# The complexity of finding optimal subgraphs to represent spatial correlation

Jessica Enright[1][0000−0002−0266−3292], Duncan Lee[2][0000−0002−6175−6800], Kitty Meeks[1][0000−0001−5299−3073], William Pettersson[1][0000−0003−0040−2088], and John Sylvester[1][0000−0002−6543−2934]

[1] School of Computing Science, University of Glasgow
[2] School of Mathematics and Statistics, University of Glasgow
firstname.lastname@glasgow.ac.uk

**Abstract.** Understanding spatial correlation is vital in many fields including epidemiology and social science. Lee, Meeks and Pettersson (Stat. Comput. 2021) recently demonstrated that improved inference for areal unit count data can be achieved by carrying out modifications to a graph representing spatial correlations; specifically, they delete edges of the planar graph derived from border-sharing between geographic regions in order to maximise a specific objective function. In this paper we address the computational complexity of the associated graph optimisation problem. We demonstrate that this problem cannot be solved in polynomial time unless P = NP; we further show intractability for two simpler variants of the problem. We follow these results with two parameterised algorithms that exactly solve the problem in polynomial time in restricted settings. The first of these utilises dynamic programming on a tree decomposition, and runs in polynomial time if both the treewidth and maximum degree are bounded. The second algorithm is restricted to problem instances with maximum degree three, as may arise from triangulations of planar surfaces, but is an FPT algorithm when the maximum number of edges that can be removed is taken as the parameter.

**Keywords:** Parameterised complexity · treewidth · colour coding · spatial statistics

## 1 Introduction

Spatio-temporal count data relating to a set of $n$ non-overlapping areal units for $T$ consecutive time periods are prevalent in many fields, including epidemiology [11] and social science [1]. As geographical proximity can often indicate correlation, such data can be modelled as a graph, with vertices representing areas and edges between areas that share a geographic boundary and so are assumed to be correlated. The count data is then represented as a weight assigned to each vertex. However, such models are often not ideal representations as geographical

proximity does not always imply correlation [9]. Instead, Lee, Meeks and Pettersson [7] recently proposed a new method for addressing this issue by deriving a specific objective function (given in full in Section 2.2), and then searching for a spanning subgraph with no isolated vertices which maximises this function. Maximising this objective function corresponds to maximising the natural log of the product of full conditional distributions over all vertices (corresponding to spatial units) in a conditional autoregressive model. Such models are typically written as a series of univariate full conditional distributions rather than a joint distribution. This objective function is highly non-linear, and rewards removing as few edges as possible, while applying a penalty that (non-linearly) increases as the difference between the weight of each vertex and the average weight over its neighbours increases. Due to the size of the data, exhaustive searches for optimal subgraphs are intractable and so efficient algorithms are required for this problem. Lee, Meeks and Pettersson [7] gave a heuristic for this problem, but point out that many standard techniques are not applicable to this problem, suggesting that this problem is hard to solve efficiently in general.

### 1.1 Our contribution

We show that the problem is indeed NP-hard, even on planar graphs, and provide examples that illustrate two of the major challenges inherent in the problem: we cannot optimise independently in disjoint connected components and we cannot iterate towards a solution. We also show that the decision variant of minimising the penalty portion of the objective function is NP-complete even when restricted to planar graphs with maximum degree at most five. We then investigate a simplification in which the goal is to find a subgraph with a penalty term of zero. We show that this is solvable linear time and space in the number of edges of the graph, and we completely characterise all such subgraphs. However, we also show that finding a subgraph with a penalty term of zero on all vertices of degree two or more is NP-complete.

In the positive direction, we give two exact algorithms that are tractable in their respective restricted settings. These both require that the input graph have bounded maximum degree: we note that graphs arising from areal studies will often have small maximum degree. The first algorithm runs in polynomial time if both the maximum degree and treewidth of the underlying graph are bounded. The second algorithm is only guaranteed to be correct if the underlying graph has maximum degree three, but is fixed-parameter tractable when parameterised by the maximum number of edges that can be removed.

### 1.2 Paper outline

Section 2 gives notation and definitions, the formal problem definition, and examples that illustrate two of the major challenges inherent in the problem. We then prove in Section 3 that, unless P=NP, there is no polynomial-time algorithm to solve the main optimisation problem, even when restricted to planar graphs. Section 4 then examines three simplifications of the problem. In Section 5 we

introduce two algorithms to exactly solve the problem in certain special cases, and we finish with concluding thoughts and open problems in Section 6. Note that some details and proofs are omitted due to space constraints.

## 2 Background

In this section we give the notation we need for this paper, define the problem, and then demonstrate why some common techniques from graph theory are not applicable to this problem.

