Computers & Industrial Engineering 60 (2011) 593–601

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A network flow model for clustering segments and minimizing total maintenance and rehabilitation cost

I-Lin Wang [a,*], Yi-Chang James Tsai [b], Feng Li [b]

[a] Department of Industrial and Information Management, National Cheng Kung University, No. 1, University Rd., Tainan 701, Taiwan
[b] School of Civil and Environmental Engineering, Georgia Institute of Technology, 210 Technology Circle, Savannah, GA 31407, United States

ARTICLE INFO

Article history:
Received 1 June 2010
Received in revised form 20 December 2010
Accepted 3 January 2011
Available online 13 January 2011

Keywords:
Pavement optimization
Spatial clustering
Uncapacitated facility location problem
Shortest path
Integer programming

ABSTRACT

Because of shrinking budgets, transportation agencies are facing severe challenges in the preservation of deteriorating pavements. There is an urgent need to develop a methodology that minimizes maintenance and rehabilitation (M&R) cost. To minimize total network M&R cost of clustering pavement segments, we propose an integer programming model similar to an uncapacitated facility location problem (UFLP) that clusters pavement segments contiguously. Based on the properties of contiguous clustered pavement segments, we have transformed the clustering problem into an equivalent network flow problem in which each possible clustering corresponds to a path in the proposed acyclic network model. Our proposed shortest-path algorithm gives an optimal clustering of segments that can be calculated in a time polynomial to the number of segments. Computational experiments indicate our proposed network model and algorithm can efficiently deal with real-world spatial clustering problems.

1. Introduction

Transportation agencies, such as the Georgia Department of Transportation (GDOT) in the USA, are facing the challenge of preserving deteriorating pavements with a shrinking budget. GDOT is responsible for maintaining 18,000-mile interstate and state highway system by contracting out projects based on different pavement conditions that require different Maintenance and Rehabilitation (M&R) methods. M&R methods and treatment methods are used interchangeably in this paper.

Each pavement M&R project ("project") consists of a group of contiguous road segments; a segment is typically one mile or less and has had a pavement condition survey revealing the pavement distress type, severity, and extent. This survey is performed annually at the segment level using a 100-ft sample section that is used to represent the overall pavement condition. Ten distress types, rutting, load cracking, block cracking, reflection cracking, patches/potholes, ravelling, edge distress, bleeding/flushing, corrugations/pushing, and loss of section, are measured. Note that even if several types of distresses may appear in one segment, a single measure (see Álvarez, López-Rodríguez, Canito, Moral, & Camacho, 2007) on the distress condition for the entire segment will be used.

Based on the distress condition of a segment, a proper M&R method is determined, and contiguous segments are clustered into a pavement M&R project. Ideally, segments are clustered by the same pavement condition that requires the same M&R method. Unfortunately, adjacent pavement segments can deteriorate at different rates, exhibit different distresses, and, consequently, require different treatment methods. Currently, the best treatment method (e.g. the one with the highest treatment unit cost) will be applied to all segments in a pavement preservation project (i.e. a cluster). The challenge is to cluster segments needing the same treatment into cost effective pavement M&R projects. We call this a *segment clustering problem* (SCP). Finding the most cost-effective segment clustering helps transportation agencies preserve more roads with a limited budget.

Fig. 1a illustrates the example of an SCP in which eight segments having different treatment costs, numbered from 1 to 8 from left to right, are to be clustered. Fig. 1b shows segments 1 and 2, 3 to 6, and 7 and 8, are clustered into three projects; Fig. 1c illustrates an alternative SCP. Currently, the best treatment method (i.e. the most expensive) is applied to all segments in a cluster. For example, when segments 1 and 2 are clustered, the most expensive treatment (i.e. the treatment on segment 2) will be applied on both segments. This will result in a higher total M&R cost as shown in the shaded areas in Fig. 2b and c. The objective of SCP is, thus, to find the cluster combination that will minimize M&R cost.

Mathematically, a k-cluster combination divides the m segments into k clusters, obtained by placing $k-1$ separators over the $m-1$ internal segment boundaries. Thus, a k-cluster for m segments may have C_{k-1}^{m-1} possible combinations, and there can be $\sum_{k=1}^{m} C_{k-1}^{m-1} = 2^{m-1}$ cluster combinations for all possible values of k.

* Corresponding author. Tel.: +886 6 2757575x53123; fax: +886 6 2362162.
E-mail address: ilinwang@mail.ncku.edu.tw (I.-L. Wang).

0360-8352/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2011.01.001

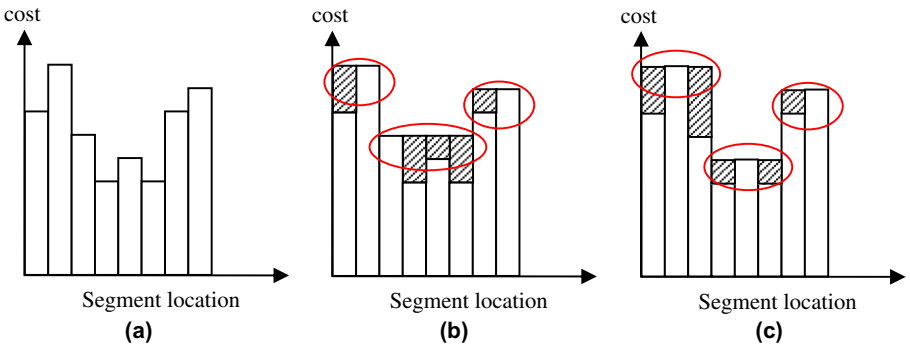

Fig. 1. An example of cost distribution along segment location and its different segment grouping results.

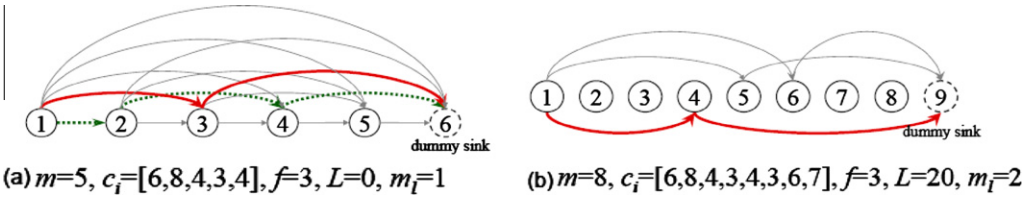

Fig. 2. Two examples of *SG* and results by TOSC.

