FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Diplomarbeitspräsentation

ALGORITHMS AND COMPLEXITY GROUP

An Interactive Optimization Framework for Point Feature Label Placement

Masterstudium:
Software Engineering & Internet Computing

Raphael Löffler

Technische Universität Wien
Institute of Logic and Computation
Arbeitsbereich: Algorithms and Complexity
Betreuer: Dr. Martin Nöllenburg

Motivation

The task of label placement is important in information visualization and especially in the area of cartography. The quality of the resulting map is highly dependent on the location of the labels for the features. They shall not obscure and overlap other features or labels and have to satisfy different guidelines or preferences for better visual appearance, unambiguity and legibility.

Problem Statement

In cartography automatization is on the rise. But automatic label placement can hardly incorporate aesthetic criteria and cognitive aspects of human vision. Automatic label placement is cheap and fast but less qualitative than solely manual map creation which is expensive and time consuming. Manual postprocessing of automated created maps is tedious.

Our Solution

An interactive optimization framework that combines

Automatic Labeling Algorithms

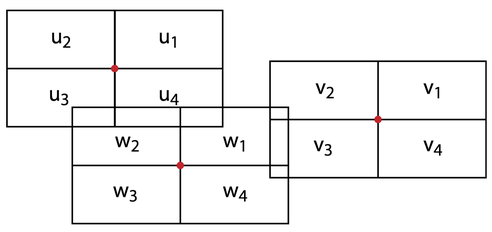
+

Human Domain Expertises

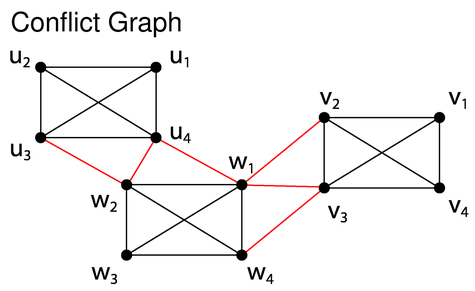
Model

We use:

- A Fixed Four Position Model (i.e., four candidates per feature with fixed positions).
- A Conflict Graph as basis for modifications and algoritms.
- A Quad Tree to store spatial data.
- Static Geographic Maps



Each label candidate is represented by a node in the conflict graph. Every label overlap is translated to an edge between the corresponding nodes.



Algorithms

6 implemented algorithms:

- Simple (greedy) Algorithm
- Independent Set Algorithm
- Three Rules Algorithm
- MaxHs (MAX-SAT) Alg.

} Label Number Maximization

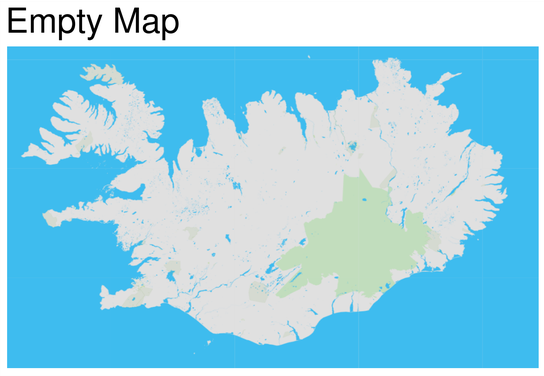
- Minimum Number Conflicts Alg.
- Integer Linear Programming Alg.

