ABSTRACT
Interactive Gantt charts are a well known technique to represent and edit interval and point events. Providing scalability to these editable charts in the presence of a large amount of events is a challenging problem that is under-explored. We propose a scalable and compact variation of editable Gantt charts based on focus+context, animation, and clustering. Drag-and-drop functionalities are available for relocating events in time. In this paper we discuss our design decisions when elaborating this interactive graphics, and how they may be applied to analyze sizeable sets of projects from a real company in the power generation sector.

CCS Concepts
• Human-centered computing → Visualization systems and tools;

Keywords
Visualization; Gantt chart; timeline; scalability

1. INTRODUCTION
Voluminous datasets are increasingly available not only for computing professionals but for enterprise managers and also for the general public. Information Visualization provides interactive graphical solutions that may help users to understand and explore these datasets. However, there is still a need for improving these solutions with respect to scalability. Indeed, Thomas and Cook [8] pointed out the need of ways to transform data into “new scalable representations that faithfully represent the underlying data’s relevant content”.

We are especially interested in providing scalable solutions for interactive Gantt charts. These charts are used worldwide to represent events along time (mainly interval events, but also point events). Despite this, few recent research reports focus on scalable Gantt charts.

In this paper we propose a scalable, compact, editable, and interactive Gantt chart. Based on a “focus + context” technique, these charts represent clusters of interval events. Users may expand these clusters, and may also inspect and drag interval events contained in such clusters. Our method extends that of Silva et al. [7].

This paper is organized as follows. Section 2 presents a bibliographic review on recent techniques to construct Gantt charts. Section 3 discusses our method. Section 4 present a case study that exemplifies the use of our approach in a real dataset stemming from the power generation industry. Section 5 concludes and suggests future research directions.

2. RELATED WORK
Few recent reports deal with scalability in Gantt charts. Scroll bars and zoom, as discussed in FlowOpt [1] and ChronoZoom [9], for instance, are simple solutions that augment the drawing space, thus providing scalability to some extent. However, they provide views of the datasets that may be too small for users, without a dataset overview to guide them on their exploration of the data space, i.e., they do not implement the overview+detail technique.

Clustering techniques may help providing this overview. SchemaLine [5] enables users to manually group point events in schemas (frames), represented as stripes. This process
may help dealing with scalability, but it is more suitable to scenarios with very few events, due to the need of user intervention. Besides, it deals with point events instead of interval events.

The automatic clustering approach of LiveGantt [3] groups interval events according to their start and end times, and displays these clusters as rectangles, hiding their clustered events. Users may use lenses with semantic zoom to investigate cluster contents. The scalability of this approach seems to be limited, since the height of interval bars may be very small when there are many concomitant events. This situation may hamper event relocation by drag-and-drop operations.

The Temporal Mosaics technique [4] applies a space-filling approach to Gantt charts. Their authors argue that these mosaics “facilitate the detection of concurrent and overlapping events, even as the number of events to be represented increases”. However, its scalability seems to suffer from the same problem as does LiveGantt.

LifeFlow [11] implements a clustering approach that hierarchically aggregates point event sequences according to prefixes of these sequences. It may present any number of records using the same amount of screen space. However, like SchemaLine, LifeFlow works only with point events and not with interval events.

LifeLines [6] and LifeLines2 [10] group points and interval events according to predefined event categories. Users may visually compact or expand these groups, based on focus+context and semantic zoom techniques. Scrolling is available for situations in which these techniques are not properly scalable.

We highlight two approaches that adapt Gantt charts in order to do compact layouts: Temporal Mosaics [4] and the project timeline of Silva et al. [7]. Both provide a compact view of interval events, in which interval bars may share vertical positions when they do not have time intersection. This is possible because the chart Y-axis is not mapped to any event variable.

Here, we are interested on editable Gantt charts, i.e., we want to provide drag-and-drop capabilities for interval event relocation in time. FlowOpt, LiveGantt and SchemaLine enable this operation. In LiveGantt, users must first reveal the events inside a cluster in order to move them. In SchemaLine, users may perform drag-and-drop operations to relocate events and to edit frames. FlowOpt’s Gantt Viewer and Silva et al.’s project timeline all implement drag-and-drop operations that enable users to move events in time. LifeLines, LifeLines2, LifeFlow, Temporal Mosaics and Chrono-zoom do not seem to provide drag-and-drop facilities for time relocation of events.

