14th SC@RUG 2017 proceedings 2016-2017

Smedinga, Reinder; Biehl, Michael; Kramer, Femke


university of groningen
faculty of science and engineering
computing science
SC@RUG 2017 proceedings
14th SC@RUG
2016-2017
Rein Smedinga, Michael Biehl and
Femke Kramer (editors)
www.rug.nl/research/jbi

SC@RUG 2017 proceedings

Rein Smedinga
Michael Biehl
Femke Kramer
editors

2017
Groningen

ISBN (e-pub pdf): 978-90-367-9847-1
ISBN (book):978-90-367-9848-8
Publisher: Bibliotheek der R.U.
Title: 13th SC@RUG proceedings 2015-2016
Computing Science, University of Groningen
NUR-code: 980

About SC@RUG 2017

Introduction

SC@RUG (or student colloquium in full) is a course that master students in computing science follow in the first year of their master study at the University of Groningen.

SC@RUG was organized as a conference for the fourteenth time in the academic year 2016-2017. Students wrote a paper, participated in the review process, gave a presentation and chaired a session during the conference.

The organizers Rein Smedinga, Michael Biehl and Femke Kramer would like to thank all colleagues who cooperated in this SC@RUG by collecting sets of papers to be used by the students and by being an expert reviewer during the review process. They also would like to thank Agnes Engbersen for her very inspiring workshops on presentation techniques and speech skills.

Organizational matters

SC@RUG 2017 was organized as follows. Students were expected to work in teams of two. The student teams could choose between different sets of papers, that were made available through the digital learning environment of the university, *Nestor*. Each set of papers consisted of about three papers about the same subject (within Computing Science). Some sets of papers contained conflicting opinions. Students were instructed to write a survey paper about this subject including the different approaches in the given papers. The paper should compare the theory in each of the papers in the set and include their own conclusions about the subject. Of course, own research was encouraged.
This year one team proposed their own subject.

After submission of the papers, each student was assigned one paper to review using a standard review form. The staff member who had provided the set of papers was also asked to fill in such a form. Thus, each paper was reviewed three times (twice by peer reviewers and once by the expert reviewer). Each review form was made available to the authors of the paper through *Nestor*.

All papers could be rewritten and resubmitted, independent of the conclusions from the review. After resubmission each reviewer was asked to re-review the same paper and to conclude whether the paper had improved. Rereviewers could accept or reject a paper. All accepted papers[1] can be found in these proceedings.

In her lectures about communication in science, Femke Kramer explained how researchers communicate their findings during conferences by delivering a compelling storyline supported with cleverly designed images. Lectures on how to write a paper and on scientific integrity were given by Michael Biehl and a workshop on reviewing was offered by Femke Kramer.

Agnes Engbersen gave workshops on presentation techniques and speech skills that were very well appreciated by the participants. She used the two minute madness presentation (see further on) as a starting point for improvements.

Rein Smedinga was the overall coordinator, took care of the administration and served as the main manager of *Nestor*.

Students were asked to give a short presentation halfway through the period. The aim of this so-called two-minute madness was to advertise the full presentation and at the same time offer the speakers the opportunity to practice speaking in front of an audience.

The conference itself was organized by the students themselves. In fact half of the group was asked to fully organize the day (i.e., prepare the time tables, invite people, look for sponsoring and a keynote speaker, create a website, etc.). The other half acted as a chair and discussion leader during one of the presentations.

The gradings of the draft and final paper were weighted marks of the review of the corresponding staff member (50%) and the two students reviews (each 25%).

Students were graded on the writing process, the review process and on the presentation. Writing and rewriting counted for 35% (here we used the grades given by the reviewers), the review process itself for 15% and the presentation for 50% (including 10% for being a chair or discussion leader during the conference and another 10% for the 2 minute madness presentation). For the grading of the presentations we used the assessments from the audience and calculated the average of these.

In this edition of SC@RUG students were videotaped during their 2 minute madness presentation and during the conference itself using the video recording facilities of the University. The recordings were published on *Nestor* for self reflection.

On 7 April 2016, the actual conference took place. Each paper was presented by both authors. We had a total of 14 student presentations this day.

[1] this year, all papers were excepted

Sponsoring

The student organizers invited one keynote speaker (Jeroen Vlek, CTO of Anchormen. The corresponding company sponsored the event by providing lunch and coffee and concluding drinks.
We are very grateful to

- Anchormen

for sponsoring this event.

Thanks

We could not have achieved the ambitious goals of this course without the invaluable help of the following expert reviewers:

- Vasilios Andrikopoulos
- Michael Biehl
- Jiri Kosinka
- Christian Manteuffel
- Jorge A. Perez
- Jos Roerdink
- Brian Setz
- Nicola Striscuiglio
- Alex Telea

and all other staff members who provided topics and provided sets of papers.
Also, the organizers would like to thank the *Graduate school of Science* for making it possible to publish these proceedings and sponsoring the awards for best presentations and best paper for this conference.
Rein Smedinga
Michael Biehl
Femke Kramer

Since the tenth SC@RUG in 2013 we added a new element: the awards for best presentation, best paper and best 2 minute madness. Therefore, from that edition on, we will have a Hall of Fame:

Best 2 minute madness presentation awards

2017
Stephanie Arevalo Arboleda and Ankita Dewan
Unveiling storytelling and visualization of data

2016
Michel Medema and Thomas Hoeksema
Implementing Human-Centered Design in Resource Management Systems

2015
Diederik Greveling and Michael LeKander
Comparing adaptive gradient descent learning rate methods

2014
Arjen Zijlstra and Marc Holterman
Tracking communities in dynamic social networks

2013
Robert Witte and Christiaan Arnoldus
Heterogeneous CPU-GPU task scheduling

Best presentation awards

2017
Siebert Looije and Jos van de Wolfshaar
Stochastic Gradient Optimization: Adam and Eve

2016
Sebastiaan van Loon and Jelle van Wezel
A Comparison of Two Methods for Accumulating Distance Metrics Used in Distance Based Classifiers

Michel Medema and Thomas Hoeksema
Providing Guidelines for Human-Centred Design in Resource Management Systems

2015
Diederik Greveling and Michael LeKander
Comparing adaptive gradient descent learning rate methods

Johannes Kruiger and Maarten Terpstra
Hooking up forces to produce aesthetically pleasing graph layouts

2014
Diederik Lemkes and Laurence de Jong
Pyschopathology network analysis

2013
Jelle Nauta and Sander Feringa
Image inpainting

Best paper awards

2017
Michiel Straat and Jorrit Oosterhof
Segmentation of blood vessels in retinal fundus images

2016
Ynte Tijsma and Jeroen Brandsma
A Comparison of Context-Aware Power Management Systems

2015
Jasper de Boer and Mathieu Kalksma
Choosing between optical flow algorithms for UAV position change measurement

2014
Lukas de Boer and Jan Veldthuis
A review of seamless image cloning techniques

2013
Harm de Vries and Herbert Kruitbosch
Verification of SAX assumption: time series values are distributed normally

Contents

An Overview of Energy Efficient Scheduling in Data Centers

Win Leong Xuan and Martin Glova

Abstract—Data centers that utilize renewable energy are one of the major trends for protecting the environment and subverting the energy crisis. We can improve the energy efficiency and save costs by matching the computational tasks of a data center to the renewable energy supply. This can be done by using efficient scheduling strategies for computational tasks against the available energy resources in the data center. In this paper, we analyze different scheduling strategies. These strategies are compared based on the utilization rate of the renewable energy, total system running cost, and the task satisfaction rate. We present three models to support the scheduling strategies. Lastly, we conclude that for the different scheduling strategies that there are a few trade-offs, in terms of choosing the best scheduling strategies with regards to the utilization rate, total costs and satisfaction rate.

Index Terms—Green data center, scheduling strategies, energy efficiency, renewable energy.

1 Introduction

Data centers consume a massive amount of energy. In 2010 it was estimated that data centers consume around 1.5% of the total electricity used in world-wide [6]. This has resulted in a massive amount of carbon emissions because most of the energy that was produced comes from fossil fuels. According to a study in 2008, data centers emit on estimated 116 million metric tons of carbon annually [10]. When comparing this value with countries, data centers emit slightly more carbon than the Czech Republic which is ranked 40^{th} in carbon emission world-wide [3]. Furthermore, the energy consumption of data centers is increasing rapidly on a yearly basis. In 2014 data centers consumed about 3% of the global electricity production while producing 200 million metric tons of carbon [2]. The energy consumption has doubled in just 4 years emitting 84 million metric ton of carbon. With the increasing world-wide awareness of environment protection and the energy crisis, it raises the demand for cleaner products and services.

Several companies are planning to build their own, so-called "green data centers", which are partially or completely powered by renewable energy. These data centers either generate their own renewable energy or draw it from an existing renewable energy plant [5]. Companies like Google, Microsoft, Yahoo! and Apple, have started to power their data centers with renewable energy making them less dependent on conventional energy, energy produced using natural oil, gas and coal. Renewable energy sources, like wind and sunlight are mostly intermittent, which makes it more expensive to produce in comparison with conventional energy. This also motivates data centers for self-generating renewable energy plant. This trend will continue, as the capital costs keep going down and governments continue to provide incentives for green power generation to promote renewable energy [8].