In other words, the solution space of the SCP grows exponentially with respect to the number of segments.

Most pavement optimization research has focused on determining the optimal treatment timing for all segments in a project without considering their condition variation or corresponding spatial relationship. Optimization techniques, including the mixed integer programming (MIP) with branch-and-bound and greedy heuristic algorithms (Ouyang & Madanat, 2004) and the analytical solution (Ouyang & Madanat, 2006) have been proposed for solving the best individual segment treatment schedule. In addition, linear programming (Abaza, 2007; Golabi, Kulkarni, & Way, 1982; Grivas, Ravirala, & Schultz, 1993) and integer programming methodologies (Al-Subhi, Johnston, & Farid, 1990; Dahl & Minken, 2008; Fwa & Chan, 2000; Jacobs, 1992) have also been used for determining adequate treatment strategies at the network level.

Other than developing techniques that calculates an exact optimal solution, a few researchers (Tsai, Yang, & Wang, 2006; Yang, Tsai, & Wang, 2009) have explored spatial distribution of segments by a Fuzzy c-means method (FCM) to cluster segments with minimal rating variation for determining the best pavement treatment strategy. In their studies, the spatial clustering problem was formulated to minimize the variation of the segment ratings (a composite pavement condition indicator) in a pavement M&R project (or a cluster) and, thus, indirectly minimize the treatment cost for a whole project. Their proposed FCM first performs spatial analysis to determine several workzones based on the hard and soft barriers. For each workzone, FCM iteratively estimates the possible range of cluster numbers for each workzone, performs the unconstrained FCM clustering method, keeps adjusting those segments from violating constraints until all constraints are satisfied, calculates the objective value as the optimal result for the current cluster number, increases the current cluster number by 1 until the cluster number reaches a specified threshold, and selects the cluster settings of the minimum objective value as the best clustering for that workzone. These steps are repeated for other workzones. Note that FCM gives no guarantee of its solution quality and running time, although it may work well in practice.

This paper extends the previous works by directly using segment treatment cost to address the spatial clustering problem that can minimize the network-wide total M&R cost. This paper is the first to minimize the total network-level pavement M&R cost by clustering pavement segments with similar condition into a project.

In this paper, we follow the techniques of previous researchers to formulate the clustering problem as a specialized, uncapacitated facility location problem (UFLP), and we use additional ordering constraints via integer-programming techniques. Our study shows the same problem can be reduced to an equivalent network flow problem in which each possible segment clustering corresponds to a path in the proposed network model. In addition, this paper presents SCP and proposes a topological-ordering based shortest-path algorithm to solve SCP.

This paper is organized as follows: Section 2 introduces notations and a specialized UFLP formulation for modelling the clustering problem. In Section 3, we propose a network model to solve the clustering problem. Several numerical experiments, using real-world and simulated segment cost data, are conducted, and the results are analyzed and summarized in Section 4. Section 5 concludes the paper and suggests future research.

2. A UFLP formulation

Suppose a road is divided by a set of m contiguous segments, denoted by M, where each segment $i = 1, \ldots, m$ is associated with an original treatment cost c_i. Let contiguous segments be grouped into a set of n clusters, denoted by N, where each cluster $j = 1, \ldots, n$ has an initial set-up cost f that includes mobilization cost, communication with agency cost, and on-site office cost. The minimum cost (or the lower bound) for a cluster is specified because of the minimum cost for contracting and managing a project.

Let L be the cost of the lower bound for each cluster, usually around \$250 k USD. The clustering problem seeks a way to assign each segment i to a cluster j that minimizes total cost. Let x_{ij} be a binary decision variable to represent whether segment i is assigned

to cluster j (i.e. $x_{ij} = 1$) or not (i.e. $x_{ij} = 0$). The final treatment cost z_{ij} for each segment i in cluster j is, at least, c_i. Note that all the segments of the same cluster must have the same final treatment cost which is at least as large as the original treatment cost for any individual segment in that cluster. In practice, each cluster should at least be 2 miles (3.2KM) in length because it is not practical to contract out a small project after expending a fair amount of administrative and project management efforts. Similarly, each cluster should contain at least m_l segments, where $m_l = 2$ in practice. Let n^* denote the optimal number of clusters in our problem, which means clusters with indices larger than n^* have to be empty. Note that a good initial n close to n^* may effectively shorten the computational time. Nevertheless, since n^* cannot be calculated beforehand, we may set $n = \lceil m/m_l \rceil$ as an initial upper bound for the number of clusters.

Let y_j be a binary decision variable to indicate whether cluster j is empty (i.e. $y_j = 0$) or not (i.e. $y_j = 1$). Therefore, the SCP can be formulated as a 0, 1 mixed-integer linear programming problem (MIP) as follows:

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} z_{ij} + \sum_{j=1}^{n} f y_j \quad \text{(SCP)}$$

$$\text{s.t.} \sum_{j=1}^{n} x_{ij} = 1, \quad \forall i \in M \tag{1}$$

$$x_{ij} \leqslant y_j, \quad \forall i \in M,\ j \in N \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} \geqslant m_l y_j, \quad \forall j \in N \tag{3}$$

$$\sum_{j' \leqslant j} x_{ij'} + \sum_{j' > j} x_{i'j'} \leqslant 1, \quad \forall i,\ i' \in M,\ i' < i; j \in N \tag{4}$$

$$y_j \geqslant y_{j+1}, \quad \forall j = 1, \ldots, n-1 \tag{5}$$

$$\left.\begin{array}{lll} -x_{ij}Q \leqslant z_{ij} \leqslant x_{ij}Q, & \forall i \in M, & j \in N \\ Z_j - (1 - x_{ij})Q \leqslant z_{ij} \leqslant Z_j + (1 - x_{ij})Q, & \forall i \in M, & j \in N \\ Z_j - c_i \geqslant (x_{ij} - 1)Q, & \forall i \in M, & j \in N \end{array}\right\} \tag{6}$$

$$\sum_{i=1}^{m} z_{ij} \geqslant L y_j, \quad \forall j \in N \tag{7}$$

$$x_{ij}, \quad y_j \in \{0,1\}; z_{ij} \geqslant 0, \quad \forall i \in M,\ j \in N \tag{8}$$

In particular, the objective function of SCP minimizes the sum of the total treatment cost for all segments and the set-up cost for all clusters. Constraints (1) ensure each segment is assigned to a single cluster. Constraints (2) model whether a cluster j is nonempty or not by whether it contains some segment i or not. Although we can use aggregated constraints $\sum_{i=1}^{m} x_{ij} \leqslant m y_j \forall j \in N$ to replace the disaggregated constraints (2), the aggregated formulation usually causes more fractional LP relaxed solutions and leads to more branch-and-bound iterations that slow down the solution process. Therefore, we use the disaggregated constraints here instead of the aggregated ones. Constraints (3) restrict the minimum number of segments for a nonempty cluster to be at least m_l.