3. METHOD

In this section we discuss the design decisions we made in the definition of our interactive Gantt chart. As most Gantt charts, it has an horizontal axis that represents time, and it depicts interval events as rectangles that do not share pixels.

The vertical axis of our chart may be split into areas according to a given nominal variable. If there is no variable assigned to this axis, we consider that the axis has a single area. Each area receives a background color in order to help users single out which events belong to this area. Each area may have concomitant interval events, in contrast to other Gantt chart implementations, e.g. LiveGantt [3]. Rectangles may also have colors, as we explain later.

The central part of our solution is related to scalability. Our solution initially clusters events and present these clusters to users as rectangular bars. When a user clicks on one of these rectangles, it expands vertically and reveals its internal events. A click in the background triggers the inverse action. Note that the height of a closed (i.e., not expanded) cluster rectangle and the height of an event rectangle are the same. Therefore, the expansion of a cluster reduces the height of the rectangles outside it. This causes a focus+context [2] effect in which users may focus on the expanded cluster without losing context. Figure 1 illustrates this approach. It is important to use a clustering method that provides a limited number of clusters (e.g. 100), otherwise the height of each rectangle become too reduced for comfortable selection and dragging operations.

We also implemented drag-and-drop capabilities for interval events. Any event rectangle may be horizontally dragged. If it is part of a cluster, it leaves this cluster and is inserted into another one. Therefore, removed projects are always visible to the user, and so it is not necessary to discover which cluster the project was moved.

We defined a compact layout inside each area, following Silva et al. [7]. For each area, given a fixed height, we wanted to reduce the amount of rectangle rows, since fewer rows imply more vertical space available, which in turn facilitates locating and clicking on rectangles. This way, our layout algorithm enables rectangles of interval or cluster events to share the same graphical row, but without visually overlapping each other.

4. RESULTS

The scenario that inspired our work stems from a real company in the power generation sector. It comprises a large set of projects (in other words, a portfolio) which have risks and monthly costs associated to them. Decision makers must analyze these data and decide if and how they should reschedule those projects in order to fit them to the annual budget. The problem is solved by the execution of an iterative 2-step process, which includes the execution of an optimization procedure (which is not the subject of this paper) followed by an interactive, visual, and exploratory data analysis phase. Our solution should detect and highlight infeasibilities as well as provide sufficient information so that users could try to solve those issues.

Given this scenario, we provided a dashboard composed by two cost and risk charts, since the original problem stipulated two kinds of resources. We also proposed an interactive Gantt chart adhering to our current proposal. Moving projects on the Gantt chart automatically changes the values on the cost and risk charts.

We customized our solution to this scenario by choosing appropriate colors for rectangles. Event rectangles are green, yellow or red according to the cost of the projects they represent. A cluster rectangle receives the color of the most expensive project that it groups. Therefore, expensive projects may be perceived by users even when they are clustered. Those projects should be the focus of decision makers when they try to turn an infeasible portfolio into a feasible one. For this scenario, we also defined that our clustering function should group projects with similar start and end times, and with similar costs. Figure 2 presents this dashboard. There is a budget infeasibility in the first year (the
red portion in the “Cost - CAPEX Projects” bar chart). An expanded red cluster at the CAPEX (orange background) area of the Gantt chart presents 3 projects. The first two are expensive (red) and may be moved in order to reach a feasible portfolio.

It is worth noting that this portfolio has about 3,000 projects, where many groups of projects have a very similar start and end date. The scalability techniques proposed in our Gantt chart seem to deal correctly with this use case.

Our approach is limited by the number of rectangle rows it generates when clusters are not expanded, the maximum number of projects grouped by clusters, and the available vertical space on the screen.

5. CONCLUSIONS

This paper presented our proposal for an interactive Gantt chart with clustering capabilities. We showed how we mapped visual properties to the available data, and exemplified our approach with a real case containing almost 3,000 events, demonstrating how our mapping deals with large sets. We learned that the customization of clustering algorithms with respect to the available data made possible better grouping results than other traditional clustering approaches.

Future work includes doing usability tests with this tool, correcting possible problems, and integrating it into a software that optimizes event relocation. Besides, we aim to insert our proposal into a risk management tool of a large power generation company.

6. ACKNOWLEDGMENTS

We thank ANEEL and AES for their financial support to our research.

7. ADDITIONAL AUTHORS

Additional author: Rafael S. V. de Barros (AES-Tietê, email: rafael.schmitz@aes.com).

8. REFERENCES

Figure 2: Visualization screen of Portfolio Optimization and Visualization System with our interactive Gantt chart.