### 2.1 Notation and definitions

A graph is a pair $G = (V, E)$, where the *vertex set* $V$ is a finite set, and the *edge set* $E \subseteq V^{(2)}$ is a set of unordered pairs of elements of $V$. Two vertices $u$ and $v$ are said to be *adjacent* if $e = uv \in E$; $u$ and $v$ are said to be the *endpoints* of $e$. The *neighbourhood* of $v$ in $G$ is the set $N_G(v) := \{u \in V : uv \in E\}$, and the *degree* of $v$ in $G$ is $d_G(v) := |N_G(v)|$. An *isolated vertex* is a vertex of degree zero, and a *leaf* is a vertex of degree one. The *maximum degree* of a graph $G$ is $\Delta(G) := \max_{v \in V} d_G(v)$. A graph $H = (V_H, E_H)$ is a *subgraph* of $G$ if $V_H \subseteq V$ and $E_H \subseteq E$; $H$ is a *spanning subgraph* of $G$ if $V_H = V$ so that $H$ is obtained from $G$ by deleting a (possibly empty) subset of edges. Given an edge $e$ in $E(G)$ (respectively a set $E' \subseteq E(G)$) we write $G \setminus e$ (respectively $G \setminus E'$) for the subgraph of $G$ obtained by deleting $e$ (respectively deleting every element of $E'$). A graph $G$ is *planar* if it can be drawn in the plane (i.e. vertices can be mapped to points in the plane, and edges to curves in the plane whose extreme points are the images of its endpoints) in such a way that no two edges cross. Given any partition of a subset of the plane into regions, we can define a planar graph whose vertices are in bijection with the set of regions, in which two regions are adjacent if and only if they share a border of positive length. In particular, if each region has three sides (i.e, the partition is a triangulation of a subset of the plane) then the resulting graph will have maximum degree three.

### 2.2 The optimisation problem

Following Lee, Meeks and Pettersson [7], we are concerned with the following optimisation problem.

**Correlation Subgraph Optimisation**
*Input:* A graph $G = (V, E)$ where $|V| = n$, and function $f : V \to \mathbb{Q}$.
*Question:* What is the maximum value of

$$\text{score}(H, f) := \sum_{v \in V} \ln d_H(v) - n \ln \left[ \sum_{v \in V} d_H(v) \left( f(v) - \frac{\sum_{u \in N_H(v)} f(u)}{d_H(v)} \right)^2 \right],$$

taken over all spanning subgraphs $H$ of $G$ such that $d_H(v) \geq 1$ for all $v \in V$?

We will say that a subgraph $H$ of $G$ is *valid* if $H$ is a spanning subgraph of $G$ and $d_H(v) \geq 1$ for all $v \in V$. Given a vertex $v$ in the input graph $G$, we will sometimes refer to $f(v)$ as the *weight* of $v$. We also define the neighbourhood discrepancy of a vertex $f$ in a graph $H$ with weight function $f$ (written $\mathrm{ND}_H(v, f)$) as

$$\mathrm{ND}_H(v, f) := \left( f(v) - \frac{\sum_{u \in N_H(v)} f(u)}{d_H(v)} \right)^2 .$$

### 2.3 Why common graph algorithm techniques fail

This problem is particularly resistant to many approaches common in algorithmic graph theory. We will describe two of these now. Firstly, on a disconnected graph $G$, combining optimal solutions on each connected component is not guaranteed to find an optimal solution on $G$. This is true even if there are only two disconnected components, one of which is an isolated edge and the other being a path, as illustrated in the following example.

*Example 1.* Consider the graph $G$ consisting of a path on four vertices ($v_1, v_2, v_3$, $v_4$) along with an isolated edge between vertices $v_a$ and $v_b$, as shown in Figure 1, and let $H = G \setminus \{v_2v_3\}$. Note that $H$ is the only proper subgraph of $G$ which has no isolated vertices. Let $f$ be defined as follows: $f(v_1) = 0$, $f(v_2) = 1$, $f(v_3) = 10$, $f(v_4) = 11$, $f(v_a) = 0$, and $f(v_b) = x$ for some real $x$. If $x = 1$ then $\mathrm{score}(G, f) < \mathrm{score}(H, f)$ but if $x = 1000$ then $\mathrm{score}(G, f) > \mathrm{score}(H, f)$.

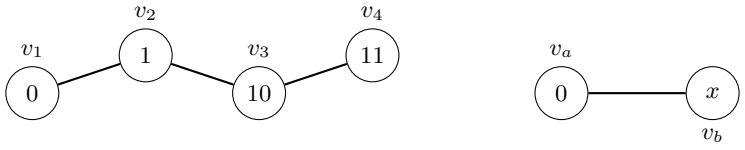


**Fig. 1.** Graph for Example 1. The value of the function at each vertex is shown inside the respective vertex.

To understand why disconnected components can affect each other in such a manner, note that the negative term in the score function contains a logarithm of a sum of neighbourhood discrepancies. This means that the relative importance of the neighbourhood discrepancy of any set of vertices depends on the total sum of the neighbourhood discrepancies across the whole graph. In other words, the presence of a large neighbourhood discrepancy elsewhere (even in a separate component) in the graph can reduce the impact of the neighbourhood discrepancy at a given vertex or set of vertices. However, the positive term in the score function is a sum of logarithms, so the contribution to the positive term from the degree of one vertex does not depend on any other part of the graph.

A reader might also be tempted to tackle this problem by identifying a "best" edge to remove and proceeding iteratively. The following example highlights that any algorithm using such a greedy approach may, in some cases, not find an optimal solution.

