every burn scar is defined as a continuous, compact object, that must be ideally encapsulated within a circular buffer. Also, several variations of the PCP resemble the problem the present article hopes to address, and the *vertex-restricted* PCP [18] seems to be the one that resembles it the most. In this vertex-restricted format the possible locations for supply nodes correspond to the already existing vertices (demand nodes) on the network (opposed to the *absolute* PCP, where supply nodes can be placed anywhere on the network, including the edges). Perhaps our problem of subsetting the CONUS domain could be modeled as a vertex-restricted PCP where p equals the total number of vertices in the network. The use of Voronoi diagrams/heuristics could provide a suitable approximation towards improving solutions to the burn scar domain subsetting problem, if a feasible way to constrain these subdomains can be found, so that not all pixels in the CONUS domain are assigned to a given burn scar's buffer. If all pixels in the domain were to be processed by the WildfireRain algorithm, its computational costs would be maximized hindering its performance dramatically.

To our knowledge, and after reviewing relevant works applicable to the domain subsetting problem presented in this research, no work to date in the literature addresses our particular problem. Therefore, the present article likely is the first research to address a domain subsetting problem for wildfire burn scars and post-wildfire hydrology applications.

3 The object encapsulation problem (OEP)

The main goal of this study is to explore the subsetting of the CONUS domain into optimal subdomains (buffers) around wildfire burn scars, in order to allow for unified, domain-wide WildfireRain product outputs. A simplified version of this problem is defined as a special case of the Object Encapsulation Problem, where burn scars are represented by objects in an evenly-gridded domain (all grid cells are of equal size), a minimum distance between burn scars is pre-defined, and the buffers to be constructed around them (encapsulating them) are to be optimally-sized circles. Because burn scars are practically immovable objects in nature, the core of the problem relies on maximizing the reach of every buffer, while guaranteeing that the buffers are disjoint. In the next subsection we provide a formal definition of the generalized OEP, along with a mathematical formulation applicable to the simplified version aforementioned, followed by an illustrative example built on a reduced synthetic data set.

3.1 OEP definition

Let there be a grid of n columns and m rows (which forms a collection of $m \times n$ cells) that contains o disjoint objects (where an object is defined as a contiguous collection of cells) with associated weights w_k , $\forall k \in \{1, 2, \dots, o\}$. The Object Encapsulation Problem (OEP) is defined as the problem that seeks to find the set of areas that encapsulate all o objects, so that the weighted sum of a given function of the encapsulations (such as their area or their perimeter) is maximized.

3.1.1 Problem formulation for circular encapsulations with fixed centers

Let us assume the special case where (i) there are at least two objects to be encapsulated, (ii) all encapsulations are circular, and (iii) the centers of all the encapsulations are fixed, and (iv) we are interested maximizing the weighted sum of the perimeters of the encapsulations. To solve the OEP for this case, we propose the following LP model:

Sets:

$O = \{1, 2, 3, \dots, o\}$: set of objects in the domain

Parameters

$d_{kk'}$: distance between the centers of encapsulation $k \in B$ and encapsulation

$k' \in O : k \neq k'$. ($d_{kk'} \in \mathbb{R}_{>0}$)

l_k : minimum radius needed to encapsulate object $k \in O$. ($l_k \in \mathbb{R}_{>0}$)

w_k : weight assigned to object $k \in O$. ($w_k \in \mathbb{R}_{>0}$)

Decision Variables

r_k : encapsulation radius for object $k \in O$

Problem Formulation

$$\text{Maximize} \quad \sum_{k \in O} w_k (2\pi r_k) \quad (1a)$$

subject to

$$r_k \geq l_k, \quad \forall k \in B \quad (1b)$$

$$r_k \leq \frac{w_k}{w_k + w_{\arg \min_{k' \in O: k \neq k'} (d_{kk'})}} [\min_{k' \in O: k \neq k'} (d_{kk'})], \quad \forall k \in Q \quad (1c)$$

$$r_k \in \mathbb{R}_{\geq 0}, \quad \forall k \in O. \quad (1d)$$

Constraints 1b dictate that radii must be greater than or equal to the minimum needed distance for each object. This guarantees that the final encapsulations contain the entirety of all the objects. Constraints 1c enforce that encapsulation radii are less than or equal to the maximum distance assignable between each object and its closest neighbor, scaled by its weight, to guarantee that the encapsulations are always disjoint. Note that, depending on the application, one may want to add a tighter upper bound to the radii, so constraints 1c would have to be updated accordingly. For example, in the case study associated with burn scars, it was important that the radii was never larger than half the distance from its closest neighbor, so the right hand side of constraints 1c would have to be modified to $w_k [\min_{k' \in O: k \neq k'} (d_{kk'})] / (2 \sum_{k \in O} w_k)$. Finally, constraints 1d denote the nature of the decision variables. Note that enforcing integrality for radii r_k could be desirable, particularly where objects are defined in a discretized domain (such as a grid).