Constraints (4) are continuity constraints to restrict contiguous assignments for segments to clusters. In other words, if both segments and clusters are aligned by their indices in ascending orders, different assignments cannot cross over each other. Similarly, constraints (5) indicate the index of the first empty cluster should be larger than the index of the last nonempty cluster.

By setting Q as a very large number, constraints (6) model the final treatment cost z_{ij} for each segment i in cluster j based on the following two observations: (i) if $x_{ij} = 0$, then $z_{ij} = 0$; and (ii) if $x_{ij} = 1$, then $z_{ij} = Z_j \geqslant c_j$, where Z_j is the final treatment cost for each segment contained in cluster j. Note that these constraints may also be replaced by $z_{ij} \geqslant c_{i'} x_{i'j}, \forall i, i' \in M, j \in N$ which seem more concise but, in fact, take up to twice the time and storage space in our computational tests. Constraints (7) force the total treatment cost for a nonempty cluster j to be sufficiently large (i.e., at least L).

SCP is an MIP with $O(mn)$ decision variables and $O(m^2 n)$ constraints. If we view a segment as a customer and a cluster as a plant (or warehouse) location, then SCP becomes a specialized uncapacitated facility location problem (UFLP) that seeks a min-cost assignment for each customer to an uncapacitated plant. The objective function and constraints (1), (2), and (8) of SCP correspond to the objective function and constraints of UFLP, respectively. The treatment cost c_{ij} can be viewed as the transportation cost from customer i to plant j, and f_j is the fixed cost to build plant j. Each plant to be built has to serve at least 2 customers with sufficient investments by constraints (3) and (7). Moreover, each customer served by the same plant has to consume the same final transportation cost, which should be no less than its original transportation cost, by constraints (6). Note that there is no upper bound in the number of customers served by a plant; thus, a plant is uncapacitated.

The UFLP has been extensively studied in operations research literature, such as clustering analysis by Mulvey and Crowder (1979), economic lot sizing by Blide and Krarup (1977), machine scheduling and information retrieval by Hansen and Kaufman (1972), portfolio management by Beck and Mulvey (1982), and network design by Mirzain (1985). Comprehensive UFLP surveys can be found in Cornuejols, Nemhauser, and Wolsey (1990) and Krarup and Pruzan (1983). UFLP is shown to be NP-hard (see Cornuejols et al. (1990), for proof), and is usually solved by commercial optimization packages, such as CPLEX. Different heuristics, such as primal greedy algorithms (Cornuejols, Fisher, & Nemhauser, 1977; Nemhauser, Wolsey, & Fisher, 1987), dual descent algorithms (Bilde & Krarup, 1977; Erlenkotter, 1978), and tabu search algorithms (Sun, 2006) have been proposed. Approximation algorithms (Barahona & Chudak, 2000; Guha & Khuller, 1999; Shmoys, Tardos, & Aardal, 1997) have also been proposed to solve the UFLP quicker. Although these UFLP solution methods can all be applied to solve SCP with some modifications, these approaches may not be efficient for SCP since UFLP is NP-hard.

Note that the SCP constraints are mostly dominated by the continuity constraints (4) that try to remove those infeasible crossover segment assignments. Although the continuity constraints seem to complicate the UFLP, they also reduce much of the feasible solution space. In fact, this special property of contiguous segments in a cluster is called *string property*, named by Vinod (1969) as follows: A partition $\{C_1, C_2, \ldots, C_n\}$ of points $\{1,2,\ldots,m\}$ is contiguous if for any three points i_1, i_2, and i_3 with $1 \leqslant i_1 < i_2 < i_3 \leqslant m$ and any partition $j = 1, \ldots, n$, $i_1, i_3 \in C_j$, then $i_2 \in C_j$.

Fisher (1958) first showed that every partition that minimizes the variances within groups has to be contiguous. Vinod (1969) and Rao (1971) investigated and proposed mathematical programming formulations for some one-dimensional clustering problems with string properties. Jensen (1969) and Bellman (1973) solved clustering problems by dynamic programming. Hansen, Jaumard, and Simeone (2002) proposed a polynomial time dynamic programming algorithm to solve nested univariate clique clustering problems in which the string property may not hold even for the one-dimensional case. Recently, Novick (2009) introduced a new class of clustering problems in which the string property is extensively discussed and generalized. He points out that string property can reduce the solution space and simplify procedures for determining optimal partitioning,

although some clustering problems with string property are still NP-hard.

All previous research has dealt with objectives that minimize the sum of within-cluster interpoint distances, which is different than our approach. In mathematical programming, the complexity of problems may be largely affected by the objective function. To the best of our knowledge, the objective function of SCP has not been investigated in literature. The string property in literature is usually a result of optimal clustering for specific objective functions. On the other hand, the string property of SCP is enforced by the problem's characteristics, such as conducting pavement treatment. If we look at SCP from the 0,1 MIP point of view, it is still an open question whether SCP is polynomial-time solvable. In the next section, we will resolve this open question by looking at SCP from a totally different viewpoint in which we discard the complicated 0,1 MIP formulation, propose a new network flow model for SCP that maps a feasible clustering to a simple path, and then solve it by a polynomial-time, shortest-path algorithm.

3. Proposed network model and solution method

Although many clustering problems can be formulated and solved by integer programming (e.g. Hansen & Jaumard, 1997; Vinod, 1969), the string property of SCP can be used to develop solution methods that are more efficient than MIP. In particular, all the segments inside a cluster have to be adjacent (i.e. have contiguous indices); thus, each possible cluster can be represented by its first and last segments since all the segments in between, if any, have to be also in the same cluster.

