Zuse Institute Berlin

Takustr. 7
14195 Berlin
Germany

STEPHAN SCHWARTZ, RALF BORNDÖRFER, GERALD BARTZ

The Graph Segmentation Problem

ZIB Report 16-53 (November 2016)

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30-84185-0
Telefax: +49 30-84185-125

E-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

The Graph Segmentation Problem

Stephan Schwartz, Ralf Borndörfer, Gerald Bartz

Abstract

We investigate a graph theoretical problem arising in the automatic billing of a network toll. Given a network and a family of user paths, we study the graph segmentation problem (GSP) to cover parts of the user paths by a set of disjoint segments. The GSP is shown to be $\mathcal{NP}$-hard but for special cases it can be solved in polynomial time. We also show that the marginal utility of a segment is bounded. Computational results for real-world instances show that in practice the problem is more amenable than the theoretic bounds suggest.

1 Introduction

In this paper, we introduce the graph segmentation problem (GSP) to cover as many parts as possible of a given set of trajectories (paths in a network) by a set of disjoint segments (which are also paths). The problem has applications in automatic toll billing where the network users have to pay a certain toll for their journey. In such a billing, a user cannot be charged for a specific segment, unless he covers the entire segment during his trip.

An obvious solution to maximize the income is to choose the "atomic" segmentation, i.e., each arc of the network represents a segment. On the other hand, every segment requires specific maintenance, e.g., during construction periods or for manual review of contentious cases. Especially, if we are confronted with a very detailed network but the vast majority of the traffic passes by a great number of nodes, the savings of having fewer segments can be significant. The graph segmentation problem thus maximizes the total income of toll revenues subject to a limited number of segments. Figure 1 depicts optimal solutions of the GSP for the example of the German motorway system.

To the best of our knowledge, no models have been proposed that address this type of graph segmentation problem. The literature on road tolling is rather focused on problems of optimal toll pricing. A classical objective would be to determine arc tolls which will cause minimal congestion in the resulting traffic flow, see e.g. [4, 5] for a survey.

The GSP can be formulated as a set packing problem (SPP) of exponential size, see [1] for a survey. In the SPP we have a set $\mathcal{U}$, called the universe, and a family $\mathcal{S}$ of subsets of $\mathcal{U}$ together with a weight $w(S)$ for every $S \in \mathcal{S}$. The aim is to find a packing $\mathcal{C} \subseteq \mathcal{S}$ with pairwise disjoint sets that maximizes the total weight $\sum_{S \in \mathcal{C}} w(S)$. By choosing $\mathcal{S}$ as the set of possible segments and properly defining weights $w(S)$, one can formulate the GSP as a weighted set packing problem. This formulation allows

us to apply approximation results for the weighted set packing problem (see e.g. [2]) and it allows column generation approaches for the respective integer program. We show in this paper that this approach benefits from a graph theoretical study of the original problem. Structural insight also paves the way for combinatorial algorithms for special cases.

The article is structured as follows. In Section 2 we give a formal definition of the graph segmentation problem. The problem is proven to be $\mathcal{NP}$-hard in Section 3 where we give a reduction of the set packing problem to the GSP. In Section 4 we show that the problem can be solved in polynomial time if the maximum number of segments is constant. We prove a *subpath property* of optimal solutions and obtain tight upper bounds on the marginal utility and on the total profit. In Section 5 a set packing IP formulation is employed to solve the GSP. Using the subpath property from the previous section, we can drastically reduce the number of variables. Finally, we report on computational results for real-world instances in Section 6 where we apply the graph segmentation problem to the network of German motorways.

2 The Graph Segmentation Problem

The GSP can be formally defined as follows. Let $G = (V, A)$ be a directed graph with arc weights $c_a \geq 0$ and let $\mathcal{P}$ denote the set of simple paths in G. We extend the weight function c to paths $P \in \mathcal{P}$ by setting $c(P) := \sum_{a \in P} c_a$. A *segment* $S \in \mathcal{P}$ is a simple path in G and a set $\mathcal{S} \subseteq \mathcal{P}$ is called a *segmentation* if any two elements $S_i \neq S_j \in \mathcal{S}$ are arc-disjoint. Now let $\mathcal{T} = \{T_1, T_2, \ldots, T_t\} \subseteq \mathcal{P}$ be a set of *user trajectories* in G with corresponding demands $d_1, d_2, \ldots, d_t \in \mathbb{N}$. Finally, we define a utility function u on the set of all segmentations of the following form:

$$u(\mathcal{S}) := \sum_{i=1}^{t} d_i \sum_{S \in \mathcal{S} \,:\, S \subseteq T_i} c(S).$$