Note that if, for a given scenario, there is an object $k \in O$ such that $l_k > \frac{w_k [\min_{k' \in O: k \neq k'} (d_{kk'})]}{w_k + w_{\arg \min_{k' \in O: k \neq k'} (d_{kk'})}}$, under the assumptions made there is no set of disjoint circular encapsulations that would solve the OEP, and therefore the proposed LP model would be infeasible. However, as long as the objects are disjoint, there always exists a feasible set of disjoint encapsulations that cover all objects (which, in the worst case scenario, would correspond to encapsulations that are equivalent to the objects they contain). Thus, the observed infeasibilities would be caused by diverse special cases regarding the shape and proximity of the objects, and the assumptions made associated with having fixed centers and circular encapsulations. In the following subsection we present a list of such special cases, along with steps that can be taken to address them.

3.2 Special cases

There may be multiple circumstances where the assumptions made to simplify the present problem may cause infeasibilities in the proposed LP model. Generally, these cases are considered uncommon: from our own experience working with data sets in the order of hundreds of burn scars over the CONUS, only a handful conflicting cases have been generally found. These were identified by attempting to solve the LP model, and subsequently addressed manually. Therefore, we expect that for most use cases and scenarios, our LP formulation should be able to provide adequate solutions. Nonetheless, in this section we aim to address all possible cases that prove infeasible with our LP formulation, and provide the reader with sufficient solutions to deal with them in case they occur.

One of this instances is where there exists a set of disjoint circular encapsulations for all objects, but the predetermined centers and l_k do not necessarily yield the min-

Fig. 13 Case where the minimum bounding box method using the burn scar's centroid for finding the minimum distance l_k , does not necessarily yield the minimum bounding circle

imum encapsulating circle, causing infeasibility. In some cases, the infeasibility may be caused by using encapsulations centers and l_k that do not result in a small enough encapsulation, even though it was possible to further reduce the circular encapsulations. This may happen if the values of l_k were determined using an approximation method, such as the minimum bounding box method.

In this instance, one should select the objects that cause infeasibility and test for the following: (i) If the bounding boxes are disjoint for these objects, the associated encapsulation circles should be updated to circumscribe the bounding boxes (i.e., make l_k equal to half the diagonal of the associated bounding box, and the center of the encapsulations should be equal to the center of the bounding box). Note that the circular encapsulation may be reduced even further than this (Fig. 13). (ii) If the bounding boxes are superposed, but there exist radii l_k (without modifying the fixed centers) that would lead to disjoint circular encapsulations. In this case, one would have to simply reduce the associated encapsulation radii until no overlap occurs (Fig. 14). (iii) If the bounding boxes are superposed, but there exist radii l_k that would lead to disjoint circular encapsulations only if modifying the fixed centers. In this case, one should select the objects that cause infeasibility and find new centers and encapsulation radii so that no overlap occurs (Fig. 15).

Another instance that may cause infeasibility is where there does not exist a set of disjoint circular encapsulations for all objects. In this instance, it would be necessary to introduce divisory lines separating the objects that cause the infeasibility.

In some particular cases it may be possible to simply add straight divisory lines to modify and fix overlapping circular encapsulations. For example, if the objects are contiguous only with vertically aligned cells, one would introduce a vertical divisory line (Fig. 16); if the objects are contiguous only with horizontally aligned cells, one would introduce a horizontal divisory line (Fig. 17); finally, if the objects are in contact over diagonally adjacent cells, the separating line between both buffers can be represented in infinite ways, but its overall orientation can be generalized by making it orthogonal to the line that passes through the centroids of the two associated objects

Fig. 14 Case where two conflicting minimum distances l_{kA} and l_{kB} can be solved by shrinking l_k as much as needed, as not to solve the conflict, but not below each burn scar's minimum bounding circle

Fig. 15 Case where using the minimum bounding circles for both burn scars still presents a conflict with their respective minimum distances. Here, the centers for each circle can be adjusted in order to eliminate the conflict

(Figure 17). In other cases where a straight divisory line can be introduced, in general there would be infinity ways of introducing divisory lines, that may go in hand with simultaneous changes in the radii and centers of the encapsulations made. For example, as illustrated in Figure 19, one option to regain feasibility would be to simply add a divisory line for the two objects in the upper region, while a different option would be to instead modify the centers and radii of the minimum encapsulations for the two upper objects so that they do not overlap, which would require adding a divisory line with the third object in the lower left corner.