} Conflicts Minimization

Implementation

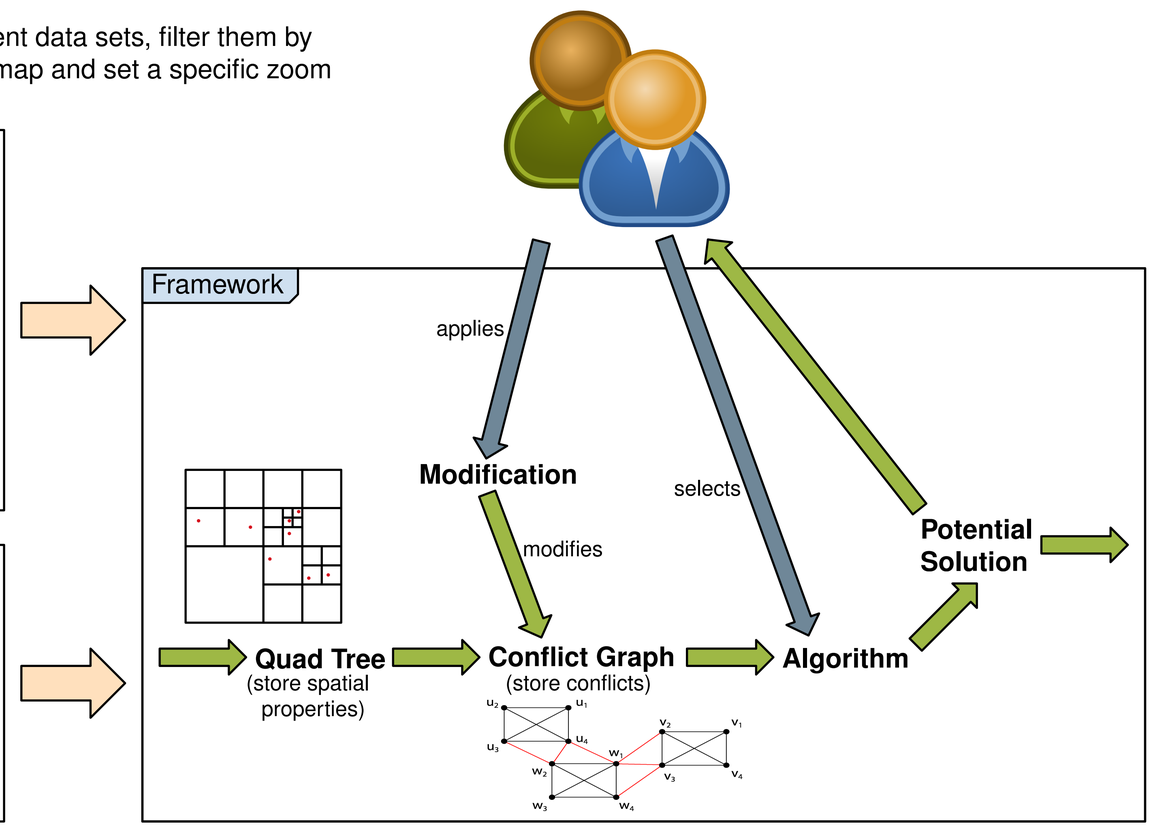
The framework enables the user to load different data sets, filter them by categories of the features, zoom and pan the map and set a specific zoom level for the final map.

Empty Map

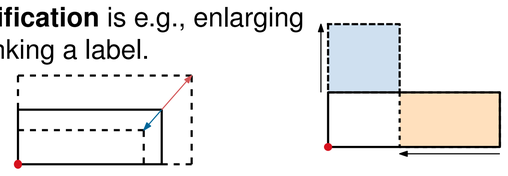


Data

```
[
{"label": "Reykjavík","mc": "place","sc": "city",
        "x": -2442598.6163094793, y": 9386931.633935148},
{"label": "Dalvík","mc": "place","sc": "town",
        "x": -2062799.011391919,"y": 9868939.164436119},
{"label": "Ólafsvík","mc": "place","sc": "town",
        "x": -2639210.8560456797,"y": 9581012.359032592},
{"label": "Akranes","mc": "place","sc": "town",
        "x": -2458307.410593027,"y": 9430743.98638051},
{"label": "Húsavík","mc": "place","sc": "town",
        "x": -1930395.7648896866,"y": 9888711.321015112}
]
```



A **Modification** is e.g., enlarging or shrinking a label.