The definition of the utility function takes the users' perspective. Every user of path $T \in \mathcal{T}$ has to pay a fee $c(S)$ if he completely covers segment S on his trip. For a given number $k \in \mathbb{N}$, the graph segmentation problem (GSP) asks for a segmentation $\mathcal{S}$ with $|\mathcal{S}| \leq k$ such that $u(\mathcal{S})$ is maximized. Note that segment $S \in \mathcal{S}$ only yields a non-zero utility for user paths $T \in \mathcal{T}$ if $S \subseteq T$:

$$S \in \mathcal{P},\ S \not\subseteq T\ \forall T \in \mathcal{T} \implies u(\{S\}) = 0. \tag{1}$$

Another important observation is that the utility function is additive.

Proposition 1. *Let $\mathcal{S} = \{S_1, \ldots, S_k\}$, then*

$$u(\mathcal{S}) = \sum_{i=1}^{k} u\big(\{S_i\}\big).$$

Proof.

$$u(\mathcal{S}) \;=\; \sum_{i=1}^{t} d_i \sum_{S\in\mathcal{S}\,:\,S\subseteq T_i} c(S) \;=\; \sum_{S\in\mathcal{S}} \sum_{T_i\in\mathcal{T}\,:\,S\subseteq T_i} d_i\; c(S) \;=\; \sum_{i=1}^{k} u\big(\{S_i\}\big).$$

□

3 Complexity of the GSP

Theorem 2. *The graph segmentation problem is $\mathcal{NP}$-hard.*

Proof. The set packing problem (SPP) with unit weights is one of the 21 classical $\mathcal{NP}$-hard problems presented by Karp [3]. In the following, we give a reduction of the SPP to the GSP by constructing a graph $G = (V, A)$ with arc weights c and user trajectories $\mathcal{T}$ with demand d. We start with an instance $(\mathcal{U}, \mathcal{S})$ of the SPP and introduce for every element $u \in \mathcal{U}$ the nodes $u^s, u^t \in V$ and an arc $u' = (u^s, u^t) \in A$ with $c_{u'} = 1$. Now consider an arbitrary set $S \in \mathcal{S}$, say $S = \{u_1, \dots u_r\}$. First, we introduce a pair of nodes $S^s, S^t \in V$ and then we add the arc (S^s, u_1^s) with costs $M - |S|$ for a sufficiently large M. We also add the arcs (u_i^t, u_{i+1}^s) for $i = 1, \dots, r-1$ and (u_r^t, S^t), all with costs 0. Finally, we define $T_S := (S^s, u_1^s, u_1^t, u_2^s, u_2^t, \dots, u_r^s, u_r^t, S^t)$ and the set of user trajectories $\mathcal{T} = \{T_S \,:\, S \in \mathcal{S}\}$ with $d \equiv 1$. The given construction can be done in polynomial time and the following two observations hold:

$$c(T_S) = M \qquad \forall\, T_S \in \mathcal{T}. \tag{2}$$

To see this, note that for every set S, the arc (S^s, u_1^s) has weight $M - |S|$ and for every element in S we have an arc of unit weight.

$$\forall S, T \in \mathcal{S}: \qquad S \cap T = \varnothing \quad \Longleftrightarrow \quad T_S \cap T_T = \varnothing. \tag{3}$$

Indeed, if $u \in S \cap T$, then by construction $(u^s, u^t) \in T_S$ and $(u^s, u^t) \in T_T$. On the other hand, $u^s \in T_S \cap T_T$ or $u^t \in T_S \cap T_T$ would imply $u \in S \cap T$ by construction of G. Moreover, with $S \neq T$ we have $S^s \neq T^s$ and $S^t \neq T^t$.

We show that SPP has a packing of size k iff GSP has a segmentation of utility $M\,k$. First, assume that $S_1, \dots, S_k \in \mathcal{S}$ are pairwise disjoint sets. Then, the segments $T_{S_1}, \dots, T_{S_k}$ are disjoint due to (3) and constitute a feasible segmentation with utility $u(\{T_{S_1}, \dots, T_{S_k}\}) = M\,k$ due to (2).