Let the road be divided by a set of m contiguous segments where each segment $i = 1, \ldots, m$ is associated with an original treatment cost c_i. Single or contiguous segments will be grouped into clusters. Let s and t be the first and last segment of cluster $[s, t]$, where $1 \leqslant s \leqslant t \leqslant m$; then, all the segments with indices $i \in [s, t]$ will also be grouped in cluster $[s, t]$; there are, at most, $\sum_{s=1}^{m}\sum_{t=s}^{m} 1 = \frac{1}{2}m(m+1)$ possible clusters.

In SCP, even if the original treatment costs of segments in the same cluster are different, their final treatment costs have to be the same and no less than their original treatment costs. In other words, each segment $i \in [s, t]$ in the same cluster $[s, t]$ will receive the same final treatment with cost $\widehat{Z}(s,t) = \max_{i\in[s,t]}\{c_i\}$. If we assume each cluster will incur a constant initial set-up cost (or mobilization cost) f, then the total cost for cluster $[s, t]$ can be calculated by $C(s,t) = (t-s+1)\cdot\widehat{Z}(s,t) + f$. In practice, the government might require the total cost for a cluster to be larger than or equal to a constant minimum amount L to avoid the generation of too many small clusters. Therefore, if the road is grouped into n clusters $[s_j, t_j]$ for $j = 1, \ldots, n$, where $s_1 = 1$, $s_{j+1} = t_j + 1$ for $j = 1, \ldots, n-1$, $t_j \geqslant s_j$ for $j = 1, \ldots, n$, and $t_n = m$, then its total cost equals to $\sum_{j=1}^{n} C(s_j, t_j)$. Our objective is to seek the minimum total cost over all the possible clustering number $n \in [1, m]$ and combinations $[s_j, t_j]$, while satisfying treatment requirements.

3.1. Segment graph and its mathematical properties

We propose a network model called *segment graph* $SG = (V, E)$ to represent an SCP. In particular, let node set $V = \{v_i: i = 1, \ldots, m+1\}$ represent the set of m segments (i.e. $v_1, \ldots, v_m$) and a dummy sink node v_{m+1}. Here v_{m+1} represents the end of a clustering. Each *cluster arc* $(v_s, v_t) \in E$ with cost $\hat{c}_{st} = C(s, t-1) \geqslant L$ for each s and t such that $1 \leqslant s \leqslant s + m_l - 1 < t \leqslant m+1$ represents a qualified cluster $[s, t-1]$. Several properties related with $SG = (V, E)$ are as follows:

Lemma 1

(a) A segment graph $SG = (V, E)$ contains at most $|V| = O(m)$ nodes and $|E| = O(m^2)$ arcs.
(b) Constructing $SG = (V, E)$ takes $O(m^2)$ time and storage space.

Proof

(a) It is clear that there are totally $m + 1 = O(m)$ nodes. Furthermore, there are at most $\sum_{s=1}^{m}\sum_{t=s+1}^{m+1} 1 = \frac{1}{2}m(m+1) = O(m^2)$ arcs.
(b) Trivial. □

Lemma 2

(a) The arcs on a $v_1 - v_{m+1}$ path in SG correspond to a feasible clustering to the SCP.
(b) The length for a $v_1 - v_{m+1}$ path equals to the total cost of that clustering.

Proof

(a) Given a $v_1 - v_{m+1}$ path in SG, all the arcs on that path are qualified clusters to the SCP; also, their order obeys successive relationships, which means all the segments will be included without gaps or overlaps. Thus, the arcs on each $v_1 - v_{m+1}$ path correspond to a feasible clustering.
(b) This is trivial since the length of each $v_1 - v_{m+1}$ path equals the sum of the arc lengths, which equals the sum of the final cost associated with its corresponding clusters lying on the path. □

Lemma 3

(a) The segment graph SG is acyclic.
(b) A shortest $v_1 - v_{m+1}$ path in SG can be identified in $O(m^2)$ time.

Proof

(a) Since each cluster arc $(v_s, v_t) \in E$ always points from a tail node of smaller subscript indices to larger subscript indices by $1 \leqslant s \leqslant s + m_l - 1 < t \leqslant m + 1$, thus SG is acyclic, there exists no directed cycle and SG is acyclic.
(b) Since SG is acyclic by (a), a shortest $v_1 - v_{m+1}$ path can be identified by a topological ordering algorithm in $O(|E|) = O(m^2)$ time, by Lemma 1(a). □

By Lemma 2 and Lemma 3, we have shown that SCP is polynomial-time solvable. This answers the open question raised in the end of Section 2 about the complexity of SCP. In particular, by constructing a segment graph $SG = (V, E)$ and solving a $v_1 - v_{m+1}$ shortest path problem on SG, we can solve the original SCP in polynomial time, which should be more efficient than solving a UFLP-based formulation by integer programming. This finding is important in that similar techniques can also be extended to calculate an exact optimal solution within short time for related spatial clustering problems investigated by Tsai et al. (2006) and Yang et al. (2009), where their proposed FCM can only give a good solution without guarantees of the running time and solution quality.

3.2. A polynomial-time algorithm for SCP and illustrative examples

Now, we give an algorithm, named TOSC, based on the idea of topological ordering, to calculate a segment clustering by a $v_1 - v_{m+1}$ shortest path in SG as follows:

Algorithm TOSC($[c_1, \ldots, c_m], f, L, m_l$)

Step 1. Calculate the cost for each possible cluster combination:
Calculate the final total cluster cost $C(s,t)$ for each $[s,t]$cluster by
$C(s,t) = (t-s+1)\cdot \widehat{Z}(s,t) + f, 1 \leqslant s \leqslant t \leqslant m+1$, where $\widehat{Z}(s,t) = \max_{i\in[s,t]}\{c_i\}$.

Step 2. Construct the set of qualified cluster arcs E in the segment graph SG:
Construct $m+1$ nodes: $v_1, v_2, \ldots, v_m, v_{m+1}$ and qualified cluster arc $(v_s, v_t) \in E$ with cost $\hat{c}_{st} = C(s,t-1) \geqslant L, 1 \leqslant s \leqslant s+m_l-1 < t \leqslant m+1$.