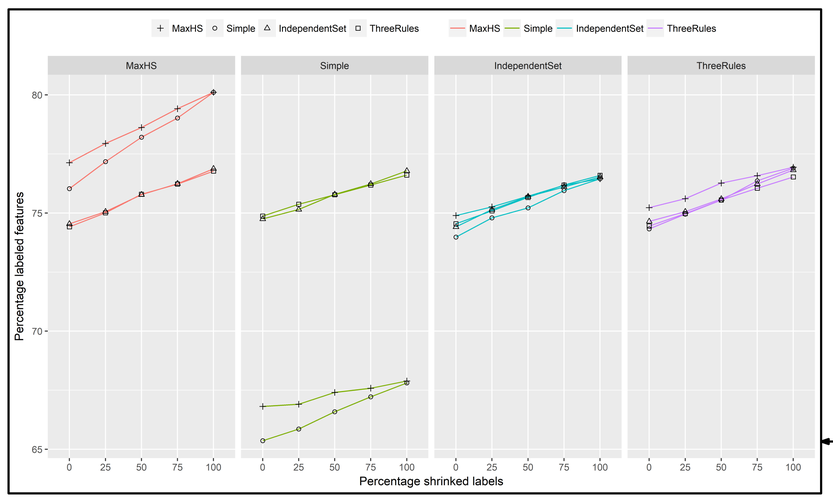


- **Nine** different types of modifications.
- Inclusive deletion and fixation of candidates.
- All modifications can be reduced to an **edge delete** and **edge add** in the conflict graph.

Labeled Map

Evaluation and Results

A typical user task is to calculate a labeling solution, modify some labels and then recalculate a new solution. This scenario was used for our evaluation to search for algorithm combinations that produce good labeling solutions.

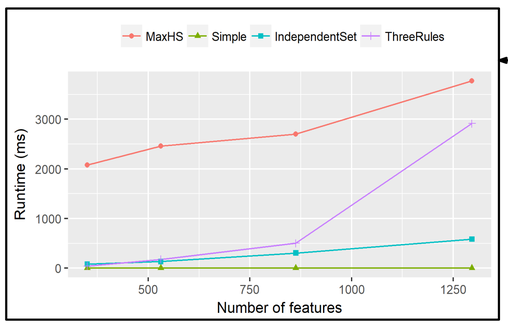


Percentage of labeled features (y-axis) for specific algorithm combinations. The line color indicates the algorithm used for calculating the initial solution and the different symbols indicate the algorithm used for recalculation. The x-axis shows the modification i.e., the percentage of labels that were shrinked.

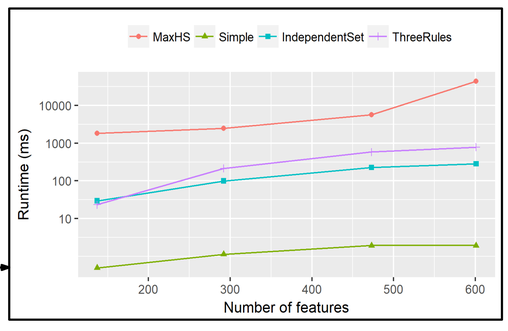
For our evaluation we use:

- eight different sparse and dense data sets,
- our four implemented label number maximization algorithms, and
- the two representative modifications “font size shrink” and “font size enlarge” (they represent an edge remove and an edge add in the conflict graph)

We modify 20% of the labels in the initial solution by shrinking a subset of them and enlarging the remaining. We run each algorithm and modification combination 100 times. The aggregated results can be seen in the figure on the left.



Performance of the Label Number Maximization Algorithms on **sparse** data sets with growing number of features to label (x-axis). The runtime is given in milliseconds (y-axis).



Performance of the Label Number Maximization Algorithms on **dense** data sets with growing number of features to label (x-axis). The runtimes are given in milliseconds (y-axis). A **log scale** is used for the y-axis.

Results:

- We built an interactive optimization framework and compared different label number maximization algorithms.
- Always applying the exact `MaxHS` algorithm produces best solutions, but is very slow, especially with large data sets. Combinations of the `ThreeRules` or `IndependentSet` algorithm with `Simple` produce fast and high quality solutions for most of the data sets → Best choice!

Kontakt: e1025967@student.tuwien.ac.at