*Example 2.* Consider the graph $G$ being a path on six vertices labelled $v_1, v_2, v_3$, $v_4, v_5$, and $v_6$ with $f(v_1) = 1000$, $f(v_2) = 2000$, $f(v_3) = 1999$, $f(v_4) = 1001$, $f(v_5) = 2019$, and $f(v_6) = 981$ as shown in Figure 2. Let $H = G \setminus \{v_2v_3, v_4v_5\}$, and let $H' = G \setminus \{v_3v_4\}$. The maximum score that can be achieved with the removal of only one edge is achieved by removing edge $v_3v_4$ and creating $H'$. However, the optimal solution to Correlation Subgraph Optimisation on $G$ is $H$, and involves removing edges $v_2v_3$ and $v_4v_5$.

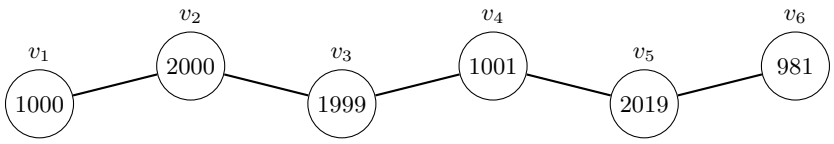


**Fig. 2.** Graph for Example 2. The value of the function at each vertex is shown inside the respective vertex.

## 3 Hardness on planar graphs

In this section we prove NP-hardness of Correlation Subgraph Optimisation on planar graphs.

**Theorem 1.** *There is no polynomial-time algorithm to solve* Correlation Subgraph Optimisation *on planar graphs unless P=NP.*

We prove this result by means of a reduction from the following problem, shown to be NP-complete in [10]; the *incidence graph* $G_\Phi$ of a CNF formula $\Phi$ is a bipartite graph whose vertex sets correspond to the variables and clauses of $\Phi$ respectively, and in which a variable $x$ and clause $C$ are connected by an edge if and only if $x$ appears in $C$.

**Cubic Planar Monotone 1-in-3 SAT**
*Input:* A 3-CNF formula $\Phi$ in which every variable appears in exactly three clauses, variables only appear positively, and the incidence graph $G_\Phi$ is planar.
*Question:* Is there a truth assignment to the variables of $\Phi$ so that exactly one variable in every clause evaluates to TRUE?

We begin by describing the construction of a graph $G$ and function $f : V(G) \to \mathbb{N}$ corresponding to the formula $\Phi$ in an instance of Cubic Planar Monotone 1-in-3 SAT; the construction will be defined in terms of an integer parameter $t \geq 1$ whose value we will determine later. Note that $G$ is not the incidence graph $G_\Phi$ of $\Phi$.

Suppose that $\Phi$ has variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$. Since every variable appears in exactly three clauses and each clause contains exactly three variables, we must have $m = n$. For each variable $x_i$, $G$ contains a variable gadget on $3t^2 + 6t + 8$ vertices. The non-leaf vertices of the gadget are:

- $u_i$, with $f(u_i) = 7t$,
- $v_i$, with $f(v_i) = 4t$,
- $z_i$ with $f(z_i) = t$,
- $z'_i$ with $f(z'_i) = 4t$, and
- $w_{i,j}$ for each $j \in \{1, 2, 3\}$, with $f(w_{i,j}) = 3t$.

The vertex $v_i$ is adjacent to $u_i$, $z_i$ and each $w_{i,j}$ with $i \in \{1, 2, 3\}$; $z_i$ is adjacent to $z'_i$. We add leaves to this gadget as follows:

- $u_i$ has $3t$ pendant leaves, each assigned value $7t + 1$ by $f$;
- $z_i$ has $3t$ pendant leaves, each assigned value $t - 1$ by $f$;
- $z'_i$ has $3t^2$ pendant leaves, each assigned value $4t$ by $f$;
- each vertex $w_{i,j}$ has exactly one pendant leaf, assigned value $3t$ by $f$.

For each clause $C_j$, $G$ contains a clause gadget on $t^2 + 2$ vertices: $a_j$ and $a'_j$, which are adjacent, and $t^2$ pendant leaves adjacent to $a'_j$. We set $f(a_j) = 2t$, and $f$ takes value $t$ on $a'_j$ and all of its leaf neighbours. We complete the definition of $G$ by specifying the edges with one endpoint in a variable gadget and the other in a clause gadget: if the variable $x_i$ appears in clauses $C_{r_1}$, $C_{r_2}$ and $C_{r_3}$, with $r_1 < r_2 < r_3$, then we have edges $w_{i,1}a_{r_1}$, $w_{i,2}a_{r_2}$ and $w_{i,3}a_{r,3}$. The construction of the variable and clause gadgets is illustrated in Figure 3.