In the more challenging cases, the encapsulations may not be fixed by adding straight divisory lines, so it would be necessary to add curve or segmented divisory lines, which would work for non-contiguous and contiguous objects, as shown in Figs. 20 , 21, respectively.

Fig. 16 Case where two burn scars are contiguous over a portion of vertically aligned pixels

Fig. 17 Case where two burn scars are contiguous over a portion of horizontally aligned pixels

Fig. 18 Case where two burn scars are contiguous over diagonally adjacent pixels

Fig. 19 Cases where proximity between burn scars require adding divisory lines, with the possible modification of the radii and centers of the minimum accepted encapsulations. In the first case, by increasing the l_{kA} , we can avoid the conflict with l_{kB} , but if l_{kA} were to be large enough, it could conflict now with other neighboring burn scars (l_{kC})

Fig. 20 Case where neighboring burn scars are not contiguous, and are not separable by a straight line

It is to be noted that every time a divisory line is added between to objects, one would have to modify the LP accordingly, so that it can be used to obtain quasi-optimal encapsulations. In these cases, constraints 1c should be modified so that the distance between the associated objects is not considered, i.e., $d_{kk'}$ should be defined as an arbitrarily large number if there is a divisory line between objects k and k' .

Having described the aforementioned special cases, it can be seen that it would not be possible to construct a general optimization model for all possible OEP cases without having to use non-circular shapes, foregoing linearity, and further increasing the complexity of the approach. Figures 13-20 illustrate these cases in the context of wild fires, where the burn scars are the objects and the encapsulations correspond to the buffers.

Fig. 21 Case where neighboring burn scars are contiguous over a gridded domain, and are not separable by a straight line

3.3 Illustrative example

Following the last subsection's OEP and LP model definitions, the following synthetic simplified scenario is proposed as a means to explore and exemplify how the formulated model works. A sparse 25×25 binary matrix with 10 1-values is generated, which represented each burn scar (object). At least 2 grid cells separate each object from the rest. This scenario can be seen represented graphically in Fig. 22.a. Note how each burn scar has a value of $l_k = \frac{s}{2} = 1$ due to the $s = 2$ grid cell separation that was assumed. Also note that all $w_k = 1, \forall k \in B$ as all burn scars are weighted equally. The necessary distance matrix was calculated (Table 1) for all objects, and all necessary sets and parameters were coded up using Python and Gurobi [13]. Optimal solution for the scenario were obtained within few iterations. Three types of solutions were calculated over the data set for the following cases: an LP model with continuous radii values and equal weights; an Integer Linear Programming (ILP) model where integer radii values were enforced with equal weights; and an LP model with continuous radii values, and unequal weights for six of the objects.

The results of solving the object encapsulation problem for 10 randomly positioned burn scar centers, represented by single cell objects, using the LP model with continuous radii values are presented in Table 2 and in Fig. 22.b. As can be seen, the optimal solution provides maximal radii for every burn scar object, which do not interfere with other neighboring buffers, and at most only *touch* a neighboring buffer.

For the case of the ILP implementation, results are presented in Table 3 and in Fig. 23.a. As can be seen, the optimal solutions provide maximal radii for every burn scar object, which do not interfere with other neighboring buffers, and also guarantee that a discernible amount of space is left between nearby buffers, due to the integer nature of the solution.

Lastly for the LP implementation with weights $w_k \neq 1$ for objects $k = \{0, 3, 6, 7, 8, 9\}$ of $\{1.4, 0.6, 0.7, 1.05, 1.2, 1.05\}$ respectively, the resulting radii can be seen in Fig. 23.b. Here it can be seen how the radii for objects 3 and 6 were reduced

Table 1 Distance matrix $d_{kk'}$ between all burn scars calculated for the test case of the object encapsulation problem