Assume on the other hand, that we find a segmentation $\{P_1, \dots, P_\ell\}$ of utility $M\,k$. Then $\ell = k$ for sufficiently large M ($\geq |\mathcal{U}| + 1$). Taking together (1) and (2), we have $u(P_i) \leq M$ and together with $\sum_{i=1}^{k} u(\{P_i\}) = M\,k$ we obtain $u(P_i) = M\ \forall i$. This implies that for every P_i we find a set $S_i \in \mathcal{S}$ such that $S_i^s \in P_i$ (since M is sufficiently large) and for every $u \in S_i$ we have $(u^s, u^t) \in P_i$ since otherwise $u(\{P_i\}) < M$. From the fact that the $P_1, \dots, P_k$ are disjoint and from (3), we can derive that the corresponding sets $S_1, \dots, S_k$ are pairwise disjoint.

□

4 Structural Properties of the GSP

After we have seen that the graph segmentation problem is $\mathcal{NP}$-hard, it is important to gain more structural insight. The following simple but important result is referred to as the *subpath property* and helps to dramatically decrease the number of possible segments.

Theorem 3 (subpath property)**.** *One can always find an optimal segmentation $\mathcal{S}$ such that*

$$\forall S \in \mathcal{S} \quad \exists T \in \mathcal{T} \; : \; S \subseteq T.$$

Proof. Let $\mathcal{S}^*$ be an optimal segmentation. If $\mathcal{S}^*$ does not have the declared property, there is a segment $S \in \mathcal{S}^*$ such that $S \not\subseteq T \; \forall T \in \mathcal{T}$. Due to (1) and Proposition 1 we can eliminate S from $\mathcal{S}^*$ without a loss of utility. By iterating this process we obtain an optimal segmentation with the desired property. □

Theorem 3 states that we only need to consider subpaths of user trajectories as possible segments. This has several important consequences. The first is a nice complexity result if the upper bound k on the number of segments is constant.

Proposition 4. *For a fixed number of segments, the graph segmentation problem can be solved in polynomial time.*

Proof. With Theorem 3 we only have to check the subpaths of the user trajectories $\mathcal{T} = \{T_1, \ldots, T_t\}$. This makes at most $t \cdot \binom{|V|}{2}$ possible segments, or respectively, $\binom{t\binom{|V|}{2}}{k} \in \mathcal{O}(t|V|^{2k})$ possible segmentations to check, where k is the number of segments. Thus, for fixed k, this brute-force approach can be executed in polynomial time. □

In particular, it is possible to efficiently find a single optimal segment ($k = 1$) of a GSP instance. This gives rise to a heuristic greedy approach where in each step an optimal segment is chosen and the set of feasible remaining segments is adjusted. We leave out the details of this greedy approach but it can be shown that the resulting solution can be arbitrarily bad. Namely, consider the instance in Figure 2 with a single path as network. The above greedy algorithm would choose the full path T_{n+1} as single most profitable segment but then, no additional disjoint segment can be chosen which results in a total profit of $n + \varepsilon n$ for the greedy approach. On the other hand, the optimal segmentation is obviously the atomic segmentation yielding a profit of $n \cdot n + (n + \varepsilon n)$.

We show next that the marginal utility of an additional segment is bounded.

Proposition 5. *Let $\mathcal{S}_k^*$ denote an optimal segmentation subject to $|\mathcal{S}_k^*| \leq k$. Then*

$$u(\mathcal{S}_{k+1}^*) \; \leq \; \frac{k+1}{k} \, u(\mathcal{S}_k^*).$$

Proof. Consider the set $\mathcal{S}_{k+1}^*$ and let

$$S_{min} \leftarrow \begin{cases} \underset{S \in \mathcal{S}_{k+1}^*}{\arg\min}_1 \, u(\{S\}) & , \; |\mathcal{S}_{k+1}^*| = k+1 \\ \varnothing & , \; \text{else.} \end{cases}$$

With $u(\mathcal{S}^*_{k+1}) \overset{\text{Prop.1}}{=} \sum_{S\in\mathcal{S}^*_{k+1}} u(\{S\})$, we have

$$u(\{S_{min}\}) \;\leq\; \frac{u(\mathcal{S}^*_{k+1})}{k+1} \tag{4}$$

and therefore,

$$\begin{aligned} u(\mathcal{S}^*_k) \;&\geq\; u(\mathcal{S}^*_{k+1} \setminus \{S_{min}\}) \;=\; u(\mathcal{S}^*_{k+1}) \;-\; u(\{S_{min}\}) \\ &\overset{(4)}{\geq}\; u(\mathcal{S}^*_{k+1}) \;-\; \frac{u(\mathcal{S}^*_{k+1})}{k+1} \;=\; \frac{k}{k+1}\, u(\mathcal{S}^*_{k+1}). \end{aligned}$$

□

With the previous result we can immediately derive an upper bound for the optimal utility in the graph segmentation problem. Using the insight of Proposition 4 we can compute the single most profitable segment $\mathcal{S}^*_1$ in $\mathcal{O}(t|V|^2)$.