Recall that a subgraph $H$ of $G$ is *valid* if $H$ is a spanning subgraph of $G$ and $d_H(v) \geq 1$ for all $v \in V$. Recall that the *neighbourhood discrepancy* of a vertex $v$ with respect to $f$ in a valid subgraph $H$, written $\mathrm{ND}_H(v, f)$, is

$$\mathrm{ND}_H(v, f) := \left( f(v) - \frac{\sum_{u \in N_H(v)} f(v)}{d_H(v)} \right)^2 .$$

The goal of Correlation Subgraph Optimisation is therefore to maximise

$$\mathrm{score}(H, f) := \sum_{v \in V} \ln d_H(v) - n \ln \left[ \sum_{v \in V} d_H(v) \, \mathrm{ND}_H(f, v) \right],$$

over all valid subgraphs $H$ of $G$. We now give several results that are necessary; the proofs of these are omitted due to space constraints but can be found in [3]. This first set of results give several properties of valid subgraphs of $G$.

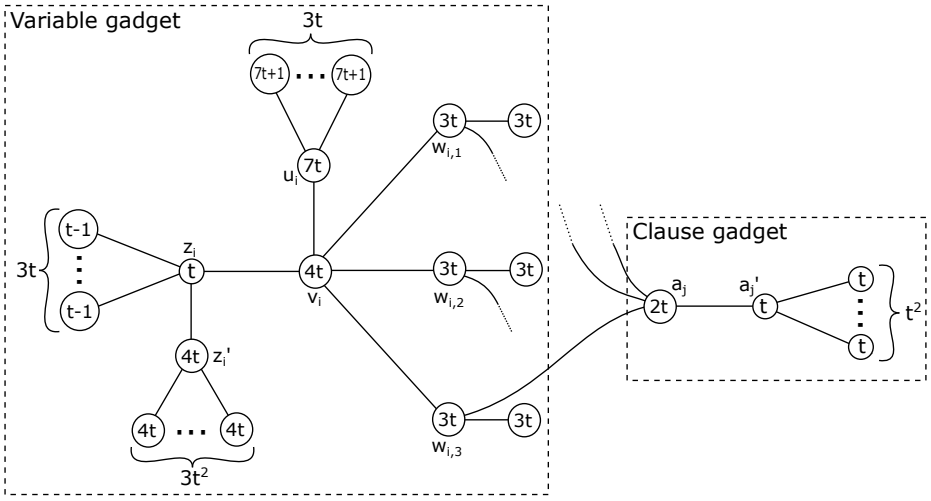


**Fig. 3.** Construction of the variable and clause gadgets.

**Lemma 1.** *For any valid subgraph $H$,*

$$\sum_{u \text{ a leaf in } G} \mathrm{ND}_H(u, f) = 6nt.$$

**Lemma 2.** *For any valid subgraph $H$,*

$$0 \leq \mathrm{ND}_H(z_i', f), \mathrm{ND}_H(a_i', f) < 1/t^2.$$

**Lemma 3.** *For any valid subgraph $H$,*

$$\sum_{v \in V} \ln d_H(v) \geq 6n \ln t + 2n.$$

**Lemma 4.** *Let $H$ be any subgraph of $G$ (not necessarily valid). Then*

$$\sum_{v \in V} \ln d_H(v) \leq 6n \ln t + 20n.$$

We now give two lemmas that relate the existence of truth assignments of a 3-CNF formulae to bounds on the neighbourhood discrepancies of some vertices within a valid subgraph $H$.

**Lemma 5.** *If $\Phi$ is satisfiable, there is a valid subgraph $H$ such that for all $v \in V \setminus \{z_i', a_i' : 1 \leq i \leq n\}$ with $d_G(v) > 1$ we have $\mathrm{ND}(v, H) = 0$.*

**Lemma 6.** *If $\Phi$ is not satisfiable, then for any valid subgraph $H$, there exists a vertex $v \in V \setminus \{z_i', a_i' : 1 \leq i \leq n\}$ with $d_G(v) > 1$ such that*

$$\mathrm{ND}_H(v, f) \geq t^2/9.$$

We now give bounds on the possible values for $\mathrm{score}(H, f)$ depending on whether or not $\Phi$ is satisfiable.

**Lemma 7.** *If $\Phi$ is satisfiable, there is a valid subgraph $H$ with*

$$\mathrm{score}(H, f) \geq 6n \ln t - n \ln(12nt).$$

**Lemma 8.** *If $\Phi$ is not satisfiable, then for every valid subgraph $H$ we have*

$$\mathrm{score}(H, f) \leq 6n \ln t + 20n - n \ln(t^2/9).$$

We are now ready to prove Theorem 1, which we restate here for convenience.

**Theorem 1.** *There is no polynomial-time algorithm to solve* Correlation Subgraph Optimisation *on planar graphs unless P=NP.*

*Proof.* We suppose for a contradiction that there is a polynomial-time algorithm $\mathcal{A}$ to solve Correlation Subgraph Optimisation on planar graphs, and show that this would allow us to solve Cubic Planar Monotone 1-in-3 SAT in polynomial time.

Given an instance $\Phi$ of Cubic Planar Monotone 1-in-3 SAT, where we will assume without loss of generality that $\Phi$ has $n > e^{47}$ variables, we proceed as follows. First construct $(G, f)$ as defined above, taking $t = n^2$; it is clear that this can be done in polynomial time in $|\Phi|$. Note that $G$ is planar: to see this, observe that repeatedly deleting vertices of degree one gives a subdivision of the incidence graph which is planar by assumption. We then run $\mathcal{A}$ on $(G, f)$ and return YES if the output is at least $\frac{17}{2} n \ln n$, and NO otherwise.