The challenge that comes with renewable energy is that it is variable: renewable energy generation is highly dependent and affected by the time of the day, the weather, the season and so on. Therefore, by using the predictions of the renewable energy sources, we can use various scheduling strategies to utilize the renewable energy usage against the computational tasks (workload) of the data center.

In our paper, we describe the different scheduling strategies to utilize renewable energy effectively matching the workload of the data center that is partially powered by renewable energy. We use the prediction information of the renewable energy supply and the grid electricity price to effectively schedule computational tasks of the data center. The goal is then to maximize the utilization of the renewable energy and to reduce the total energy cost.

- *Win Leong Xuan is a MSc. Computing Science student at the University of Groningen, E-mail: w.l.xuan@student.rug.nl.*
- *Martin Glova is a MSc. Computing Science exchange student at the University of Groningen, E-mail: m.glova@student.rug.nl.*

For our research, we focus on different energy efficient scheduling methods like the Smart Green Energy-Efficient Scheduling Strategy (SGEESS), Green-Scheduling Strategy (GSS), Price-Scheduling (PSS) and Greedy-Energy-Efficient Strategy (GEES) [8]. These strategies are evaluated based on the utilization rate of renewable energy, the impact on the total cost and the impact on the satisfaction rate. Three models are also presented for the task energy-efficient scheduling in a partially powered by renewable energy data center. These three models are the Task Model, the Energy Model, and the Scheduling Model.

This paper is organized as follows: Section 3 describes the models that we are going to use for setting up the energy-efficient scheduling strategies and we also describe the problem of our paper. In Section 4 the different scheduling strategies are presented with a brief description of each strategy. Section 5 contains the results for each scheduling strategy. In Section 6 we discuss the results. The Future Works for this research is provided in Section 7. Lastly, in Section 8 we make a conclusion for our research.

2 Related work

Nowadays, research in energy efficient scheduling is a very active topic. GreenSwitch is a strategy for scheduling workload in data centers with using batteries. The greatest difference between the GreenSwitch strategy and strategies described in this paper is using batteries in the GreenSwitch strategy. The input for GreenSwitch strategy it is predictions of future renewable energy availability, predictions of future computational load, the current amount of energy stored in the batteries, analytical models of workload behavior, battery use, and electricity cost and the characteristics of the green data center and the prices of the grid energy and peak power [5]. GreenSwitch prepare configuration for Hadoop which is a bit modified by implementing three server power states: active, decommissioned and sleep [9]. The goal of the strategy is to minimize the grid electricity cost when the grid is up and minimize the performance degradation.

An example of a green data center prototype using GreenSwitch to dynamically manage workload demands is Parasol. There are used multiple energy sources (renewable energy, batteries, and grid) and multiple energy stores (batteries and net metering). Parasol uses air-side economizer cooling whenever outside temperatures are low enough and regular air conditioning otherwise [5].

Wong presented the Peak Efficiency Aware Scheduling (PEAS) for highly energy proportional servers [18]. PEAS consists of a global peak energy efficiency aware scheduler and on each server an energy efficiency profiling daemon. The function of the profiler is to dynamically capture the energy efficiency of each server, which the global scheduler utilizes as a heuristic for scheduling. Wong proposed that instead of naively packing each server until peak utilization, or uniformly spreading the load to all servers, it is more efficient to pack servers until their peak efficiency point to reduce the low utilization, but non-zero, regions of data centers. In comparison with the methods

that are provided within this paper, the PEAS method does not take the overall running cost of the data center into consideration.

In paper [17] authors focus on using control knobs, that modulate the power consumption of IT equipment for optimizing data center electric utility bills. By the control knobs are meant, for example, Dynamic Voltage Frequency Scaling (DVFS) for CPUs, server/cluster shutdown, load balancing and scheduling of jobs/requests, partial execution to consume lower power, offer low-quality results or smart cooling systems, etc. The authors are not concerning with green energy specifically as we are in the paper.

3 MODELS

We provide in this section the three mathematical models that help us describe the different scheduling methods. In Figure 1 we can see a system architecture for a partially powered data center by renewable energy and conventional energy. The data center consists of m amount of computing nodes and one scheduler. The main job of the scheduler is to submit tasks for the computing nodes to execute these tasks. The tasks are being submitted by the users. The scheduler first obtains the prediction of the renewable generation, and then collect the status information of the data center system which is

1. tasks running on each node,
2. tasks located in the local queues of each computing node,
3. the start and finish time of the tasks.

By using this information, the scheduler can then use the defined scheduling strategy to assign tasks to the suitable computing node and their position in the local queues.

Due to the intermittent nature of the renewable energy sources, the output of renewable energy is inconsistent which makes utilizing between renewable energy and conventional energy harder in practice. For this paper the prediction of renewable energy generation is not the main focus, we consider a short-term renewable energy prediction as input for the scheduler.

As mentioned, conventional energy is also used in the data center in the case, when the available renewable energy is not sufficient for the demand. Grid electricity prices vary depending on the time of the day. Therefore, by using the grid price as input to the scheduler, we can determine the cost for using grid energy. This is important as the grid electricity price also affects the operation cost of the data center. By using the three models proposed by Lei et al. [8] which are the Task, Energy and Scheduling model, we define the various energy efficient scheduling methods.

3.1 Task Model

We consider tasks as soft real-time tasks which are independent and aperiodic, which means each task can only be assigned to one computing node and cannot be partitioned among multiple nodes. The task set is denoted as $T = \{t_1, ..., t_i, ..., t_{|T|}\}$. Each task is represented by $t_i = \{at_{ij}, l_i, w_i, dt_i\}$, where, at_{ij} is the arrival time of the task at computing node n_j, l_i is the length of computing time of the task t_i, w_i denotes the importance measurement of the task and dt_i is the deadline of the task. We assume that l_i is constant since the computing nodes are homogeneous and they all run at the same frequency and voltage. The start time st_{ij} of each task on a computing node can be computed as follows:

$$st_{ij} = max\{at_{ij}, ft_{0j} + 1\} + \sum_{t_k \in T_j^i} l_k, \tag{1}$$

where, ft_{0j} denotes the finish time of the task which is running when the task t_i arrives. We can consider $ft_{0j} = 0$ if there is no task running and T_j^i is the task set run before task t_i in the waiting queue on computing node n_j. By determining the start time st_{ij}, we find the maximum time between the arrival time at_{ij} and the finish time $ft_{0j} + 1$ of the currently running task. By incrementing ft_{0j} with 1, we can determine the time that the next task can be started. Therefore, if there is also other tasks in the waiting queue T_j^i, we sum the length l_k of each task and sum this up with the start time of the next task to determine the start time st_{ij} of each task. We can then compute the finish time ft_{ij} of task t_i on computing node n_j as follows:

$$ft_{ij} = st_{ij} + l_i = max\{at_{ij}, ft_{0j} + 1\} + \sum_{t_k \in T_j^i} l_k + l_i. \tag{2}$$

Each task is partitioned into two categories. In our case, this would be the "crucial" and the "non-crucial" tasks. Crucial tasks require an urgent response. Non-crucial tasks are common tasks and do not require the urgent response. Therefore, the value w_i for each task t_i can be denoted as $\{crucial, noncrucial\}$. Crucial tasks have priority running on a computing node.

When a new task t_i arrives at computing node n_j, we analyze if task t_i is crucial or not. If task t_i is crucial, it is placed in the local queue of computing node n_j at the end of the "crucial" tasks but before the "non-crucial" tasks. If task t_i is not crucial, it is placed at the end the local waiting queue of computing node n_j.

However, each task in the local waiting queue has a deadline. Inserting new tasks into the queue may cause the violation of certain deadlines, especially when we are inserting "crucial" tasks into the queue. When inserting new tasks into the queue, we look at 3 cases for the local waiting queues.

- If the tasks waiting in the local queue Q_j have looser deadlines, this means that there is no violation of the deadlines of the tasks in Q_j.
- If the tasks waiting in the local queue Q_j have some tight deadlines, then the insertion of a new task may cause some violation of the deadlines.
- If the tasks waiting in the local queue Q_j have tighter deadlines, then the insertion of new tasks may lead to chain violations of deadlines.

Figure 2 illustrates different cases when inserting new crucial tasks into the local queue of each computing node.