Corollary 6.

$$u(\mathcal{S}^*_k) \;\leq\; k \cdot u(\mathcal{S}^*_1).$$

Note that both of the given bounds are tight. The instance of Figure 2 provides an example for tight bounds if we omit the user trajectory T_{n+1}. Then, only atomic segments yield positive (but identical) profit and thus, $u(\mathcal{S}^*_{i+1}) = \frac{i+1}{i}\; u(\mathcal{S}^*_i)\;\forall i$ and consequently, $u(\mathcal{S}^*_k) = k \cdot u(\mathcal{S}^*_1)$.

While Proposition 5 bounds the marginal utility, the utility function is not concave in general (which would lead to diminishing marginal costs). To see this, consider the network consisting of a path $(1,2,3,4)$ with arc weights $c_a \equiv 1$. Assume trajectories $T_1 = (1,2,3,4)$ and $T_2 = (2,3)$ with demands $d_1 = d_2 = n$. For $k = 1$ the segment $(1,2,3,4)$ is optimal yielding a profit of $3n$. For $k = 2$ this is still optimal while for $k = 3$, the segmentation $\{(1,2),(2,3),(3,4)\}$ is optimal with a total profit of $4n$.

5 Solving the GSP

In this section, we present a set packing type integer programming (IP) formulation to solve the graph segmentation problem. We will use a binary variable x_S for every possible segment $S \in \mathcal{P}$ to decide if S is part of an optimal segmentation. In addition, we compute the parameters $c_S = c(S)$ and we define the incidences

$$p^T_S := \begin{cases} 1 & S \subseteq T, \\ 0 & S \not\subseteq T, \end{cases}$$

which indicate whether segment S is included in trajectory T. We can now state the integer program.

$$\max_{x} \quad \sum_{S \in \mathcal{P}} c_S \, x_S \sum_{T_i \in \mathcal{T}} d_i \, p_S^{T_i} \tag{5a}$$

$$\text{s.t.} \quad \sum_{S \in \mathcal{P} \,:\, a \in S} x_S \;\leq\; 1 \qquad \forall a \in A \tag{5b}$$

$$\sum_{S \in \mathcal{P}} x_S \;\leq\; k \tag{5c}$$

$$x_S \;\in\; \{0,1\} \qquad \forall S \in \mathcal{P} \tag{5d}$$

The objective (5a) of the IP is to maximize the utility of the segmentation. In (5b), we ensure that the segments are disjoint, while (5c) guarantees that we choose at most k segments.

The number of parameters and variables in the stated program is potentially huge as we consider every possible path $P \in \mathcal{P}$ as a potential segment. With Proposition 3 however, we can reduce the number of variables and parameters to be of polynomial size. In particular, the number of constraints is in $\mathcal{O}(|A|)$ while the number of variables is in $\mathcal{O}(t|V|^2)$. It is also interesting to consider variants of the problem by restricting the set of possible segments. One possibility is to introduce a maximum number L of arcs per segment. For small L this can lead to a significant reduction in the number of variables.

6 Computational Study

We applied the model (5) to the network of German motorways. Here, a truck toll for trucks weighing 7.5 tonnes or more was introduced in 2005. The collection of the fee is done by the Toll Collect GmbH. It is based on the Global Positioning System (GPS) for automatic billing and a web application for manual booking. For the automatic billing, the trucks need to be equipped with an *On-Board-Unit* which identifies the used segments and computes the toll to be paid.

The network consists of over 4000 nodes and 8000 arcs. About 370,000 user trajectories of a special "metering fleet" were considered. We used a standard computer with 4 cores at 2GHz and 8GB RAM, the IP was solved with Gurobi 6.51. The computation time for all of the instances in Table 1 taken together was less than half a day.

In Table 1, the results of several instances of the graph segmentation problem are displayed. We consider 4×5 instances varying in the number of allowed segments and for different maximum segment lengths. We are interested in finding a reasonable compromise between the amount of coverage, computational complexity, and the complexity of the solution. That is, we would like to cover most of the traffic with a small number of short segments.