Step 3. Initialize for the distance labels and predecessors:
Let $D(v_i) = \infty$and $P(v_i) = NULL$ denote the initial distance label and predecessor for each node v_i, respectively.
Set $D(v_1) = 0$ and $P(v_1) = NULL$

Step 4. Calculate a $v_1 - v_{m+1}$ shortest path with topological ordering operations:
For $i = 1$ **to** $i = m$ **do**
For each qualified cluster arc (v_i, v_j) **do**
If $D(v_j) > D(v_i) + \hat{c}_{ij}$ **then**
$D(v_j) = D(v_i) + \hat{c}_{ij}; P(v_j) = v_i$;

Step 5. Trace the best clustering:
If $D(v_{m+1}) = \infty$ **then**
there exists no feasible clustering, **STOP**.
Else
output the best clustering by tracing from v_{m+1} back to v_1 based on predecessors. The best clustering has the total cost equal to $D(v_{m+1})$.

In algorithm TOSC, the bottlenecks are Steps 1, 2,and 4, which take $O(m^2)$ time to calculate individual cluster cost, construct the segment graph, and calculate the shortest path, respectively. On the other hand, Steps 3 and 5 take $O(m)$ time.

Note that Step 4 conducts topological ordering operations that scan each qualified cluster arc once. Since there are totally $O(m^2)$ qualified cluster arcs by Lemma 1(a), it takes $O(m^2)$ time. Step 4 calculates the shortest path from v_1 to v_{m+1} since SG is acyclic. As a result, although the Fibonacci heap implementation of Dijkstra's algorithm proposed by Fredman and Tarjan (1987) also takes $O(|E| + |V| \log |V|) = O(m^2)$ time to calculate a shortest $v_1 - v_{m+1}$ path, it requires more sophisticated data structures and is more difficult to implement than the topological ordering operations of TOSC.

We give two examples, as illustrated in Fig. 2, to explain the structure of SG and how TOSC works.

In Fig. 2a, there are five segments to be clustered with individual treatment costs equal to 6, 8, 4, 3, and 4, respectively. Each cluster induces a fixed cost 3. In this example, all the 15 generated cluster arcs are qualified since $L = 0$ and $m_l = 1$. Take a $v_1 - v_6$ path in SG that passes cluster arcs (v_1, v_2), (v_2, v_4) and (v_4, v_6); for example, its length equals $C(1,1) + C(2,3) + C(4,5) = 39$, since $C(1,1) = 6 + 3 = 9$, $C(2,3) = 2 \cdot \max\{8,4\} + 3 = 19$, and $C(4,5) = 2 \cdot \max\{3,4\} + 3 = 11$. On the other hand, the $v_1 - v_6$ shortest path calculated by TOSC passes cluster arcs (v_1, v_3) and (v_3, v_6) with length equal to $C(1,2) + C(3,5) = (2 \cdot \max\{6,8\} + 3) + (3 \cdot \max\{4,3,4\} + 3) = 19 + 15 = 34$. In the second example, illustrated in Fig. 2b, there are eight segments to be clustered with individual treatment costs equal to 6, 8, 4, 3, 4, 3, 6, and 7, respectively. Each cluster induces a fixed cost 3 and is qualified if it includes at least two segments with a total cost larger than or equal to 20. In this example, only 6 out of all the $C_2^9 = 36$ generated cluster arcs are qualified and connected to both v_1 and v_6. The $v_1 - v_9$ shortest path calculated by TOSC passes cluster arcs (v_1, v_4) and (v_4, v_9) with length equal to $C(1,3) + C(4,8) = 27 + 38 = 65$.

4. Computational experiments and discussion

In this section, we conduct computational experiments for solving SCPs. In particular, we test our proposed algorithm TOSC and compare its performance with CPLEX, a MIP solver, over two real datasets and eight categories of simulated datasets.

4.1. Settings for our computational experiments

We implement the TOSC algorithm in C language, compiled by Visual C++ 2005. To validate the correctness and efficiency of TOSC, we also solve the MIP formulation of SCP by CPLEX 11.2. All the experiments are conducted on a personal computer with Windows XP OS, 1 GB RAM, and Intel Core Duo 2 CPU of 1.86 GHz.

We use a subset of continuous roadway pavement segments in Georgia (see Table A1 in Appendix) as the source to generate two real test datasets ($m = 34$ and $m = 96$) for our experiments. In particular, the $m = 34$ case includes the last 34 segments as listed in Table A1. The $m = 96$ case includes the entire 96 segments in Table A1, which was unable to be solved by the Fuzzy c-means algorithm proposed by Tsai et al. (2006) and Yang et al. (2009). The original data only records the rating for each segment, which can be further used to derive its corresponding treatment cost based on the relationship between the segment rating and treatment cost, as shown in Fig. 3.

Simulated datasets are artificially generated to test the robustness and capability of TOSC in solving small-, medium-, and large-scale SCPs. Eight categories of simulated datasets ($m = 30$, 50, 100, 250, 500, 1000, 2500 and 5000), where each with ten random roadway pavement segments (thus totally 800 random simulated datasets), are generated and tested, respectively. The roadway pavement segment rating sequences for each random simulated dataset are random integers generated uniformly from the range [40, 100]. We then convert these rating into treatment costs based on Fig. 3.

We set the minimum number of segments in each project cluster, m_l, to be 2; the cost lower bound for a cluster, L, to be 0 USD; and the initial set-up cost, f, is set to be 150 thousand USD for all the simulated datasets.

4.2. Results of our computational experiments

Table 1 lists the computational results of CPLEX and TOSC for solving two real datasets. Table 2 summarizes the computational results of CPLEX and TOSC for solving small-scale simulated datasets, whereas Tables 3 and 4 only record the computational results of TOSC for solving simulated datasets of medium and large sizes, respectively.

The running times listed in Tables 1 and 2 show that TOSC is very efficient and takes much less time than CPLEX. Take the real dataset that includes 96 segments, for example; CPLEX could not solve it within 1 h, even for the tightest upper bounds for the number of clusters (i.e. $n = n^*$), whereas TOSC calculates an optimal solution for the same problem in less than 1 s. Note that the same case was too hard to be solved by the Fuzzy c-means algorithms proposed by Tsai et al. (2006) and Yang et al. (2009) in their segment clustering problems.