It remains to demonstrate that this procedure gives the correct answer. Suppose first that $\Phi$ is satisfiable. In this case, by Lemma 7, we know that there exists a subgraph $H$ of $G$ with

$$\begin{aligned}
\mathrm{score}(H, f) &\geq 6n \ln t - n \ln(12nt) \\
&= 6n \ln n^2 - n \ln(12n^3) \\
&= 12n \ln n - 3n \ln n - n \ln 12 \\
&\geq 9n \ln n - 3n \\
&> \frac{17}{2} n \ln n,
\end{aligned}$$

since $3 < \ln n/2$, so our procedure returns YES.

Conversely, suppose that $\Phi$ is not satisfiable. In this case, by Lemma 8 we know that, for every valid subgraph $H$ we have

$$\begin{aligned}
\mathrm{score}(H, f) &\leq 6n \ln t + 20n - n \ln(t^2/9) \\
&= 6n \ln n^2 + 20n - n \ln(n^4/9) \\
&= 12n \ln n + 20n - 4n \ln n + n \ln 9 \\
&\leq 8n \ln n + 23n \\
&< \frac{17}{2} n \ln n,
\end{aligned}$$

since $23 < \ln n/2$, so our procedure returns NO. □

# 4 Simplifications of the problem

One may wonder if the hardness of Correlation Subgraph Optimisation is due to the interplay between the two parts of the objective function. We show in Section 4.1 that just determining if there is a valid subgraph with total neighbourhood discrepancy below some given constant is NP-complete, even if the input graph is planar and has maximum degree at most five. In Section 4.2 we that show that subgraphs that have zero neighbourhood discrepancy everywhere (if they exist) can be found in time linear in the number of edges, however determining if there exists a subgraph that has zero neighbourhood discrepancy everywhere excluding leaves is NP-complete.

## 4.1 Minimising neighbourhood discrepancy

Consider the following problem, which questions the existence of a subgraph whose total neighbourhood discrepancy is below a given constant.

**Average Value Neighbourhood Optimisation**
*Input:* A graph $G = (V, E)$, a function $f : V \to \mathbb{Q}$, and $k \in \mathbb{Q}$.
*Question:* Is there a spanning subgraph $H$ of $G$ such that $d_H(v) \geq 1$ for all $v \in V$ and

$$\sum_{v \in V} \left( f(v) - \frac{\sum_{u \in N_H(v)} f(u)}{d_H(v)} \right)^2 \quad \leq \quad k \quad ?$$

First observe that the Average Value Neighbourhood Optimisation is clearly in NP. The NP-hardness of Average Value Neighbourhood Optimisation can be shown by giving a reduction from Cubic Planar Monotone 1-in-3 SAT, which we used earlier in Section 3. The full proof is omitted due to space constraints but can be found in [3].

**Theorem 2.** Average Value Neighbourhood Optimisation *is NP-complete, even when restricted to input graphs $G$ that are planar and have maximum degrees at most five.*

## 4.2 Ideal and near-ideal subgraphs

An obvious upper-bound to score$(H, f)$ is given by $\sum_{v \in V(H)} \ln d_H(v)$ (i.e. assume every vertex has zero neighbourhood discrepancy), so a natural question to ask is whether, for a given graph $G$ and function $f$, a valid subgraph $H$ of $G$ can be found that achieves this bound. In such a graph, it must hold that $\mathrm{ND}_H(v, f) = 0$ for every $v \in V(H)$. We say such a graph $H$ is *$f$-ideal* (or simply ideal, if $f$ is clear from the context). We now show that this definition is equivalent to saying that a graph $H$ is $f$-ideal if and only the restriction of $f$ to any connected component of $H$ is a constant-valued function.

**Theorem 3.** *A graph $H$ is $f$-ideal if and only if for each connected component $C_i$ in $H$ there exists some constant $c_i$ such that $f(v) = c_i$ for all $v \in V(C_i)$.*

*Proof.* Let $P$ denote a path of maximal length in an $f$-ideal graph such that the weights of the vertices of $P$ strictly increase as one follows the path. In an ideal graph, any edge between vertices of different weights means that $P$ must contain at least two distinct vertices, however the first and last vertices in such a path cannot have zero neighbourhood discrepancy. Thus, no such path on one or more edges can exist in an ideal graph, so a graph $G$ is ideal if and only if for each connected component $C_i$ in $G$ there exists some constant $c_i$ such that $f(v) = c_i$ for all $v \in V(C_i)$. □

Thus, ideal subgraphs can be found by removing any edge $uv$ if $f(u) \neq f(v)$ (in $O(|E|) = O(n^2)$ time), and if necessary we can test if such a graph has no isolated vertices (and thus is valid) quickly. The proof of Theorem 3 highlights that maximal paths with increasing weights must start and end on vertices that do not have zero neighbourhood discrepancy, so one might be tempted to relax the ideal definition to only apply on vertices that are not leaves. We therefore say a graph $H$ is *$f$-near-ideal* if $\mathrm{ND}_H(v, f) = 0$ for every $v \in V(H)$ with $d_H(v) \geq 2$. In other words, we now allow non-zero neighbourhood discrepancy, but only at leaves, motivating the following problem.