$d_{kk'}$	0	1	2	3	4	5	6	7	8	9
0	-	11.05	20.02	6.71	15.81	13.60	23.60	23.26	30.41	22.02
1	11.05	-	9.00	9.43	8.25	13.89	15.26	20.02	22.83	24.19
2	20.02	9.00	-	17.72	10.63	20.00	13.04	22.36	20.10	29.70
3	6.71	9.43	17.72	-	10.44	7.07	17.89	16.55	24.21	16.49
4	15.81	8.25	10.63	10.44	-	9.85	7.81	12.37	15.00	19.10
5	13.60	13.89	20.00	7.07	9.85	-	15.03	10.00	19.70	10.30
6	23.60	15.26	13.04	17.89	7.81	15.03	-	11.40	7.62	21.54
7	23.26	20.02	22.36	16.55	12.37	10.00	11.40	-	12.00	11.05
8	30.41	22.83	20.10	24.21	15.00	19.70	7.62	12.00	-	23.02
9	22.02	24.19	29.70	16.49	19.10	10.30	21.54	11.05	23.02	-

Table 2 Optimal LP results for all burn scars calculated for the test case of the object encapsulation problem, for $r_k \in \mathbb{R}$

r_k	0	1	2	3	4	5	6	7	8	9
0	-	-	-	3.35	-	-	-	-	-	-
1	-	-	-	-	4.12	-	-	-	-	-
2	-	4.50	-	-	-	-	-	-	-	-
3	3.35	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	3.90	-	-	-
5	-	-	-	3.53	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	3.80	-
7	-	-	-	-	-	5.00	-	-	-	-
8	-	-	-	-	-	-	3.80	-	-	-
9	-	-	-	-	-	5.15	-	-	-	-

Table 3 Optimal ILP results for all burn scars calculated for the test case of the object encapsulation problem, for $r_k \in \mathbb{N}$

r_k	0	1	2	3	4	5	6	7	8	9
0	-	-	-	3.0	-	-	-	-	-	-
1	-	-	-	-	4.0	-	-	-	-	-
2	-	4.0	-	-	-	-	-	-	-	-
3	3.0	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	3.0	-	-	-
5	-	-	-	3.0	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	3.0	-
7	-	-	-	-	-	5.0	-	-	-	-
8	-	-	-	-	-	-	3.0	-	-	-
9	-	-	-	-	-	5.0	-	-	-	-

Fig. 22 Graphical representation of the binary matrix, which shows the 10 randomly generated objects over the gridded domain, and the optimal solutions found resulting from the LP model. In both subfigures, the black squares represent cells with a value of 1, which correspond to hypothetical burn scar centers; a 1 grid cell minimum radius is also shown for each object. For subfigure b) the continuous optimal radii ($r_k \in \mathbb{R}$) found for each burn scar are shown as well, according to the locations of the other objects

considerably as their weights were less than 1. Objects 7 and 9 present a very slight increase in radius due to a $w_k = 1.05$, and the increase in radius for objects 0 and 8 is rather noticeable due to the magnitude of their weights.

Having defined the object encapsulation problem and proposed an LP model to address it (using both integer and continuous radii), it was successfully tested on a synthetic scenario. As a means for validating this method, Section 4 addresses the application of the developed LP model over a real-world data set, for a selection of wildfire burn scars over the US.

4 Case Study: 2012 fires which burned over 30% of basin areas

As a result of successfully testing the proposed solution to the object encapsulation problem using a synthetic dataset, a real-world validation case study was sought after. 2012 is on record as one of the top five years with the largest wildfire acreage burned since 1960 [14], making it a suitable candidate for this study. From the over 80,000 wildfires that occurred that year [14], only a few particular burn scars were selected under very specific considerations: burn scars left after fires that burned 30% or more of a given basin, and that burned only once throughout 2012. This vastly reduced the dataset of available candidate burn scars, but also guaranteed that the remaining ones represented the most at-risk areas that were affected by any single wildfire event. Figure 24 shows their location on a map subset of the CONUS.

Optimal ILP solution for single-cell object encapsulation
with $n(B) = 10; l_k = 1, w_k = 1, \forall k \in B$

(a) Optimal ILP solution radii ($r_k \in \mathbb{N}$)
for the synthetic scenario

Optimal LP solution for single-cell object encapsulation
with $n(B) = 10; l_k = 1, \forall k \in B; w_k = 1, \forall k = \{1, 2, 4, 5\}$
 $w_k \neq 1, \forall k = \{0, 3, 6, 7, 8, 9\}$

(b) Optimal solution radii ($r_k \in \mathbb{R}$)
for the synthetic scenario, with
different weights for each burn scar