3.2 Energy Model

The scheduling time horizon is the interval of time in which is divided into time slots to determine the energy at each time slot e.g. the period of one day divided by hours. The scheduling time horizon is denoted as $PT = \{pt_1, ..., pt_k, ..., pt_{|PT|}\}$ where, pt_k is the k-th scheduling time slot and $pt_{|PT|}$ is the last scheduling time slot in which the last task is finished. The energy model of the data center can be defined as follows:

$$< \overline{RG}, \overline{CG}, \widetilde{RG}, \widetilde{CG}, D >, \tag{3}$$

where, $\overline{RG} = \{\overline{rg}_1, ..., \overline{rg}_k, ..., \overline{rg}_{|PT|}\}$ is the prediction output of the renewable energy during the scheduling time horizon PT, $\overline{CG} = \{\overline{cg}_1, ..., \overline{cg}_k, ..., \overline{cg}_{|PT|}\}$ is the planning conventional energy supply during PT, $\widetilde{RG} = \{\widetilde{rg}_1, ..., \widetilde{rg}_k, ..., \widetilde{rg}_{|PT|}\}$ is the deviation of the real output to the prediction output of the renewable energy, $\widetilde{CG} = \{\widetilde{cg}_1, ..., \widetilde{cg}_k, ..., \widetilde{cg}_{|PT|}\}$ is the deviation of the real supply to the planning conventional energy supply and $D = \{ed_1, ..., ed_k, ..., ed_{|PT|}\}$ is the energy demand of the data center during the scheduling time horizon PT. In order to ensure the energy supply of the data center, we have the following condition as:

$$\overline{rg}_k + \widetilde{rg}_k + \overline{cg}_k + \widetilde{cg}_k \geq ed_k \qquad (k = 1, ..., |PT|). \tag{4}$$

The energy demand D of the data center is dependent on the energy consumption E. The energy consumption E consists of two components which are the dynamic energy consumption E_{dyn} for the activities of the devices and circuit in the system, and the static energy consumption E_{sta} is for the leakage currents in the devices and circuit. Therefore, the energy demand D can be calculated as follows:

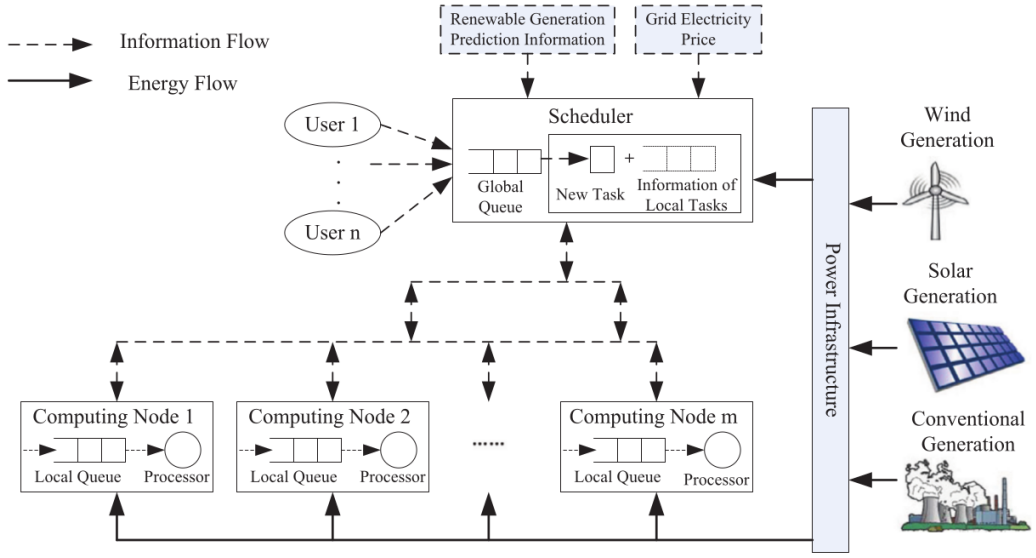


Fig. 1: Architecture of data center [8].

$$D = E = E_{dyn} + E_{sta} \tag{5}$$

In practice, the energy consumption of a data center consists mostly of dynamic energy consumption [19]. Therefore, we mainly focus on the dynamic energy consumption. We approximate the dynamic energy consumption which is equivalent to the total energy consumption of the data center by the processors of the computing nodes during the scheduling time horizon PT. Formally, we consider the following formulation for the energy demand for the k-th scheduling time slot:

$$\begin{aligned} ed_k &= E_{pt_k} \\ &= E^{dyn}_{pt_k} \\ &= \sum_{j=1}^{m} E_{j,pt_k} \\ &= \sum_{j=1}^{m} (E^{act}_{j,pt_k} + E^{idl}_{j,pt_k}) \qquad (k = 1, ..., |PT|), \end{aligned} \tag{6}$$

where, $pt_k \in PT$ is k-th time slot, E_{j,pt_k} is the total energy consumption of computing node n_j at the k-th scheduling time slot. E^{act}_{j,pt_k} and E^{idl}_{j,pt_k} are the energy consumption of each computing node n_j when they are in the active and idle state at the k-th scheduling time slot.

3.3 Scheduling Model

The maximization of the utilization of renewable energy for scheduling model is defined as:

$$max \left\{ \frac{\sum_{k=1}^{|PT|} rg_k}{\sum_{k=1}^{|PT|} (\overline{rg}_k + \widetilde{rg}_k)} \right\}, \tag{7}$$

where,

$$rg_k = \begin{cases} E_{pt_k} & \text{if } \overline{rg}_k + \widetilde{rg}_k \geq E_{pt_k}, \\ \overline{rg}_k + \widetilde{rg}_k & \text{otherwise.} \end{cases} \tag{8}$$

rg_k is the real used renewable energy at the k-th time slot. The minimum of the price is computed by the sum of all the prices in all the time slots. The satisfaction rate is computed as follows

$$max \left\{ \frac{\overline{n}_{PT}}{n_{PT}} \right\}, \tag{9}$$

where, n_{PT} is the count of tasks during the scheduling time horizon PT and $\overline{n}_{PT}$ is the count of the tasks that are finished before their deadlines.

4 Methods

The main goal of the scheduling methods is to pick the best computing node and eventually the best time period slot for a new task. We describe four methods in this section: Smart green energy-efficient scheduling strategy (SGEESS), Green-Scheduling Strategy (GSS), Price-Scheduling Strategy (PSS) and Greedy-Energy-Efficient Strategy (GEES) [8]. The output of all the scheduling method is the same which gives a computing node and a suitable time slot according to the scheduling method. We describe the different methods with the following equation which returns a computing node to which the task is scheduled to:

$$n(N) = \arg \min_{n_\gamma \in N} \left\{ \sum_{t \in T^{local}_\gamma} l_t \right\}, \tag{10}$$

where, N is a subset of all computing nodes CN used as the input for Equation 10, $n_\gamma \in N$ represents the iteration throughout the N set for each computing node n_γ and T^{local}_γ is the set of tasks in the local queue of each computing node n_γ.

We define four different cases when choosing a computing node n^i and a time slot pt^i to schedule the new incoming task t_i:

Case 1 $pt^i = pt_k \in PT$ is the time slot with the largest remaining prediction amount rg of renewable energy when task t_i can be scheduled without violation of the task deadlines, $rg \neq 0$, and subset of computing nodes is defined as $N = \{n_\gamma | n_\gamma \in CN, pt_{k-1} \leq st_{i\gamma} < pt_k, st_{i\gamma} + l_i \leq dt_i, \forall t_z \in T^i_\gamma : st_{z\gamma} + l_z \leq dt_z\} \neq \emptyset$, where T^i_γ is the set of tasks running after task t_i on n_γ after insertion of t_i.
n^i is the computing node defined by Equation 10 as follows $n^i = n(N)$.

Case 2 $pt^i \in PT$ is the time slot with the cheapest grid electricity price when task t_i can be located without violation of task deadline.

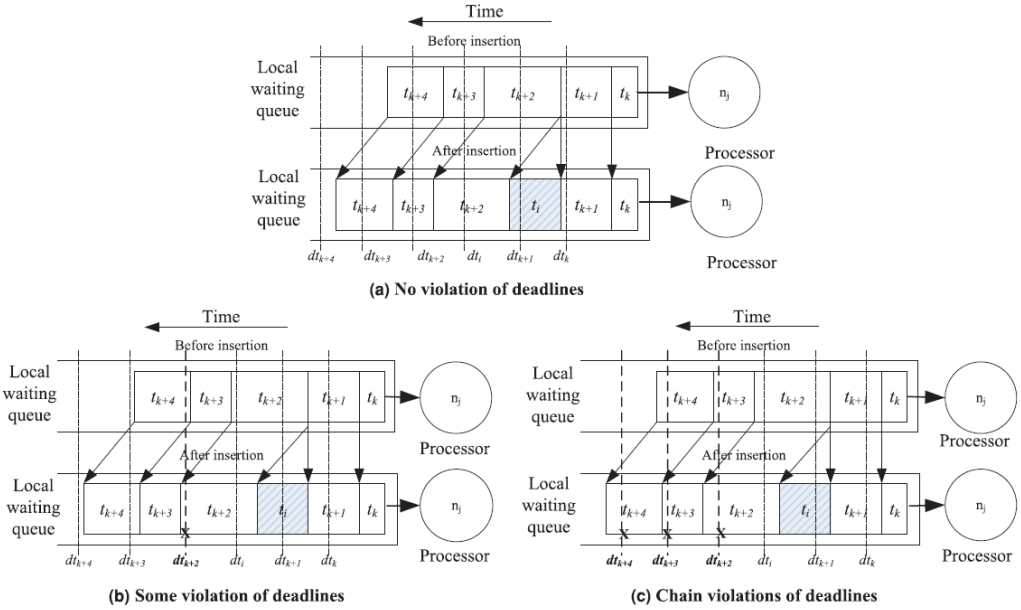


Fig. 2: Assigment of new "crucial" tasks [8].