**Near Ideal Subgraph**
*Input:* A graph $G = (V, E)$ where $|V| = n$, and a function $f : V \mapsto \mathbb{Q}$.
*Question:* Is there a valid subgraph $H$ of $G$ such that $H$ is $f$-near-ideal?

While an ideal subgraph (if one exists) can be found quickly, it turns out that solving Near Ideal Subgraph is NP-complete, even on trees. We reduce from subset-sum, which is NP-complete [6], and which we define as follows.

**Subset Sum**
*Input:* An integer $k$, and a set of integers $S = \{s_1, s_2, \ldots, s_n\}$.
*Question:* Is there a subset $U \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{u \in U} s_u = k$?

Given an instance $(S, k)$ of Subset Sum, we will construct a graph $G$ with weight function $f$ such that $(G, f)$ has a near-ideal subgraph if and only if there is a solution to our instance of Subset Sum.

The graph $G$ contains $3n + 3$ vertices labelled as follows:

- $v_t$ for the target value, $v_s$ for a partial sum, and $v_z$ for a pendant, and
- $v_p^j$ for $p \in \{1, \ldots, n\}$ and $j \in \{1, 2, 3\}$.

Vertex $v_s$ is adjacent to vertices $v_t$, $v_z$, and $v_p^1$ for $p \in \{1, \ldots, n\}$. For each $p \in \{1, \ldots, n\}$, $v_p^1$ is adjacent to $v_p^2$, and $v_p^2$ is adjacent to $v_p^3$. This graph can be seen in Figure 4. We then define $f$ as follows:

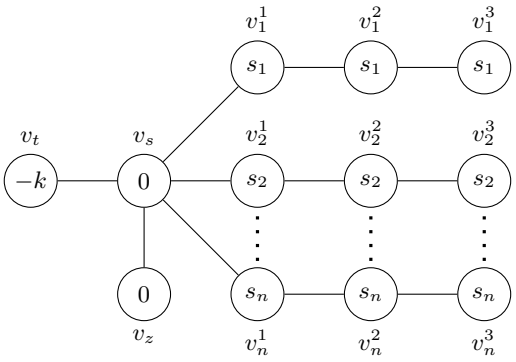


**Fig. 4.** Diagram of graph for reduction from Subset Sum. The values inside the vertices are their associated weights.

- $f(v_t) = -k$,
- $f(v_s) = f(v_z) = 0$, and
- $f(v_p^j) = s_p$ for $p \in \{1, \ldots, n\}$, and for $j \in \{1, 2, 3\}$.

Note that for the condition $d_H(v) \geq 1$ to hold for our subgraph $H$, the only edges in $G$ that might not be in $H$ are of the form $v_s v_p^1$ or $v_p^1 v_p^2$ for some $p \in \{1, \ldots, n\}$. Additionally, for any $p \in \{1, \ldots, n\}$, at most of one of $v_s v_p^1$ or $v_p^1 v_p^2$ can be removed. We can then show that $G$ has a near-ideal subgraph if and only if it is constructed from a yes-instance of Subset Sum. The complete proof is omitted due to space constraints but can be found in [3].

**Theorem 4.** Near Ideal Subgraph *is NP-complete, even if the input graph $G$ is a tree.*

## 5 Parameterised results

In this section we describe two parameterised algorithms for Correlation Subgraph Optimisation. We make use of two parameterised complexity problem classes to describe these. A problem is in the *fixed parameter tractable* (or FPT) class with respect to some parameter $k$ if the problem can be solved on inputs of size $n$ in time $f(k) \cdot n^{O(1)}$ for some computable function $f$. Note in particular that the exponent of $n$ is constant. Another class of parameterised problems is XP: a problem is in XP with respect to some parameter $k$ if the problem can be solved on inputs of size $n$ in time $O(n^{f(k)})$. In XP problems, the exponent of $n$ may change for different values of $k$, but if an upper bound on $k$ is given then this also upper bounds the exponent of $n$. For further background on parameterised complexity, see [2].

In Section 5.1 we show that Correlation Subgraph Optimisation is in XP parameterised by the maximum degree when treewidth is bounded, and is in FPT parameterised by treewidth when the maximum degree is bounded.

Then in Section 5.2 we consider the more restricted case where $G$ has maximum degree three, and show that with this restriction Correlation Subgraph Optimisation is in FPT parameterised by the number of edges that are removed. We highlight that this restriction on the maximum degree occurs naturally in triangulations of surfaces, such as can occur when discretising geographic maps.

### 5.1 An exact XP algorithm parameterised by treewidth and maximum degree

We now briefly describe an exact XP algorithm for solving Correlation Subgraph Optimisation on arbitrary graphs that leads to the following result.