Fig. 23 Additional results from running the optimization model for ILP and LP using different parametrizations. In both subfigures, the black squares represent cells with a value of 1, which correspond to hypothetical burn scar centers; a 1 grid cell minimum radius is also shown for each object. For subfigure a) the optimal integer radii ($r_k \in \mathbb{N}$) found for each burn scar are shown, according to the locations and equal weights of the other objects. For subfigure b) optimal continuous radii ($r_k \in \mathbb{R}$) found for each burn scar are shown as, according to unequal weights assigned to given objects. It must be mentioned that the weighting scheme implemented to obtain these results are such that the sum of all weights assigned to all objects were equal to the cardinality of B ($\sum_{k \in B} w_k = n(B)$)

Fig. 24 Burn scar locations for the 2012 wildfires which burned only once, and impacted over 30% of their respective basin(s)

Fig. 25 Graphical representation of a burn scar on the gridded domain and its associated centroid

The starting point for the preparation of the data set was a raster image file of these burn scars over the whole CONUS domain (represented by burned and unburned pixels). The raster image corresponded to the operational domain resolution of 1-km grid cells over the CONUS, which amounts to 7000x3500 pixels. This data was then analyzed using scikit-image, an image processing library for Python [30]. Through this analysis, the contiguous burned pixels were first classified as definite *objects* (B), which rendered a count of 24 distinct burned areas across the gridded domain. Then, centroids (single-cell object representation) were calculated for each burn scar object, as well as the rectangular bounding box needed to contain each and every one of them. By calculating the distances between each corner of the bounding box and each centroid, and keeping the maximum value, the radius of a minimum bounding circle (l_k) was defined for every burn scar object. Even though the apothem was used in a similar fashion by Kampas *et al.* [15], due to the irregular nature of these burn scar objects, the maximum distance to the farthest vertex was chosen (radius).

Note that using the *minimum bounding circle* (MBC) method to calculate l_k could potentially yield smaller minimum radii values, than when using the *minimum bounding box* (MBB) method (see Figure 13). However, in cases such as the one presented in Fig. 15, using a MBC to define l_k would still require displacing the circle's center away from the burn scar's centroid to address infeasibilities. For our problem abstraction we allowed the circle's centers to remain consistent with the single-pixel representation of our burn scars (centroid-based), as this simplified our modeling approach. And even though the MBB method is not necessarily the only method that can achieve this consistency (as it is also achievable using MBC), making use of it helped us reduce computational complexity. This is due to the fact that calculating the bounding box is a trivial task, compared to determining the maximum distance from the centroid to any and every pixel in the burn scar's edge.

Having constructed a single-cell representation for every burn scar, as well as defining the minimum radius needed to guarantee coverage of all cells that make up every object, the distance matrix between all burn scars in the domain was calculated ($d_{kk'}$).

Table 4 Optimal burn scar radii and parameters

Burn Scar	x	y	l_k	r_k	w_k
1	2154.37	664.87	3.37	142.0	1
2	949.51	769.70	15.41	77.0	1
3	843.78	883.98	8.78	70.0	1
4	2276.80	921.28	5.85	142.0	1
5	1378.69	942.15	7.94	26.0	1
6	1350.92	987.84	3.77	26.0	1
7	828.10	1023.83	7.58	70.0	1
8	1495.48	1056.36	16.59	37.0	1
9	1457.37	1122.08	20.95	37.0	1
10	1948.37	1238.86	6.43	155.0	1
11	1583.96	1270.59	16.07	97.0	1
12	2464.09	1271.26	11.60	167.0	1
13	1178.10	1295.11	24.68	164.0	1
15	794.19	1396.23	11.73	77.0	1
16	688.0	.92	6.46	39.0	1
17	1843.61	1531.38	7.09	24.0	1
18	1889.14	1547.64	4.40	24.0	1
19	1776.39	1562.70	6.89	37.0	1
20	738.33	1569.07	8.20	39.0	1
21	2505.48	1604.13	8.76	167.0	1
22	3358.17	1898.32	11.06	451.0	1
23	1851.18	2102.50	8.07	149.0	1
24	2425.26	2157.78	5.23	142.0	1
25	2140.75	2175.62	17.04	142.0	1

Just like in Sect. 3, equal weights were defined for all objects. With all the parameters and sets defined to represent the real-world data, the Linear Programming model implementation was used. The resulting optimal radii are shown in Table 4, alongside each burn scar's parameters and label. A visualization of these results are shown in Fig. 26.