N is defined as in Case 1.
n^i is the computing node defined by Equation 10 as follows $n^i = n(N)$.

Case 3 $pt^i \in PT$ is the time slot with the largest remaining prediction amount rg of renewable energy, and $N = \{n_\gamma | n_\gamma \in CN, pt_{k-1} \leq st_{i\gamma} < pt_k, \forall t_z \in T^i_\gamma : st_{z\gamma} + l_z \leq dt_z\} \neq \emptyset$, where T^i_γ is defined same as in Case 1.
n^i is the computing node defined by Equation 10 as follows $n^i = n(N)$.

Case 4 $pt^i \in PT$ is the time slot with the cheapest grid electricity price, and N is defined as in Case 3.
n^i is the computing node defined by Equation 10 as follows $n^i = n(N)$.

4.1 SGEESS

The implementation of SGEESS uses as input a new task t_i (as described in Section 3.1). For each new task, the predicted amount of the renewable energy with its deviation (as described in Section 3.2), and a grid electricity price is updated in the scheduler.

This scheduling method tries to find the feasible time slot with the largest remaining predicted amount of renewable energy, without the violation of the deadlines of tasks as described in Case 1. If such time slot exists, the scheduler continues computing pt^i and n^i. If not, SGEESS tries to find time slot using Case 2 and then Case 4 to compute pt^i and n^i as the output.

4.2 GSS

The input for GSS is similar to SGEESS. The scheduling method uses a new task as input and predicted amount of the renewable energy is updated in the scheduler.

At first, GSS tries to find the time slot with the largest remaining predicted amount of renewable energy, without violating deadlines of tasks as described in Case 1, but GSS does not require $rg \neq 0$. If there is no such time slot, GSS defines subset N by using Case 3. In Case 3, the deadlines of the original tasks are kept in the local queue of the selected node and the scheduler selects the time slot with the larger remaining predicted amount of renewable energy.

4.3 PSS

For PSS, the input is defined by a new task and the grid electricity price is updated in the scheduler. Afterwards, the scheduler tries to fulfill Case 2, so the computing node that does not violate the task deadlines and the time slot with the cheaper grid electricity price is chosen. If such time slot does not exist, Case 4 is applied. In Case 4, the scheduler chooses the time slot with the cheapest grid electricity price, with respect to the deadlines of the original tasks in the local queue of the selected node.

4.4 GEES

GEES is very similar to SGEESS. The only difference is that the predicted amount of the renewable energy is only updated once in the scheduler. In SGEESS, the predicted amount of the renewable energy is updated for each new task. Choosing the suitable time slot and computing node is then computed the same way trying to apply cases in the following order: Case 1, Case 2 and Case 4.

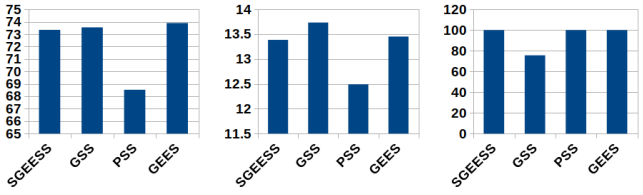
5 Results

We compare the results of strategies with the impact on the renewable energy input, the number of computing nodes, the deadline of the task and the length of the task. Used metrics for evaluating the strategies are:

- Equation 7 multiplied by 100% for computing utilization of renewable energy (*RE utilization*),
- total cost is computed as sum of costs in all the time slots PT depending on the total energy consumptions in all the nodes.
- Equation 9 multiplied by 100% for computing satisfaction rate of tasks,

Figure 3 shows results with the impact of the renewable energy input. Renewable predictions are based on real data of the solar generation and in results is computed the mean of six values with increasing impact of renewable energy. We can observe that the utilization rates of renewable energy under GSS, GEES, and SGEESS are higher (more than 73%) but the PSS has the lowest utilization rate (less than 69%). On the other hand, PSS has the best results for the total cost (in this case the lowest value is better). The GSS has the lower task satisfaction rate. It is because GSS reacts very sensitively to changes in

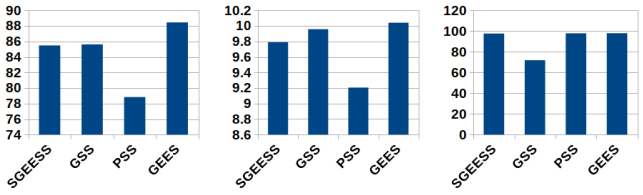
renewable energy. However, when there exists more renewable energy that can be used in the reasonable time slots, the scheduling can be more elastic and the task satisfaction rate certainly goes up.



(a) RE utilization. (b) Total cost. (c) Task satisfaction.

Fig. 3 Impact of the renewable energy input.

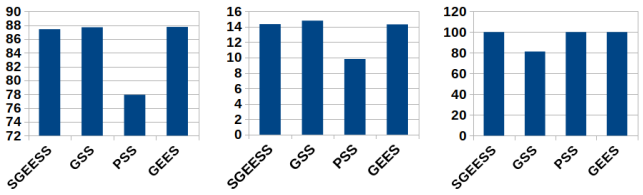
Figure 4 illustrates results with the impact of the number of computing nodes. We can directly see poor results for GSS (in Figure 4b and 4c). For PSS, as in is with the impact of the renewable energy input, values with respect to renewable energy utilization are not good but the price is the best as it was in the previous case.



(a) RE utilization. (b) Total cost. (c) Task satisfaction.

Fig. 4 Impact of the number of computing nodes.

The impact of the deadline of the task is visualized in Figure 5. The SGEESS, GSS, and GEES have very similar utilization rates of renewable energy, and the PSS strategy has the lowest utilization rate. But PSS reaches the lowest total cost values comparing with other strategies. For the task satisfaction, all the strategies are almost the same except GSS which the worst results.



(a) RE utilization. (b) Total cost. (c) Task satisfaction.

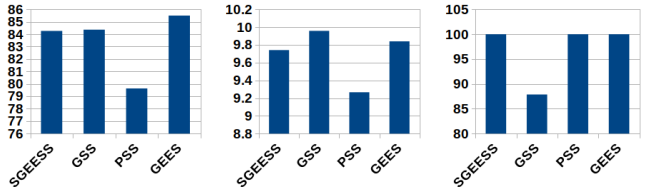
Fig. 5 Impact of the deadline of task.

The last comparison deals with the impact of the length of a task. The results are shown in Figure 6. For utilization of the renewable energy the worse rate is for PSS. SGEESS and GSS are slightly worse than GEES. Lowest total cost is reached (as in each other cases) by PSS. The task satisfaction rate drops under GSS because the task scheduling becomes tougher when the task length becomes longer and the violation of task deadline becomes more likely.

6 Discussion

Figure 7 shows total comparison of the results. The values presented in the bar charts are computed as the mean of all the experiments provided in the previous section and divided into three categories:

1. renewable energy utilization (Figure 7a),
2. total costs (Figure 7b),



(a) RE utilization. (b) Total cost. (c) Task satisfaction.

Fig. 6 Impact of the length of task.

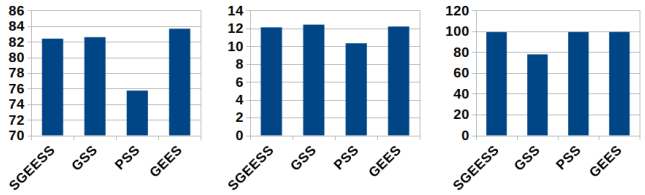
3. task satisfaction (Figure 7c).

This way we can compare all algorithms with impact to these categories.

Figure 7a compares computed utilization of renewable energy for each scheduling method. We can see that the worse results are for PSS algorithms and best for GEES. Algorithms SGEESS and GSS have very similar results as GEES. The difference is only little more than 1%.

In the total cost comparison, PSS has the best results and other methods are almost on the same level as shown in Figure 7b. It is reasonable as far as PSS optimize new task with choosing time slot with the cheapest electricity price.

Comparing the results for task satisfaction, as shown in Figure 7c, we can conclude that GSS has a lower rate. All other algorithms are almost at the same level, in average very near to 100%. Depending on the purpose, presented advantages and disadvantages and by choosing a subject of optimization we can choose an appropriate strategy.



(a) RE utilization. (b) Total cost. (c) Task satisfaction.

Fig. 7 Comparing of strategies.