We used the packing IP formulation (5) with different parameters k for the number of allowed segments. In addition, we introduced an upper bound L on the number of arcs in a segment in order to reduce the number of possible segments in the IP

Table 1: Results for the GSP on German motorways. We report the fraction $\frac{u(\mathcal{S}^*_{k,L})}{u(A)}$ of the optimal objective value $u(\mathcal{S}^*_{k,L})$ of the respective instance and the objective value $u(A)$ of the atomic segmentation. The parameter L bounds the number of arcs in a segment from above, k is the allowed number of segments.

L \ k	10	100	500	1000	4000	$\lvert A\rvert$
2	3%	18%	52%	72%	98%	100%
3	4%	22%	59%	78%	99%	100%
15	6%	31%	67%	82%	99%	100%
150	6%	32%	67%	82%	99%	100%

formulation. The entries of the matrix describe the objective value of the respective instance in relation to the maximal possible income with an atomic segmentation.

As expected, the total income increases with the number of allowed segments. Also, the marginal utility seems to decrease. Thus, the optimal solution is much better than the theoretic bounds suggest. As we have hoped, we only need 1000 (instead of $|A| > 8000$) segments to obtain 82% of the maximal profit. When the number of segments is reduced from $|A|$ to 4000, the income only decreases by one percent. One can also observe that reasonable choices for the parameter L do not affect the quality of the optimal solution. Indeed, segments with at most 15 arcs seem to suffice for practical purposes. This allows to deal also with larger networks, e.g., the entire street network of Germany.

References

[1] R. Borndörfer, *Aspects of set packing, partitioning, and covering* Diss. Technische Universität Berlin, 1998

[2] M. Halldórsson, *Approximations of Weighted Independent Set and Hereditary Subset Problems*, Journal of Graph Algorithms and Applications **4** (2000), 1–16

[3] R. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, Springer (1972), 85–103

[4] M. Labbé and A. Violin, *Bilevel programming and price setting problems*, 4OR **11** (2013), 1–30

[5] M. Yildirim, *Congestion toll pricing models and methods for variable demand networks*, Diss. University of Florida, 2001

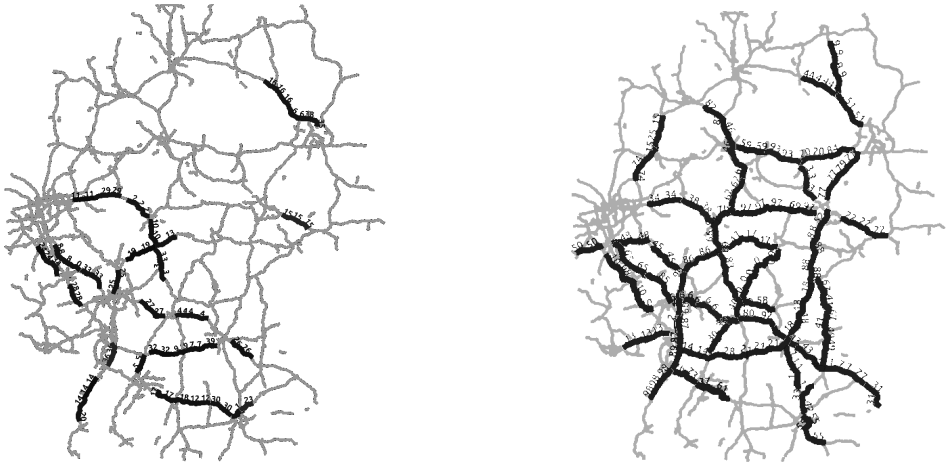

Figure 1: Optimal segmentation on German motorways with $k = 40$ segments (left) and $k = 100$ segments (right).

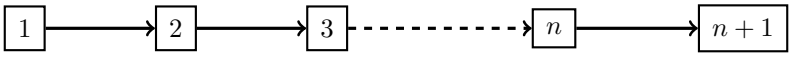

Figure 2: Exemplary instance of the GSP. Consider the arc weights $c_a \equiv 1$ and trajectories $T_1 = (1, 2),\ T_2 = (2, 3), \ldots,\ T_n = (n, n{+}1)$ and $T_{n+1} = (1, 2, \ldots, n{+}1)$ with demands $d_1 = d_2 = \cdots = d_n = n$ and $d_{n+1} = 1 + \varepsilon$ and let $k = n$.