As can be seen from the above representations, the output of the LP model resembles the expected behavior observed on the simplified example scenario built on synthetic data. As can be seen, even though burn scar number 22 could have a radius much larger than it was assigned, because of constraint 1c in the model's formulation (Section 3), the maximum distance assignable to every object's buffer can not exceed half of the distance to its closest neighbor. Even though this may be a limiting factor, this can be circumvented by modifying the weights assigned to each burn scar, as these will scale all optimal resulting radii proportionally. This constraint was designed with this maximum cutoff as a means to guarantee an object's maximum buffer expansion without hindering any neighboring object buffer's coverage. Also, as mentioned before, by not

Optimal LP solution for single-cell burn scar encapsulation with $n(B) = 24; l_k = r_k^{\min}, w_k = 1, \forall k \in B$

Fig. 26 Graphical representation of the optimal solution for the validation case of the object encapsulation problem. The shaded pixels represent burned areas; the red dots represent the burn scar centers, and the minimum radius is also shown for each object alongside the optimal radius found for each burn scar

covering all the domain grid cells, the computational complexity of the WildfireRain product is reduced.

Concerning the real-world dataset, the need to produce buffers with a discernible distance between them has not been identified. Therefore making use of the LP model over this dataset using integer radii was not considered necessary. Additionally, no weighting scheme was tested over the real-world dataset as an operationally-sound weighting scheme should be defined for assessing this aspect. Even though each burn scar's area could be used to propose a weighting scheme, this factor is merely one of many that should be considered for this task; slope, burn severity, recovery time and proximity to populated areas should also play a relevant role in characterizing the weighting scheme as well. Nonetheless, as seen in the results presented in Sect. 3, our model has been proven capable of handling the aforementioned tasks if required.

5 Conclusions and future work

In the present work, we proposed and defined the Object Encapsulation Problem (OEP), and developed a Linear Programming (LP) model to solve it. This model was successfully applied as a tool to solve the Object Encapsulation Problem: initially for a simplified example scenario of the domain subsetting problem relating to burn scars across the CONUS domain, and later for a real-world dataset of burned areas that occurred in 2012. This paper shows an efficient and highly scalable method to create optimal separate subdomains for each burn scar (effectively encapsulating them), while maximizing their buffer's reach and guaranteeing non-interference with neigh-

boring burn scars. The capabilities developed through the present work are likely the first of its kind, which address a specific domain subsetting problem aimed at enabling more efficient computational post-wildfire hydrology applications. Even though several uncommon special cases that prove infeasible have been identified, and plausible strategies to address them have been outlined, the present work proposes our LP formulation as a simplified yet capable alternative to provide optimal solutions for the OEP.

Future work could explore incorporating ways of handling special cases where different burn scars are placed very closely to one another, and the notion of a minimum radius for each burn scar would result in infeasible problem spaces for the current model. A viable approach to solving the special case portrayed in Figure 15, where the centers of minimum distance circles would have to be displaced, could include the use of a bi-level optimization model. This way the optimal placement of centers for circular encapsulation within each burn scar could be nested within the larger problem of encapsulating all objects in the scenario. This would enable to first find the optimal centers for each burn scar (not necessarily equal to the centroid), and then proceed with finding the optimal buffers around them. However, as shown in Subsect. 3.2, shifting the centers would not necessarily guarantee feasibility (see Figs. 20, 21), and foregoing linearity, as well as increasing complexity may be required to properly address these cases.

Taking inspiration from Voronoi diagrams, the current model could also be expanded to enable encapsulations that hybridize circular and polyhedral shapes, without having to cover all possible pixels in the domain. Similarly, incorporating or modifying models to solve the *vertex-restricted* p-center problem could also provide a feasible avenue towards improving modeling efforts, and removing some of the relaxations and assumptions made in this present work about both burn scars and the Object Encapsulation Problem. Also, comprehensive, operationally-sound methods for weighting burn scars according to their morphological and physical properties, as well as their burn recovery times should be sought after. Such improvements could lead to multi-year weighted burn scar schemes, which could provide the WildfireRain algorithm a means of optimally assessing incoming precipitation over all active burn scars in the CONUS domain.