7 Future Works

Improvements for the energy efficient scheduling methods could be provided in more aspects. Firstly, we can optimize the scheduling methods themselves in terms of complexity e.g. by adding additional parameters. Secondly, we can also add additional criteria when utilizing green renewable energy for the scheduling methods.

The scheduling methods provided within this paper only uses the data provided at each time slot to schedule new tasks. This means we do not take future incoming tasks into consideration. Using machine learning methods for scheduling the new tasks can bring some improvements [7]. Minton [11] provided an overview of planning or scheduling methods using machine learning. For example by using Q-learning for effective planning or scheduling of new incoming tasks [16]. By predicting the new incoming tasks within a time period e.g. new incoming tasks within 1 week, we can optimize the scheduling methods to reduce the overall running costs more within the predicted time period.

We can also optimize the utilization of green energy by making some improvements in practical aspects. The majority of the electricity used by data centers is for cooling. Therefore, by extracting the heat generated from each server within the data center, we are able to reuse this heat for district/plant/water heating [4]. As mentioned, 25% of energy consumption of data centers is being used only for cooling [1, 12, 15, 13]. Also, we can distribute the load of a data center across

multiple data centers where the amount of renewable energy is higher and easier to produce in order to execute the tasks using less fossil fuel as possible.

Recycling of Li-ion battery is also a trending topic [14]. Li-ion battery usage is expected to triple by 2025 [14]. Therefore, by using recycled Li-ion batteries we are able to store the renewable energy generated when the utilization of a data center is low and use this energy when it is needed. By using data mining methods, we are able to predict the weather.

8 Conclusion

In this paper, we introduce the motivations of using green energy and optimize the level of increasing energy use in data centers. The main reason is the increasing of carbon emission necessity of decreasing using conventional energy sources, but also energy at all because of protection of the environment and decreasing costs. To do so we present mathematical models, different types of scheduling strategies based on these models and results of experimental use of these strategies.

Results section shows that optimization can be provided with impact to different features looking at results from different perspectives. As the impacts, we use renewable energy input, the number of computing nodes, the deadline of the task and the length of the task. As result measures, we use renewable energy utilization, total costs, and task satisfaction. We can not pick the best strategy because results use different optimization strategies (ex. PSS has poor results looking at renewable energy utilization but the best results regarding total cost).

However, the main problem for green energy is that it is not easy to predict the amount of produced energy and schedule it as a primary source. The reason is that it depends on many different factors. For solar energy, it could be the time of day. The wind energy depends mostly on weather, season or geographical area. When there is not enough green energy, strategies should optimize the use of conventional energy to decrease the production of fossil fuels as much as possible.

Acknowledgements

The authors wish to thank Brian Setz, for his expert review and other classmates for reviewing this paper.

References

[1] Efficient cooling of the data center. http://www.rahisystems.com/blog/cooling-of-the-data-center/. Accessed: 2017-03-27.

[2] Undertaking the challenge to reduce the data center carbon footprint. http://www.datacenterknowledge.com/archives/2014/12/17/undertaking-challenge-reduce-data-center-carbon-footprint/. Accessed: 2017-03-28.

[3] World carbon emissions: the league table of every country. https://www.theguardian.com/environment/datablog/2012/jun/21/world-carbon-emissions-league-table-country. Accessed: 2017-03-06.

[4] K. Ebrahimi, G. F. Jones, and A. S. Fleischer. A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities. *Renewable and Sustainable Energy Reviews*, 31:622–638, 2014.

[5] Í. Goiri, W. Katsak, K. Le, T. D. Nguyen, and R. Bianchini. Parasol and greenswitch: Managing datacenters powered by renewable energy. In *ACM SIGARCH Computer Architecture News*, volume 41, pages 51–64. ACM, 2013.

[6] J. Koomey. Growth in data center electricity use 2005 to 2010. *A report by Analytical Press, completed at the request of The New York Times*, 9, 2011.

[7] C.-Y. Lee, S. Piramuthu, and Y.-K. Tsai. Job shop scheduling with a genetic algorithm and machine learning. *International Journal of production research*, 35(4):1171–1191, 1997.

[8] H. Lei, T. Zhang, Y. Liu, Y. Zha, and X. Zhu. Sgeess: Smart green energy-efficient scheduling strategy with dynamic electricity price for data center. *Journal of Systems and Software*, 108:23–38, 2015.

[9] J. Leverich and C. Kozyrakis. On the energy (in) efficiency of hadoop clusters. *ACM SIGOPS Operating Systems Review*, 44(1):61–65, 2010.

[10] J. Mankoff, R. Kravets, and E. Blevis. Some computer science issues in creating a sustainable world. *Computer*, 41(8), 2008.

[11] S. Minton. *Machine learning methods for planning*. Morgan Kaufmann, 2014.

[12] J. D. Moore, J. S. Chase, P. Ranganathan, and R. K. Sharma. Making scheduling" cool": Temperature-aware workload placement in data centers. In *USENIX annual technical conference, General Track*, pages 61–75, 2005.

[13] C. D. Patel, C. E. Bash, and A. H. Beitelmal. Smart cooling of data centers, June 3 2003. US Patent 6,574,104.

[14] P. Patel and L. Gaines. Recycling li batteries could soon make economic sense. *MRS Bulletin*, 41(06):430–431, 2016.

[15] R. K. Sharma, C. E. Bash, C. D. Patel, R. J. Friedrich, and J. S. Chase. Balance of power: Dynamic thermal management for internet data centers. *IEEE Internet Computing*, 9(1):42–49, 2005.

[16] R. S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the seventh international conference on machine learning*, pages 216–224, 1990.

[17] C. Wang, B. Urgaonkar, Q. Wang, and G. Kesidis. A hierarchical demand response framework for data center power cost optimization under real-world electricity pricing. In *Modelling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2014 IEEE 22nd International Symposium on*, pages 305–314. IEEE, 2014.

[18] D. Wong. Peak efficiency aware scheduling for highly energy proportional servers. In *Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on*, pages 481–492. IEEE, 2016.

[19] D. Zhu, R. Melhem, and B. R. Childers. Scheduling with dynamic voltage/speed adjustment using slack reclamation in multiprocessor real-time systems. *IEEE Transactions on Parallel and Distributed Systems*, 14(7):686–700, 2003.

Explaining Multidimensional Visualization Techniques

Carlos H. Paz Rodriguez and Harry Jackson Arroyo

Abstract—Visualizing multi-dimensional datasets can be confusing, especially because the concept of more than 3 dimensions is hard to understand. In order to study high-dimensional datasets on a 2D screen, many techniques have been developed. We briefly talk about analyzing data, dimensionality reduction, and color mapping, among other introductory topics and then we proceed to talk about some visualization tools and techniques for multi-dimensional datasets. We start with hue-based scatter plots, then continue on with attribute based clustering and then with multiple correspondence analysis for categorical data; we finish with a brief discussion on how to go from high-dimensional to three-dimensional datasets, how an interactive technique can be applied to any Dimensionality reduction (DR) technique and lastly by quickly comparing 2D and 3D visualizations. There is no ideal tool, each has its own analysis goal and its limitations. Implementation effort and ease of interpretation are two difficult challenges. Many more visualization tools are still under development.

Index Terms—Multidimensional visualization, dimensionality reduction, hue based scatter plot, attribute based clustering, MCA, three-dimensional projection interaction

1 Introduction

With the increasing amount of available data, an interest in exploiting this information has emerged. Companies try to find relations and patterns in this data that describe past events and predict possible outcomes, in order to make better decisions and strategic plans.

In the simple scenario of 2 or 3 variables we can easily plot the observations with traditional tools (bar charts, histograms, scatter plots, 3D surfaces, 3D cubes, etc), however when we want to visualize information with many variables, a tool able to deal with those datasets becomes necessary.

Given the limitation of displaying results in a 2 dimensional screen or paper sheet most solutions generate a 2D or 3D visualization.

Multiple visualization techniques and tools have been developed, they attempt to summarize high dimensional information, each tool addresses different problems and have particular advantages and weaknesses for each scenario.

In this paper we review relevant studies for visual analysis techniques for multidimensional datasets, in order to provide the reader with a better understanding of the different visualization scenarios, the available tools and the pros and cons of each one.

We first discuss the pre-visualization analysis that typically has to be done in order to choose the appropriate technique, like converting numerical to categorical data and normalization. Once the dataset is prepared, we briefly enumerate the most common algorithms like Principal Component Analysis and its variations or Distance based projections, that provide the fundamentals for all of the following tools and techniques which even though differ on approach, they all share the same mathematical framework.

Finally, we address a list of the most representative tools, from a simple hue based scatter plot or worlds within worlds technique to very complex ones like multiple correspondence analysis using voronoi cells or semantic graphics. For each of which we provide an example, a short explanation and the main advantages and disadvantages.

2 Analyzing and pre-processing the data

The first step is to understand what kind of data we have, in order to select an appropriate tool. In this phase we can adjust the data to fit our models, we can also exclude irrelevant features in order to focus on a particular analysis and improve the performance. It is also important to normalize the data in order to compare different magnitudes correctly.