The efficiency of CPLEX highly depends on n, the initial upper bound for the number of clusters in SCP. In particular, the closer n is to n^*, the less time CPLEX takes. Take the $m = 34$ real dataset as an example; it takes 718, 9328, and 64,187 ms to solve the same SCP when n is set to 3, 5, and 10, respectively. In this example, n^*=3, so the solution calculated by setting $n = 3$ takes the least

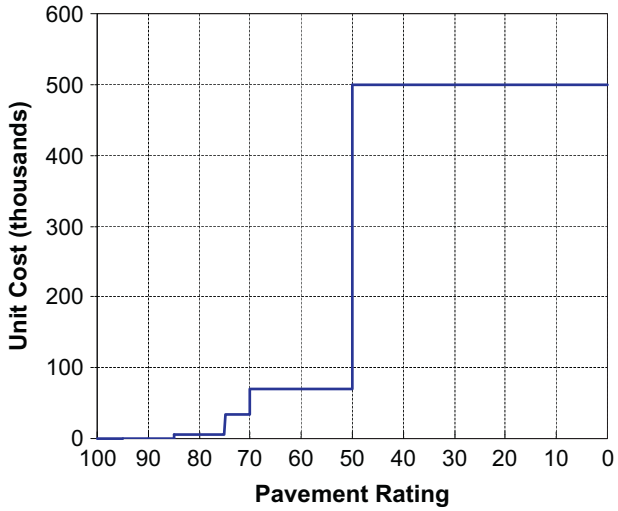

Fig. 3. Relationship between pavement segment rating and treatment unit cost (source: Yang et al., 2009).

Table 1
Computational results of CPLEX and TOSC for two real datasets.

m	CPLEX			TOSC		
	Obj.	n^*	Time (ms)	Obj.	n^*	Time (ms)
34	2254	3	718	2254	3	0
96	4851	10	–	4851	10	15

–, cases that take time longer than 1 h to get an optimal solution.
Here n is initially set to be 3 and 10 for these two real datasets, respectively.

time, whereas the time spent by setting $n = 5$ or $n = 10$ grows exponentially. In general, there is no clear way beforehand to set n as close to n^* as possible. However, even if we set $n = n^*$ for CPLEX to solve small-scale datasets, it still takes much more time than our proposed TOSC algorithm. Therefore, in the rest of the computational experiments, we only test the performance of TOSC for solving medium-scale and large-scale datasets.

Tables 3 and 4 show that TOSC still runs very fast with very stable performance for all the simulated datasets of medium to large sizes. Furthermore, if we take log to the average computational time (denoted by T), we observe that $\log(T)$ is proportional to $\log(m)$, as shown in Fig. 4. This observation validates the theoretical complexity of TOSC, $T = O(m^2)$, as derived in Section 3.

In summary, our proposed TOSC algorithm is very efficient and robust. It outperforms CPLEX even for small-scale datasets. It can deal with medium-scale and large-scale datasets within minutes. On the other hand, CPLEX usually takes more than 1 h, even for small-scale problems. The performance of CPLEX can be improved via good initial upper bounds for n. However, even by exploiting our proposed binary search techniques, CPLEX still performs very poorly in solving datasets of medium to large sizes. Therefore, we conclude that our proposed TOSC algorithm is very promising in grouping pavement segments that result in minimal total M&R cost.

5. Conclusions

In order to allocate and arrange the resources for pavement treatments along a highway more cost effective, a new segment clustering problem (SCP) is first formulated to address this need in which the best grouping of pavement M&R projects (i.e. segments) has to be identified to minimize the total M&R cost. In addition, we have proposed a new methodology and algorithm to address this problem. It is hoped the new problem formulation and proposed algorithm/methodology can effectively address this urgent need and stimulate other algorithm development (with the new problem formulation).

By viewing each segment as a customer and each cluster as a warehouse, we first propose a specialized uncapacitated facility location problem (UFLP) integer programming model to solve SCP. The constraints of UFLP provide more insights into SCP. In particular, the string property requiring the clustered segments to be spatially contiguous inspires us to develop a network flow model called segment graph (SG) for solving SCP. In SG, each node corresponds to a segment, and each arc represents a qualified cluster. This paper presents construction of the nodes and arcs of SG in $O(m^2)$ time by using $O(m^2)$ storage space. We also show SG is acyclic and propose an $O(m^2)$ time algorithm, TOSC, to determine a shortest path, which corresponds to a best clustering, based on idea of topological ordering.

The results of our computational experiments indicate TOSC is very efficient. In particular, TOSC dramatically outperforms CPLEX for all the real and artificially generated datasets that have been tested. It has demonstrated the proposed TOSC algorithm is very promising to group pavement segments that result in minimum total M&R cost.

For future research, we suggest investigation of more realistic and complicated segment clustering problems by considering different objective functions that take some nonlinear terms, such

Table 2
Computational results of CPLEX and TOSC for the small-scale simulated datasets.

Random cases	$m = 30$						$m = 50$					
	CPLEX			TOSC			CPLEX			TOSC		
	Obj.	n^a	Time (ms)	Obj.	n^*	Time (ms)	Obj.	n^a	Time (ms)	Obj.	n^*	Time (ms)
1	5538	7	12,781	5538	7	0	–	–	–	9408	12	0
2	4528	6	3429	4528	6	0	7105	8	134,030	7105	8	0
3	5334	7	6828	5334	7	0	4810	3	1499	4810	3	0
4	2250	1	15	2250	1	0	4810	3	828	4810	3	0
5	3260	2	46	3260	2	0	–	–	–	11,278	13	0
6	4570	5	1921	4570	5	0	7130	7	142,499	7130	7	0
7	2250	1	15	2250	1	0	–	–	–	7063	9	0
8	5583	8	15,421	5583	8	0	3566	3	2343	3566	3	0
9	4255	5	1609	4255	5	0	4790	5	33,374	4790	5	0
10	3260	2	46	3260	2	0	3544	2	296	3544	2	0
Avg.			4211			0			44,981			0

–, cases that take time longer than 1 h to get an optimal solution.
[a] Time spent by CPLEX is based on setting $n = n^*$.

Table 3
Computational results of TOSC for medium-scale simulated datasets.

Random cases	$m = 100$			$m = 250$			$m = 500$		
	Obj.	n^*	Time (ms)	Obj.	n^*	Time (ms)	Obj.	n^*	Time (ms)
1	9470	5	0	23,425	12	16	40,810	11	109
2	10,630	7	0	31,525	23	15	37,428	6	109
3	12,888	14	0	21,090	10	15	39,490	7	109
4	13,210	12	0	21,070	8	15	38,610	9	109
5	10,568	10	0	23,390	12	15	50,025	32	109
6	8285	4	0	18,768	4	16	47,413	33	110
7	14,328	15	0	23,408	12	15	42,984	19	109
8	8310	3	0	23,408	12	15	38,538	10	109
9	9470	5	0	31,144	26	15	39,790	9	125
10	8310	3	0	26,651	21	15	45,570	21	110
Avg.			0			15			111

Table 4
Computational results of TOSC for large-scale simulated datasets.