Acknowledgements The research reported herein was supported and funded by: 1) the Center for Risk Based Community Resilience Planning, funded by the National Institute of Standards and Technology (NIST) under Cooperative Agreements No. 70NANB15H044 and 70NANB20H008; 2) the NOAA/Office of Oceanic and Atmospheric Research (OAR) under NOAA-University of Oklahoma Cooperative Agreement No. NA11OAR4320072, U.S. Department of Commerce, National Oceanic and Atmospheric Administration; 3) the NOAA/OAR - Joint Technology Transfer Initiative (JTTI) program under Federal Award No. NA20OAR4590354, U.S. Department of Commerce, National Oceanic and Atmospheric Administration. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NIST, NOAA, CIMMS, The University of Oklahoma, or the U.S. Department of Commerce.

Data Availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Alizadeh, R., Nishi, T.: Hybrid Covering Location Problem: Set Covering and Modular Maximal Covering Location Problem. In: 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp. 865–869. IEEE, Macao, Macao (2019). <https://doi.org/10.1109/IEEM44572.2019.8978601>
2. Andrade, R., Birgin, E.G.: Symmetry-breaking constraints for packing identical rectangles within polyhedra. *Optimization Lett.* **7**(2), 375–405 (2013). <https://doi.org/10.1007/s11590-011-0425-9>
3. Appel, K., Haken, W.: Every planar map is four colorable Part I: discharging. *Illinois J. Math.* **21**(3), 429–490 (1977). <https://doi.org/10.1215/ijm/1256049011>
4. Calik, H., Labbé, M., Yaman, H.: p-Center Problems. In: G. Laporte, S. Nickel, F. Saldanha da Gama (eds.) *Location Science*, pp. 79–92. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-13111-5_4
5. Cannon, S.H., Kirkham, R.M., Parise, M.: Wildfire-related debris-flow initiation processes, storm king mountain, colorado. *Geomorphology* **39**(3), 171–188 (2001). [https://doi.org/10.1016/S0169-555X\(00\)00108-2](https://doi.org/10.1016/S0169-555X(00)00108-2)
6. Cerdà, A.: Changes in overland flow and infiltration after a rangeland fire in a Mediterranean scrubland. *Hydrological Processes* **12**(7), 1031–1042 (1998). [https://doi.org/10.1002/\(SICI\)1099-1085\(19980615\)12:7%3c1031::AID-HYP636%3e3.0.CO;2-V](https://doi.org/10.1002/(SICI)1099-1085(19980615)12:7%3c1031::AID-HYP636%3e3.0.CO;2-V)
7. Chu, Y., Lin, H.: Repack: Dense object packing using deep cnn with reinforcement learning. In: 2019 International Automatic Control Conference (CAC), pp. 1–5 (2019)
8. Du, B., Zhou, H., Leus, R.: A two-stage robust model for a reliable p-center facility location problem. *Applied Mathematical Modelling* **77**, 99–114 (2020) <https://doi.org/10.1016/j.apm.2019.07.025>. <http://www.sciencedirect.com/science/article/pii/S0307904X19304263>
9. Ebel, B.A., Moody, J.A.: Rethinking infiltration in wildfire-affected soils. *Hydrol. Process.* **27**(10), 1510–1514 (2013). <https://doi.org/10.1002/hyp.9696>
10. Fragoso, R., Rego, C., Bushenkov, V.: Clustering of Territorial Areas: A Multi-Criteria Districting Problem. *J. Quantitative Econom.* **14**(2), 179–198 (2016). <https://doi.org/10.1007/s40953-016-0030-y>
11. Gourley, J.J., Flamig, Z.L., Vergara, H., Kirstetter, P.E., Clark, R.A., Argyle, E., Arthur, A., Martinaitis, S., Terti, G., Erlingis, J.M., Hong, Y., Howard, K.W.: The FLASH project: Improving the tools for flash flood monitoring and prediction across the united states. *Bull. Am. Meteorol. Soc.* **98**(2), 361–372 (2017). <https://doi.org/10.1175/bams-d-15-00247.1>
12. Gourley, J.J., Vergara, H., Arthur, A., Clark, R.A., III, Staley, D., Fulton, J., Hempel, L., Goodrich, D.C., Rowden, K., Robichaud, P.R.: Predicting the Floods that Follow the Flames. *Bull. Am. Meteorol. Soc.* **101**(7), E1101–E1106 (2020). <https://doi.org/10.1175/BAMS-D-20-0040.1>
13. Gurobi Optimization, L.: Gurobi optimizer reference manual (2020). <https://www.gurobi.com/documentation/9.1/refman/index.html>
14. Hoover, K., Hanson, L.A.: Wildfire statistics, august 20, 2020 - IF10244. IN FOCUS (2018)
15. Kampas, F.J., Castillo, I., Pintér, J.D.: Optimized ellipse packings in regular polygons. *Optimiz. Lett.* **13**(7), 1583–1613 (2019). <https://doi.org/10.1007/s11590-019-01423-y>