- *Carlos H. Paz Rodriguez is a MSc. Computing Science student at the University of Groningen, E-mail: p.r.carlos@student.rug.nl.*
- *Harry Jackson Arroyo is a MSc. Computing Science student at the University of Groningen, E-mail: h.jackson.arroyo@student.rug.nl.*

2.1 Data structure

A dataset is just a collection of observations (data points) and attributes (or dimensions) related to this observations.

Attributes can be quantitative or categorical. Quantitative data can be either numerical (integer or decimal numbers) or ordinal (ordered list of values like ”small”, ”medium” and ”large”). Categorical data are simply ”labels” that assign the observation to a particular category and that does not have any particular order.

2.2 Dimensionality reduction

We live in a physical three dimensional world, thus it is difficult for us to think of visualizing data in spaces of higher order. One ”natural” approach is the Worlds within Worlds [6] where the authors intend to nest a whole 3D plot inside each cell of an outer 3D plot iteratively, allowing the user to slice and zoom through 3D spaces into higher or lower level of detail.

All modern visualization techniques have limitations, being them resolution capacity, computational processing limits, color coding limits, etc. Some algorithms like Principal Component Analysis (PCA) or its general version for categorical data, Multiple Component Analysis (MCA), try to find correlations between dimensions in order to detect and remove redundant attributes. Visualization tools like the one proposed by Bertjan Broeksema et. al [2], worlds within worlds [6], semantic interactions [5], hull based visualizations [8] allow the user to have a first glimpse of the data, and further refine the attributes and values that become of interest for the particular analysis.

Projection algorithms explain the same information with less number of dimensions, this allows us to plot data into simple 2D plots. However this reduction comes with loss of information so we need additional information explaining the visualization that helps us understand the data without arriving to wrong conclusions about it.

Most multidimensional models work with numerical data, since they try to map high dimensional datasets into 2D or 3D space keeping the distances between points as similar as possible as in the complete dimensional space. Renato R. O. da Silva et. al. [4] proposes a model which uses Euclidean distances and variance between dimensions to discover the most relevant dimensions for a particular neighborhood of data points. However, in many scenarios some attributes are not numerical, therefore we need to use a different approach. As explained by Bertjan Broeksema et. al. [2] Multiple Correspondence Analysis (MCA) is used for converting categorical data into numerical using a binary encoding mask. After the transformation projection algorithms like Single Value Decomposition (SVD) can be applied to project the data into a 2D model.

3 Projection algorithms

Although there are various techniques for reducing the dimensionality of data, most of them share the same principles and are just adapted for particular situations.

3.1 Principal Component Analysis

With the use of eigen-vectors and eigen-values, PCA finds covariances between attributes, and transforms the data into a new set of attributes that maximizes the variance and reduces the number of dimensions required for explaining the data.[9]

3.2 Multiple Correspondence Analysis

"CA generalizes PCA by using the importances of all observations and attributes to discriminate between observations. CA computes two sets of factor scores, one for observations and one for attributes. Since both score sets share the same variance, they can be both shown in the same 2D scatterplot, which helps the reading of such plots. Multiple Correspondence Analysis (MCA) extends CA to handle categorical data." [2].

3.3 Multidimensional Scaling

"MDS projects N-dimensional points in $K < N$ dimensions while trying to keep distance ratios between projected point pairs and original point pairs." [2].

3.4 Distance based algorithms

On a more geometrical context, the other most common "projection" algorithm is based on distance between points. It is possible to calculate the Euclidean distances (or any other distance) between observation points in order to detect the most relevant dimensions, and then map the observations on a 2 dimensional plot, but keeping the relative distance between all data points.

4 Visualization tools

In this section we discuss the most common visualization tools and we give a basic explanation of how to read them.

4.1 Hue based scatter plot

On this section we analyze a dataset of recorded audios of different persons saying the alphabet letters. As seen on figure 1, each data point represents an observation (left), on this particular technique Principal Component Analysis algorithm was used to reduce the dimensionality, therefore we can conclude that the axis correspond to 2 important principal components, data points keep their relative distance so we can use hue shapes to find clusters of similar elements (center). Each cluster is color coded as displayed on the legend below. On dense regions the relative distances very small, thus the certainty of a point to belong to different clusters is high, this is visualized on the third image (right) where we can see a plot of the projection precision, dark colors mean low precision and clear read color means high precision.

Table 1. Hue based scatter plot comparison.

Advantages
• Scatter plot helps easily visualize the relative distance between observations and find groups of observations. • Color mapping help easily distinguish between clusters.
Disadvantages
• The more colors are used the more difficult it gets to distinguish between different clusters. • On dense regions it is difficult to tell with enough certainty that a point belong to a specific cluster group.

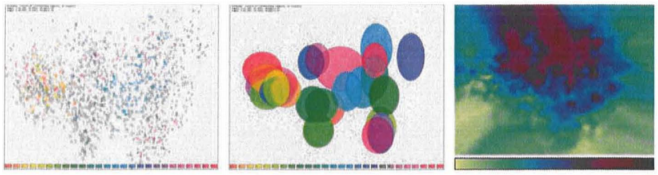


Fig. 1. PCA-based projection of the high-dimensional ISOLET spoken letter dataset to 2D. Left: A scatter plot using printed class labels. Each data sample belongs to one of 26 classes. Middle: A plot using hull-based aggregation of points by class membership. Rainbow colors are used to help distinct point classes by the user. Right: Visualization of projection precision in an interpolated precision map image. [8]

4.2 Attribute based clustering

A dataset of wine quality is shown with 12 dimensions, Local Affine Multidimensional Projection (LAMP) was used for dimensionality reduction. As seen on figure 2, the idea is to explain the data with the least amount of dimensions, therefore observation points are colored with the most representative attribute. With the use of color mapping we can visually detect the most important dimensions and how much of the data they describe. However not all data points can be described with one or a few dimensions, therefore an additional explanation tool needs to be added, which says how many data points are best described by that dimension, this tells us that if many dimensions have similar count of observations, then more than 1 dimension is necessary to explain those data points.

Table 2. Attribute based clustering comparison.

Advantages
• Visualized regions easily allow to understand what is the most important feature. • The count of occurrences helps us decide which dimensions we should use for further analysis.
Disadvantages
• Again color coding limitations. • When many dimensions have similar relative importance, we need to go into deeper detail to understand how many features are important.

4.3 Multiple Correspondence Analysis for categorical data

One of the most complete tools, it uses MCA for dealing with both categorical and numerical data. It allows to visually find relations between attribute values, so we can further turn them into clusters of similar occurrences. By using Voronoi cells it displays different attribute values, also aided with color mapping and brightness. This tool allows the user to interactively choose which attributes and attribute values to visualyze, also it allows to group contiguous cells. It provides a relevance bar at the right which helps understand which attributes are more frequent in the observations. Since MCA computes factors which are mapped to the x and y axis, 2 additional bar plots are added which help understand how original dimensions contribute the most to the classification. A third bar plot called error plot, sums up all the missing attributes contribution to the plot, in case they are not being displayed.

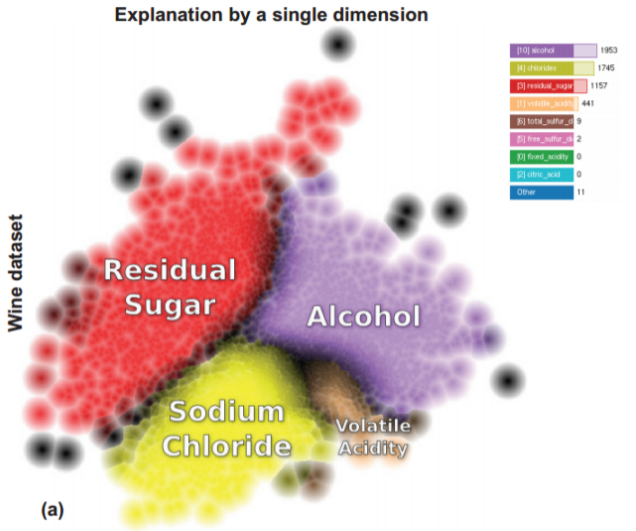


Fig. 2. Visual explanations using a single dimension Portuguese vinho verde wine [4]

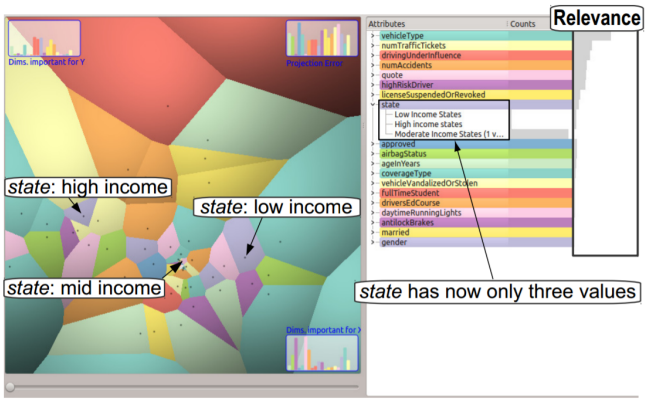


Fig. 3. MCA projections view and dimensions view of car insurance data using Voronoi cells [2]

Table 3. MCA for categorical data comparison.