Random cases	$m = 1000$			$m = 2500$			$m = 5000$		
	Obj.	n^*	Time (ms)	Obj.	n^*	Time (ms)	Obj.	n^*	Time (ms)
1	82,595	27	797	194,510	42	11,750	376,247	67	92,235
2	87,429	34	781	195,774	58	11,750	359,688	41	92,016
3	93,281	47	812	196,594	56	11,719	385,519	81	92,094
4	86,039	39	796	186,528	33	11,781	381,811	80	92,156
5	84,893	35	796	181,681	30	11,734	379,485	64	91,875
6	85,942	29	796	187,618	32	11,734	402,768	118	93,078
7	96,364	52	796	199,929	63	11,750	397,924	100	92,079
8	95,357	52	796	196,911	51	11,734	391,442	91	92,923
9	81,644	22	796	189,805	39	11,734	371,798	47	92,782
10	86,136	40	796	201,370	63	11,796	401,619	112	92,641
Avg.			796			11,748			92,388

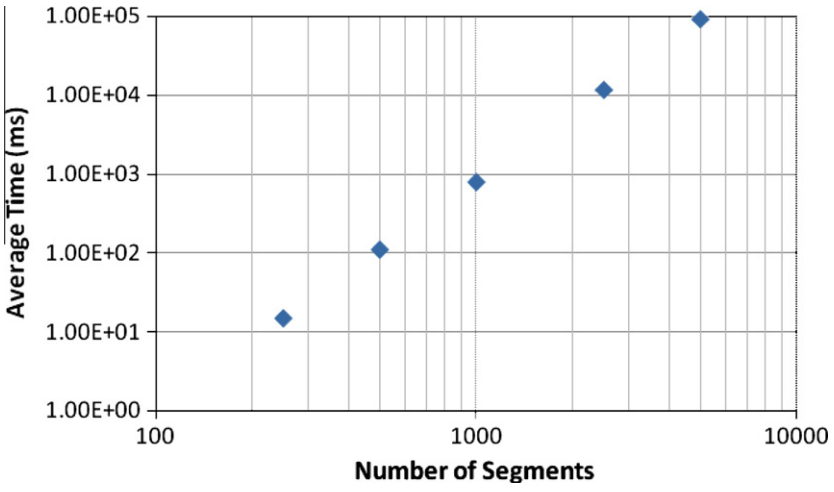

Fig. 4. Relationship between the average computational time and the number of segments for the tested medium-scale and large-scale simulated datasets.

as the sum of the squares of the difference in Euclidian distances of segments, by adding additional constraints, by relaxing the string property, or by extending the one-dimensional SCP to a two-dimensional SCP on a planar graph. In these cases, our proposed UFLP model can be flexibly modified more easily and serve as a good start for developing more efficient solution methods.

Acknowledgements

The authors would like to thank the Georgia Department of Transportation for providing the data to support our study. I-Lin Wang was partially supported by the National Science Council of Taiwan under Grant NSC98-2410-H-006-015-MY2.

Appendix A

See Table A1.

References

Abaza, K. A. (2007). Expected performance of pavement repair works in a global network optimization model. *Journal of Infrastructure Systems, 13*, 124–134.

Table A1
A real subset of continuous roadway pavement segments in the state of Georgia.

County	Segment from	Segment to	Rating	Cost[a]	County	Segment from	Segment to	Rating	Cost[a]
Oglethorpe	0	1	90	0	McDuffie	2	3	89	0
Oglethorpe	1	2	88	0	McDuffie	3	4	82	6
Oglethorpe	2	3	77	6	McDuffie	4	5	82	6
Oglethorpe	3	4	83	6	McDuffie	5	6	82	6
Oglethorpe	4	5	83	6	McDuffie	6	7	82	6
Oglethorpe	5	6	75	35	McDuffie	7	8	82	6
Oglethorpe	6	7	100	0	McDuffie	8	9	100	0
Oglethorpe	7	8	100	0	McDuffie	9	10	98	0
Oglethorpe	8	9	63	70	McDuffie	10	11.3	100	0
Oglethorpe	9	10	88	0	McDuffie	11.3	12	82	6
Oglethorpe	10	11	91	0	McDuffie	12	13	91	0
Oglethorpe	11	12	82	6	McDuffie	13	14	75	35
Oglethorpe	12	13	91	0	McDuffie	14	15	89	0
Oglethorpe	13	14	100	0	McDuffie	15	15.8	82	6
Oglethorpe	14	15	91	0	McDuffie	15.8	17	76	6
Oglethorpe	15	16	81	6	McDuffie	17	18	61	70
Oglethorpe	16	17	88	0	McDuffie	18	19	64	70
Oglethorpe	17	18	90	0	McDuffie	19	20	81	6
Oglethorpe	18	18.69	88	0	McDuffie	20	21	80	6
Wikes	0	1	72	35	McDuffie	21	22	75	35
Wikes	1	2	85	6	McDuffie	22	23	78	6
Wikes	2	3	74	35	McDuffie	23	24	75	35
Wikes	3	4	81	6	McDuffie	24	25.41	80	6
Wikes	4	5	90	0	Columbia	0	1	62	70
Wikes	5	6	87	0	Columbia	1	2	57	70
Wikes	6	7	87	0	Columbia	2	3	62	70
Wikes	7	8	84	6	Columbia	3	4	86	0
Wikes	8	9	87	0	Columbia	4	5	68	70
Wikes	9	10	84	6	Columbia	5	6	71	35
Wikes	10	11	77	6	Columbia	6	6.93	62	70
Wikes	11	12	63	70	Richmond	0	1	74	35
Wikes	12	13	63	70	Richmond	1	2	80	6
Wikes	13	14	62	70	Richmond	2	2.9	73	35
Wikes	14	15	70	70	Richmond	2.9	4	95	0
Wikes	15	16	90	0	Richmond	4	5	95	0
Wikes	16	17.1	90	0	Richmond	5	6	98	0
Wikes	17.1	18	82	6	Richmond	6	6.5	98	0
Wikes	18	19	80	6	Richmond	6.5	7.3	98	0
Wikes	19	20	83	6	Richmond	7.3	8	79	6
Wikes	20	21	60	70	Richmond	8	9	83	6
Wikes	21	22	81	6	Richmond	9	10.12	80	6
Wikes	22	23	87	0	Richmond	10.12	11	79	6
Wikes	23	24	87	0	Richmond	11	12	65	70
Wikes	24	25	93	0	Richmond	12	13	58	70
Wikes	25	26	90	0	Richmond	13	14	58	70
Wikes	26	26.89	81	6	Richmond	14	15	76	6
McDuffie	0	1	82	6	Richmond	15	16	73	35
McDuffie	1	2	82	6	Richmond	16	17.2	54	70

[a] Cost is in unit 1000 USD.