16. Kaveh, A., Nasr, H.: Solving the conditional and unconditional p-center problem with modified harmony search: A real case study. *Scientia Iranica* **18**(4), 867–877 (2011) <https://doi.org/10.1016/j.scient.2011.07.010>. <http://www.sciencedirect.com/science/article/pii/S1026309811001246>
17. Kong, Y., Zhu, Y., Wang, Y.: A center-based modeling approach to solve the districting problem. *Int. J. Geograp. Inf. Sci.* **33**(2), 368–384 (2019). <https://doi.org/10.1080/13658816.2018.1474472>
18. Laporte, G., Nickel, S., Saldanha da Gama, F. (eds.): *Location Science*. Springer International Publishing, Cham (2015). <https://doi.org/10.1007/978-3-319-13111-5>
19. López, C.O., Beasley, J.E.: Packing a fixed number of identical circles in a circular container with circular prohibited areas. *Optimiz. Lett.* **13**(7), 1449–1468 (2019). <https://doi.org/10.1007/s11590-018-1351-x>
20. Ma, S.: Resilient jammed packing: A novel feature of a classic geometry problem. *SIAM Undergraduate Research Online* **11**, (2018). <https://doi.org/10.1137/18s016667>
21. Mallozzi, L., Puerto, J.: The geometry of optimal partitions in location problems. *Optimiz. Lett.* **12**(1), 203–220 (2018). <https://doi.org/10.1007/s11590-017-1156-3>
22. Martin, D.A., Moody, J.A.: Comparison of soil infiltration rates in burned and unburned mountainous watersheds. *Hydrol. Process.* **15**(15), 2893–2903 (2001). <https://doi.org/10.1002/hyp.380>
23. OU Cooperative Institute for Mesoscale Meteorological Studies, NOAA National Severe Storms Laboratory: Flash - data page (2016). <http://flash.ou.edu/new/>. [Last accessed 09-01-2020]
24. Robichaud, P.: Fire effects on infiltration rates after prescribed fire in Northern Rocky Mountain forests, USA. *J. Hydrol.* **231–232**, 220–229 (2000). [https://doi.org/10.1016/S0022-1694\(00\)00196-7](https://doi.org/10.1016/S0022-1694(00)00196-7)
25. Scott, D., Van Wyk, D.: The effects of wildfire on soil wettability and hydrological behaviour of an afforested catchment. *J. Hydrol.* **121**(1), 239–256 (1990). [https://doi.org/10.1016/0022-1694\(90\)90234-O](https://doi.org/10.1016/0022-1694(90)90234-O)
26. Staley, D.M., Negri, J.A., Kean, J.W., Laber, J.L., Tillery, A.C., Youberg, A.M.: Prediction of spatially explicit rainfall intensity–duration thresholds for post-fire debris-flow generation in the western United States. *Geomorphology* **278**, 149–162 (2017). <https://doi.org/10.1016/j.geomorph.2016.10.019>
27. Stoyan, Y., Yaskov, G.: Packing unequal circles into a strip of minimal length with a jump algorithm. *Optimiz. Lett.* **8**(3), 949–970 (2014). <https://doi.org/10.1007/s11590-013-0646-1>
28. Suzuki, A., Drezner, Z.: The p-center location problem in an area. *Location Science* **4**(1), 69–82 (1996) [https://doi.org/10.1016/S0966-8349\(96\)00012-5](https://doi.org/10.1016/S0966-8349(96)00012-5). <http://www.sciencedirect.com/science/article/pii/S0966834996000125>
29. Tran, T.C., Dinh, T.B., Gascon, V.: Meta-heuristics to Solve a Districting Problem of a Public Medical Clinic. In: *Proceedings of the Eighth International Symposium on Information and Communication Technology - SoICT 2017*, pp. 127–134. ACM Press, Nha Trang City, Viet Nam (2017). <https://doi.org/10.1145/3155133.3155146>
30. Van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T.: Scikit-image: Image processing in Python. *PeerJ* **2**, e453 (2014)
31. Westerling, A.L., Hidalgo, H.G., Cayan, D.R., Swetnam, T.W.: Warming and earlier spring increase western U.S. forest wildfire activity. *Science* **313**(5789), 940–943 (2006). <https://doi.org/10.1126/science.1128834>