Advantages
• Each color corresponds to an attribute, so we can visually grasp the distribution of values and the proximity between different attributes values mean that they usually occur together. • The grouping option allows to have color labels available to display other values.
Disadvantages
• The size of the cell does not tell anything about the data, outlier values are always displayed near the edges and frequent values tend to clamp on the center of the plot. • As the user applies filtering and grouping the orientation of the plot keeps turning around, so it gets difficult to compare one analysis with another.

4.4 Worlds within worlds

Worlds within worlds, embed 3 dimensional spaces into other 3 dimensional space, therefore we first choose a value of the outest 3 dimensional cartesian plane, and by keeping those values constant, we can zoom in and visualize a nested 3 dimensional cartesian plane. On figure 4 we visualize stock market values, for each combination of strike value, time to maturity and foreign interest rate, we have a complete nested 3 dimensional plot that shows the value, volatility and the spot price.

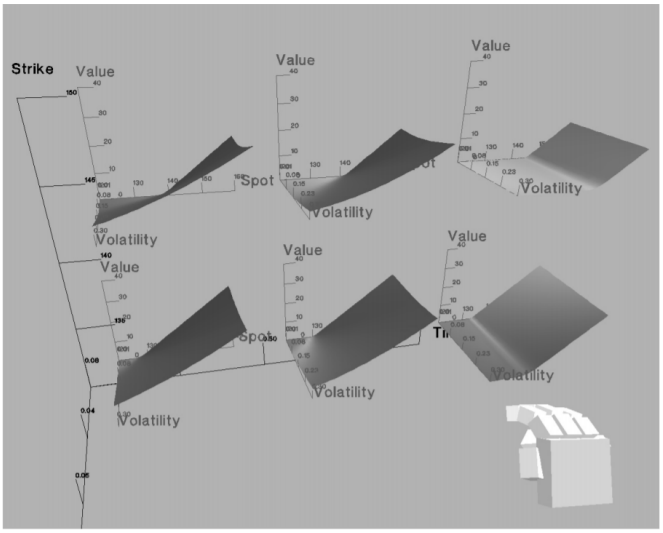


Fig. 4. Worlds within worlds metaphor for high dimensional stock data [6]

Table 4. Worlds within worlds comparison.

Advantages
• Very intuitive because it emulates a real virtual world. • Traditional 3D plots can be shown inside each world.
Disadvantages
• Depending on the setup of dimensions to axis, we might not be able to visualize an important trend from an outer world, because we only see it at static values. • Since it requires user interaction, it cannot easily be printed out or displayed on a screen and have a general overview of the data.

4.5 Semantic graphs

A very particular application is the semantics interaction tool, it keeps track of the user interactions, and tries to predict the user reasoning, highlighting in the end important features according to the analyst actions. On this particular example PCA was used to reduce the dimensionality of text documents. Figure 5, shows the evolution of detecting text documents related to a terrorist attack.

4.6 Contingency wheel

Contingency wheel [1] allows the analysis of high dimensional data by using the most contributing factors of PCA algorithm as pivotal attributes displayed as pie slices, inside each slice it shows the number of occurrences. With the use of color coding and a customizable parameter, it allows the user to visualize how frequent a certain combination of values occur, with the use of the connecting lines. In figure 6, we visualize book publishing by author and by country, we observe that for a particular writer, some countries are strongly related, which in this case is due to the language they share.

Table 5. Semantic graphs comparison.

Advantages
• User interacts only with filtering and selecting particular values. • Visual reading is very simple.
Disadvantages
• The system depends on the user expertise to provide filtering and actions.

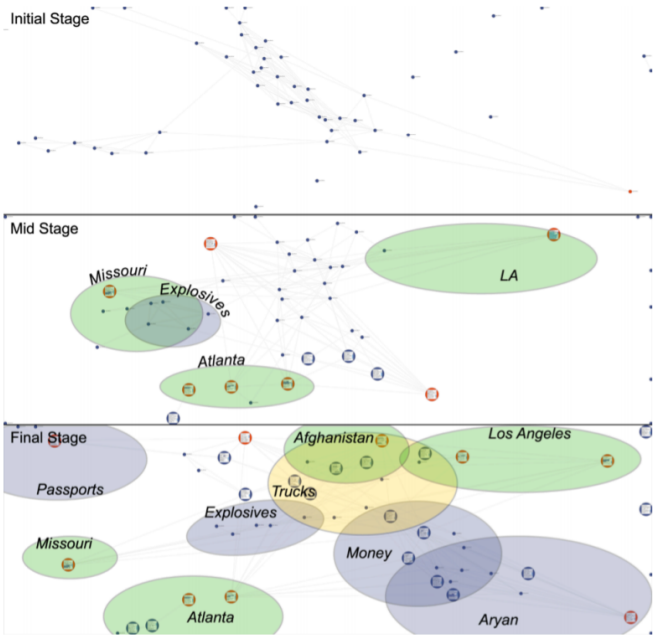


Fig. 5. Semantic graph for text documents. Up: Unprocessed data. Down: Final graph after processing user interactions [5]

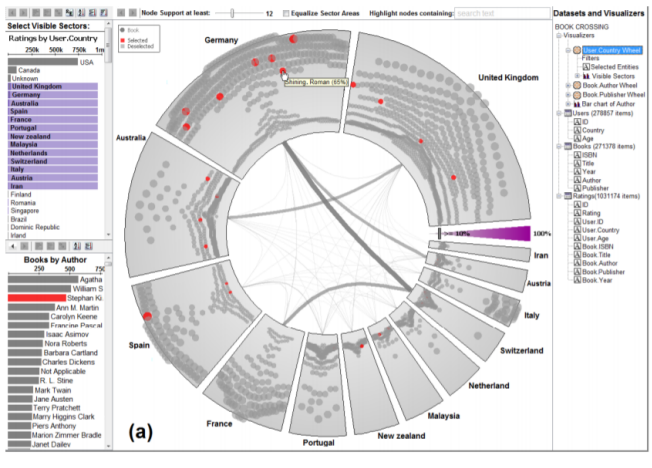


Fig. 6. Contingency wheel of books publications, from the top-left bar chart, the visible wheel sectors are selected. From the bottom-left bar chart, nodes that represent books by "Stephan King" are highlighted [1]

4.7 N-dimensional to three-dimensional

Many dimensionality reduction techniques can be used to map the N-dimensional data points to 3-dimensional data points, however, local affine multidimensional projection turned out to be the best option for this interactive technique [3].

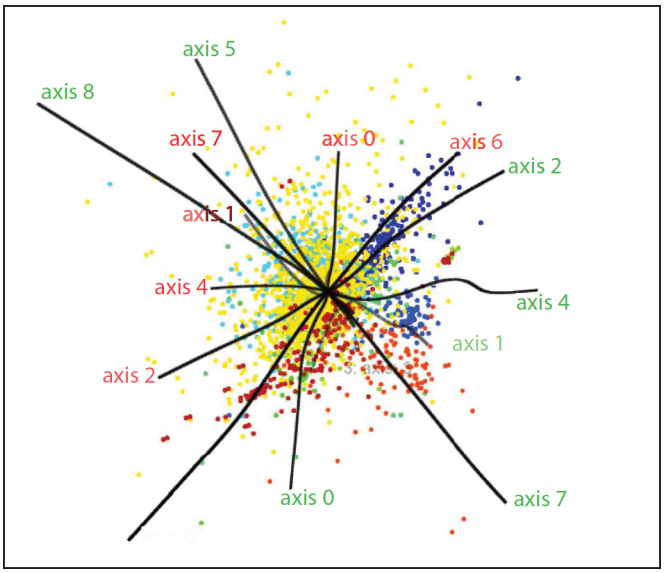


Fig. 7. Scatterplot with added biplot curves of a 9-dimensional dataset [3].

Once a DR technique has been applied to the dataset, representing this data on a screen can be done by using biplots and other variations of them. Biplots are similar to scatterplots but use three dimensions instead of only two, and easily reflect the relationships between observations and their values. Figure 7 shows a scatterplot with added curved biplot axes, which show the spread and non-linearity of the projection by the length and bends of the curves.

As stated before, 3D projections offer higher fidelity than 2D projections when it comes to data structure and may also be easier to interpret [3], given an optimal viewpoint from which to observe the data.

4.8 3D projection interaction

In most cases, there probably is more than only one optimal viewpoint, therefore an easy way to let the users visualize the data from the right viewpoints is to allow them to interact with the projection. Besides the assumed ability to rotate the plot using a track-pad or dragging the screen with a cursor, other techniques have been used.