**Theorem 5.** Correlation Subgraph Optimisation *can be solved in time*

$$O(2^{2\Delta(G)(tw(G)+1)} \cdot n^{2\Delta(G)+1}).$$

The algorithm follows fairly standard dynamic programming techniques on a nice tree decomposition $T$ of $G$ with treewidth $tw(G)$ that is rooted at some arbitrary leaf bag. A *nice tree decomposition* is a tree decomposition with one leaf bag selected as a root bag so that the children of a bag are adjacent bags that are further from the root, and the additional property that each leaf bag is empty, and each non-leaf bag is either a introduce bag, forget bag, or join bag, which are defined as follows. An introduce bag $\nu$ has exactly one child below it, say $\mu$, such that $\nu$ contains every element in $\mu$ as well as precisely one more element. A forget bag $\nu$ has exactly one child below it, say $\mu$, such that $\nu$ contains every element in $\mu$ except one. A join bag $\lambda$ has exactly two children below it, say $\mu$ and $\nu$, such that $\lambda$, $\mu$, and $\nu$, all have precisely the same elements. See [2], in particular Chapter 7, for an introduction to tree decompositions, and a formal definition of nice tree decompositions.

We will outline the core ideas here; full details are omitted due to space constraints but can be found in [3]. We first define some specific terminology that will be useful when describing the algorithm. Let $T$ be a tree decomposition (not necessarily nice) with an arbitrary bag labelled as the root. For each bag $\nu \in T$, denote by $G_\nu$ the induced subgraph of $G$ consisting precisely of vertices that appear in bags below $\nu$ but do not appear in $\nu$, where we take below to mean further away from the root bag. The set of edges between a vertex in $\nu$ and a vertex in $G_\nu$ will be important to our algorithm, so we will write $E_\nu = \{uv \in E(G) \mid u \in \nu \wedge v \in G_\nu\}$ to be the set of edges with one endpoint in $G_\nu$ and the other in $\nu$. An example of a graph, a tree decomposition, $G_\nu$, and $E_\nu$ are shown in Figure 5.

Our algorithm will process each bag, from the leaves towards the root, determining a set of states for each bag such that we can guarantee that the optimal solution will correspond to a state in the root bag. Given a bag $\nu$ of a tree decomposition and a set of edges $I \subseteq E_\nu$, define $\mathcal{G}'_{\nu,I}$ to be the set of graphs $G'$ with $V(G') = V(G_\nu) \cup \nu$, $E(G') \subseteq E(G)$, and for any edge $uv \in E_\nu$, $uv \in E(G')$ if and only if $uv \in I$.

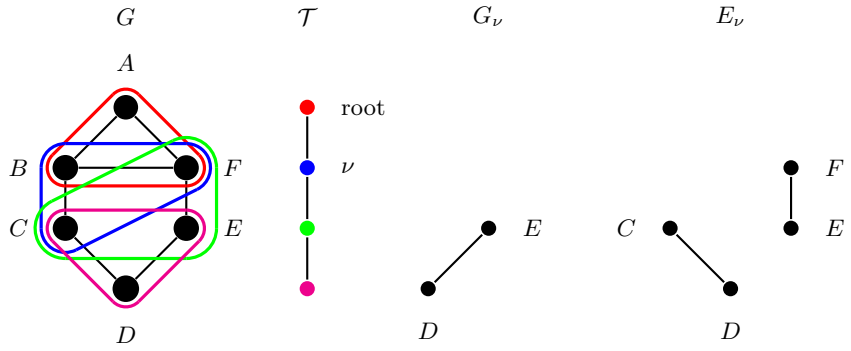


**Fig. 5.** From left to right, we have a graph $G$ with a tree decomposition displayed by circling vertices, the tree indexing a tree decomposition of $G$ drawn as a graph with the root and the bag $\nu$ labelled, the graph $G_\nu$ consisting of the induced subgraph on vertices $D$ and $E$, and the the set of edges $E_\nu = \{CD, EF\}$ (i.e., the edges of $G$ that are between a vertex in $G_\nu$ and a vertex in $\nu$).

**Definition 1.** *For a bag $\nu$, the set of all valid states at $\nu$ is*

$$S_\nu = \{(I, D) \mid I \subseteq E_\nu,\ \mathcal{G}'_{\nu,I} \neq \emptyset,\ \textit{and there exists a graph } H \in \mathcal{G}'_{\nu,I} \textit{ with } D = \sum_{v \in G_\nu} \ln d_H(v),\ \textit{and } d_H(v) \geq 1\ \forall v \in V(G_\nu)\}.$$

Each state corresponds to at least one graph ($H$ in the definition) but there may be multiple graphs that all lead to the same state. For each state we will also store the best possible (i.e., lowest) value of $\sum_{v \in H} d_H(v)\,\mathrm{ND}_H(v, f)$ (i.e., total neighbourhood discrepancy summed over vertices that only appear below the current bag) over all of the graphs $H$ that correspond to a given state. This allows us to compute the contribution to the penalty portion of the objective function from the subtree under consideration.

### 5.2 Parameterisation by $k$ in low degree graphs

We also study the problem when $G$ has maximum degree three and we want to bound the maximum number of edges that can be removed. In this setting we define $k$-Correlation Subgraph Optimisation and show that it is in FPT when parameterised by $k$, the maximum number of edges that can be removed from $G$ to create $H$. This setting is of interest as the dual graph of any triangulation has maximum degree three and triangulations are often used to represent discretised surfaces [5, 8].