Al-Subhi, K., Johnston, D. W., & Farid, F. (1990). Resource-constrained capital budgeting model for bridge maintenance, rehabilitation, and replacement. *Transportation Research Record, 1268*, 110–117.

Álvarez, P., López-Rodríguez, F., Canito, J. L., Moral, F. J., & Camacho, A. (2007). Development of a measure model for optimal planning of maintenance and improvement of roads. *Computers and Industrial Engineering, 52*(3), 327–335.

Barahona, F., & Chudak, F., 2000. Solving large scale uncapacitated facility location problems. In P. Pardalos (Ed.), *Approximation and complexity in numerical optimization* (pp. 48–62).

Beck, M. P., & Mulvey, J. M., 1982. Constructing optimal index funds. Rep. EES-82-1. School of Engineering and Applied Science, Princeton University, Princeton, New Jersey.

Bellman, R. (1973). A note on cluster analysis and dynamic programming. *Mathematical Biosciences, 18*(3–4), 311–312.

Bilde, O., & Krarup, J. (1977). Sharp lower bounds and efficient algorithms for the simple plant location problem. *Annals of Discrete Mathematics, 1*, 79–97.

Cornuejols, G., Fisher, M. L., & Nemhauser, G. L. (1977). Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science, 23*, 789–810.

Cornuejols, G., Nemhauser, G. L., & Wolsey, L. A. (1990). The uncapacitated facility location problem. In R. L. Francis & P. B. Mirchandani (Eds.), *Discrete location theory* (pp. 119–171). New York: Wiley.

Dahl, G., & Minken, H. (2008). Methods based on discrete optimization for finding road network rehabilitation strategies. *Computers & Operations Research, 35*(7), 2193–2208.

Erlenkotter, D. (1978). A dual-based procedure for uncapacitated facility location. *Operations Research, 26*, 992–1009.

Fisher, W. D. (1958). On grouping for maximum homogeneity. *Journal of the American Statistical Association, 53*, 789–798.

Fredman, M. L., & Tarjan, R. E. (1987). Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM, 34*, 596–615.

Fwa, T. F., & Chan, W. T. (2000). Multiobjective optimization for pavement maintenance programming. *Journal of Transportation Engineering, 126*, 367–374.

Golabi, K., Kulkarni, R., & Way, G. (1982). A statewide pavement management system. *Interfaces, 12*(6), 5–21.

Grivas, D. A., Ravirala, V., & Schultz, B. C. (1993). State increment optimization methodology for network-level pavement management. *Transportation Research Record, 1397*, 25–33.

Guha, S., & Khuller, S. (1999). Greedy strikes back: Improved facility location algorithms. *Journal of Algorithm, 31*, 228–248.

Hansen, P., & Jaumard, B. (1997). Cluster analysis and mathematical programming. *Mathematical Programming, 79*, 191–215.

Hansen, P., Jaumard, B., & Simeone, B. (2002). Polynomial algorithms for nested univariate clustering. *Discrete Mathematics, 245*(1), 93–105.

Hansen, P., & Kaufman, L. (1972). An algorithm for central facilities location under an investment constraint. In P. Van Moeseke (Ed.), *Mathematical programs for activity analysis*. Amsterdam: North-Holland Publishers.

Jacobs, T. L. (1992). Optimal long-term scheduling of bridge deck replacement and rehabilitation. *Journal of Transportation Engineering, 118*, 312–322.

Jensen, R. E. (1969). A dynamic programming algorithm for cluster analysis. *Operations Research, 17*(6), 1034–1057.
Krarup, J., & Pruzan, P. M. (1983). The simple plant location problem: Survey and synthesis. *European Journal of Operational Research, 12*, 36–81.
Mirzain, A. (1985). Lagrangian relaxation for the star-star concentrator location problem: Approximation algorithm and bounds. *Networks, 15*, 1–20.
Mulvey, J. M., & Crowder, H. L. (1979). Cluster analysis: An application of lagrangian relaxation. *Management Science, 25*, 329–340.
Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (1987). An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming, 14*, 265–294.
Novick, B. (2009). Norm statistics and the complexity of clustering problems. *Discrete Applied Mathematics*. doi:10.1016/j.dam.2009.01.003.
Ouyang, Y., & Madanat, S. (2004). Optimal scheduling of rehabilitation activities for multiple pavement facilities: Exact and approximate solutions. *Transportation Research Part A: Policy and Practice, 8*, 347–365.
Ouyang, Y., & Madanat, S. (2006). An analytical solution for the finite-horizon pavement resurfacing planning problem. *Transportation Research Part B: Methodological, 40*, 767–778.
Rao, M. R. (1971). Cluster analysis and mathematical programming. *American Statistical Association, 66*, 622–626.
Shmoys, D., Tardos, E., & Aardal, K. (1997). Approximation algorithms for facility location problems. In *Proceedings of the 9th ACM symposium on theory of computing* (pp. 265–274).
Sun, M. (2006). Solving the uncapacitated facility location problem using tabu search. *Computers and Operations Research, 33*, 2563–2589.
Tsai, Y., Yang, C., & Wang, Z. (2006). Spatial clustering for determining economical highway pavement preservation projects. In *Proceedings of GeoCongress* (pp. 1–6).
Vinod, H. D. (1969). Integer programming and the theory of grouping. *Journal of the American Statistical Association, 64*, 506–519.
Yang, C., Tsai, Y., & Wang, Z. (2009). Algorithm for spatial clustering of pavement segments. *Computer-Aided Civil and Infrastructure Engineering, 24*, 1–16.