$k$-Correlation Subgraph Optimisation
*Input:* A graph $G = (V, E)$ where $|V| = n$, an integer $k$, and a function $f : V \to \mathbb{Q}$.
*Question:* What is the maximum value of

$$\text{score}(H, f) := \sum_{v \in V} \ln d_H(v) - n \ln \left[ \sum_{v \in V} d_H(v) \left( f(v) - \frac{\sum_{u \in N_H(v)} f(u)}{d_H(v)} \right)^2 \right],$$

taken over all spanning subgraphs $H$ of $G$ such that $|E(G \setminus H)| \leq k$ and $d_H(v) \geq 1$ for all $v \in V$?

**Theorem 6.** *For an integer $k \geq 1$, $k$-Correlation Subgraph Optimisation can be solved on graphs with maximum degree three in time $2^{k(2 \log k + O(1))} n \log n$.*

This can be proven using the following guide; full details are omitted due to space constraints but can be found in [3]. Consider in turn each possibility $R$ for the graph consisting of deleted edges, and for each such graph we consider in turn the possibilities of the degree sequence of the remaining graph. The number of distinct graphs $R$ that must be considered is independent of $n$, and for each $R$ the number of degree sequences of $G \setminus R$ is linear in $n$. As $R$ has maximum degree two and therefore consists only of paths and cycles, it has treewidth at most two. We can therefore adapt well-known colour-coding methods (see [4, Section 13.3] for more details) for finding subgraphs with bounded treewidth in FPT time so that we can identify a subgraph $R$ in $G$ whose removal gives the biggest improvement to the neighbourhood discrepancy term while still maintaining the correct degree sequence of $G \setminus R$.

## 6 Discussion and conclusions

Correlation Subgraph Optimisation is a graph optimisation problem arising from spatial statistics with direct applications to epidemiology and social science that we show is intractable unless P=NP. We also show that it is resistant to common techniques in graph algorithms, but can be solved in polynomial time if both the treewidth and maximum degree of $G$ are bounded, or if $G$ has maximum degree three and we bound the maximum number of edges that can be removed. However the question still remains as to whether Correlation Subgraph Optimisation itself is hard when the maximum degree of the input graph is bounded. We also note as an interesting open problem whether Correlation Subgraph Optimisation admits efficient parameterised algorithms with respect to (combinations of) parameters other than the maximum degree. Additionally, the original paper that introduced Correlation Subgraph Optimisation gives one heuristic for solving the problem, but leaves open any guarantee on the performance of this heuristic. Thus the investigation of the

performance of this heuristic, or indeed of any new approximation algorithms, form two other significant open problems for correlation subgraph optimisation.

## Acknowledgements

All authors gratefully acknowledge funding from the Engineering and Physical Sciences Research Council (ESPRC) grant number EP/T004878/1 for this work, while Meeks was also supported by a Royal Society of Edinburgh Personal Research Fellowship (funded by the Scottish Government).

## References

1. Jonathan R Bradley, Christopher K Wikle, and Scott H Holan. Bayesian spatial change of support for count-valued survey data with application to the american community survey. *Journal of the American Statistical Association*, 111(514):472–487, 2016.
2. Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 5. Springer, 2015.
3. Jessica Enright, Duncan Lee, Kitty Meeks, William Pettersson, and John Sylvester. The complexity of finding optimal subgraphs to represent spatial correlation, 2021. arXiv:2010.10314.
4. Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. *SIAM Journal on Computing*, 33(4):892–922, 2004.
5. Qingsong He, Chen Zeng, Peng Xie, Yaolin Liu, and Mengke Zhang. An assessment of forest biomass carbon storage and ecological compensation based on surface area: A case study of Hubei Province, China. *Ecological Indicators*, 90:392–400, 2018.
6. Richard M. Karp. Reducibility Among Combinatorial Problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103, New York, 1972. Plenum Press.
7. Duncan Lee, Kitty Meeks, and William Pettersson. Improved inference for areal unit count data using graph-based optimisation. *Statistics and Computing*, 31(4):51, Jun 2021.
8. Jennifer S Mindell, Paulo R Anciaes, Ashley Dhanani, Jemima Stockton, Peter Jones, Muki Haklay, Nora Groce, Shaun Scholes, Laura Vaughan, et al. Using triangulation to assess a suite of tools to measure community severance. *Journal of transport geography*, 60:119–129, 2017.
9. Richard Mitchell and Duncan Lee. Is there really a "wrong side of the tracks" in urban areas and does it matter for spatial analysis? *Annals of the Association of American Geographers*, 104(3):432–443, 2014.
10. Cristopher Moore and John Michael Robson. Hard tiling problems with simple tiles. *Discrete & Computational Geometry*, 26(4):573–590, 2001.
11. Oliver Stoner, Theo Economou, and Gabriela Drummond Marques da Silva. A hierarchical framework for correcting under-reporting in count data. *Journal of the American Statistical Association*, 114(528):1481–1492, 2019.