One technique is adding axis legends, which work as indicators of the current axes being shown in the projection. Figure 8 shows how the y axis legend highlights variable 6 as it is the most relevant variable from this perspective and also how the x axis legend has variables 0,2 and 7 as the other relevant variables, meaning they are the best read variables from this position; the observability legend shows how least relevant (and visible) the variables are from the current viewpoint.

In the event that there are tens or hundreds of variables, only the 20 (or whichever value the users decide) more relevant variables from a certain perspective will be listed, as they will appear or disappear accordingly.

Another technique is to add a viewpoint legend, which is a sphere that can be clicked on and rotated, and shows how well can the variation between two variables be seen from the current viewpoint. Each variable pair receives a unique color, the brightness reflects if there is a viewpoint from which to examine a variable pair or if there is not. See Figure 9.

Additionally, a matrix is added next to the viewpoint legend sphere to help interpret it, each cell in the matrix corresponds to one variable pair. Clicking on the cell will rotate the sphere as well as the viewpoint and also update the axis legends.

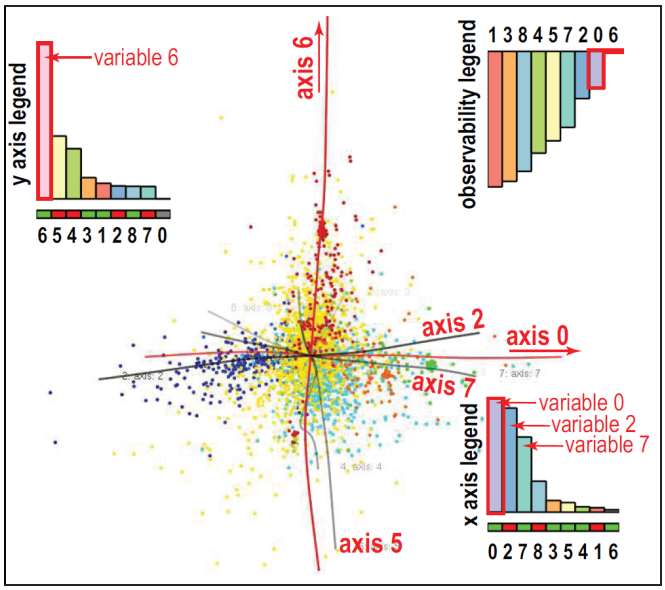


Fig. 8. Biplot with axis legends [3].

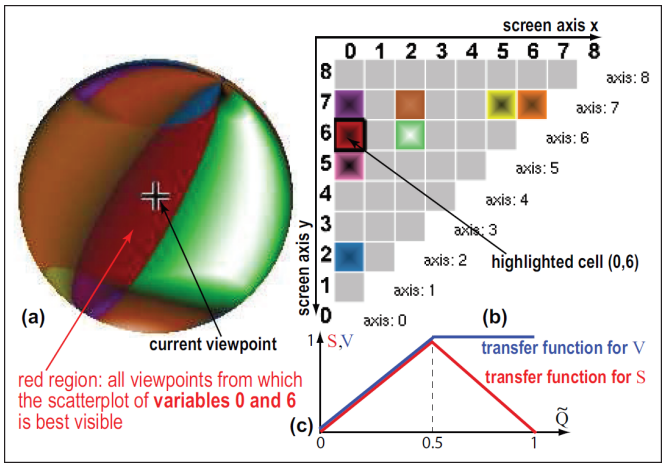


Fig. 9. Viewport legend [3].

Table 6. 3D projection interaction comparison.

Advantages
• Any non-linear dimensionality reduction technique can work directly without the need to modify said DR technique. • Simple to implement. • Scalable memory complexity, $O(N \cdot D)$.
Disadvantages
• Additional time to learn it (approx 20 min). • Viewpoint legend can be seen as complicated. • Large scatterplots can generate hard to disambiguate occlusion.

4.8.1 2D vs 3D

Thanks to the different perspectives that a 3D plot offers, it is easier to identify when a 2D plot is unintentionally hiding information. Such is the case of a dataset where three clusters exist but are only visible in a 3D projection since the 2D projection displays only two. Figure 10 shows this particular case as well as how the three-dimensional visualization technique looks with all the features in place.

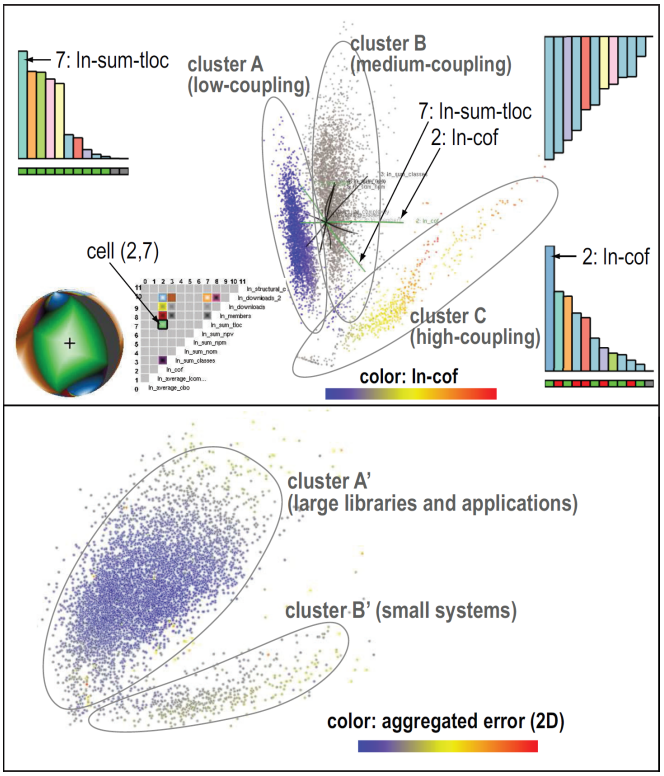


Fig. 10. 2D vs 3D clusterization of dataset [3].

The effectiveness of three-dimensional visualization techniques, however, depend on the quality of the dimensionality reduction projection. If there are missing patterns, complementary techniques are recommended. Furthermore, viewpoint and interaction heavily influence the usefulness of 3D projections.

With this in mind, a 3D projection can allow a user to examine the information from multiple viewpoints and then the user can decide if there is a single or multiple viewpoints that can best explore the projection.

4.9 Other tools

Other simpler but less powerful techniques can include tables (like a spreadsheet) where each observation is a row and each attribute is a column, a scatter plot matrix that shows the distribution of all observations between each pair of possible combinations of dimensions, parallel coordinates which shows attributes as parallel axis and observations as perpendicular fractured lines that cross all attribute coordinates, and many more. [7]

5 Summary

In this paper we looked at the general workflow that has to be performed in order to apply the desired tool. We need to understand if our data is numerical or categorical, most of the times is a mixture of both, in such case we can decide to convert the numerical dimensions into categorical with the use of bin ranges or convert the categorical into numerical by converting each category into a new attribute and then assigning a binary value depending if it appears or not. Based on that data we choose the most appropriate projection algorithm, in order to represent the maximum amount of information within a limited 2D or 3D tool. Then, depending on the kind of analysis we want

to perform, the implementation effort we are willing to invest and the user knowledge of these tools we can decide for a particular tool.

We also discussed the main algorithms employed to deal with high dimensional data in order to show the data into fewer dimensions (typically 2D). All algorithms aim to express the same data behaviour with less complex but more representative attributes. The most common ones are PCA and distance based metric.

Finally, we provided an example of each of these tools in order to explain what the plot means, how to interpret the results and what each extra visual aid element means.

6 Conclusion

All the tools use the same concept as scatter plots for showing an observation as a simple point in space, this allow them to scale well with the number of data elements, since they are only points in a 2D space.

Regarding the dimensionality of the data, as the number of dimensions increases,it gets more difficult to visualize explain the data, to distinguish similarities and dissimilarities between data points and it also results into difficulty to understand which dimensions are more relevant to the observations.

As observed during the explanation of each technique there is no definitive tool, every tool has its own advantages and disadvantages since all of them have a loss of data, this is unavoidable because we have limited resources (resolution, number of colors, size of visualization space, etc.).

There will always be two core problems, one is the implementation complexity of the tool and the other is the training required to properly understand the tool. As mentioned on sections 2 and 3, there is no out of the box tool that can be directly applied to any real world dataset, it is always necessary a brief pre-processing of the dataset to be given as input for a particular tool and even worse, it has to be specific for each tool, there is still not enough standardization. It is always necessary a certain degree of expert knowledge in order to decide which tool could best show the findings we are interested in. The more advanced the tool becomes, the more difficult it results for an untrained user to use and interpret the results. There is always a trade off between completeness, complexity and cost (in terms of effort).

We realize this is just the beginning of a new era of data analysis and we expect many more of these methods to appear, however the core principles will not differ much from what we discussed.

Acknowledgements

This work was supported in part by a grant from CONACYT / SICDET Michoacan, Mexico.

References

[1] B. Alsallakh1, E. Grller, S. Miksch, and M. Suntinger. Contingency wheel: Visual analysis of large contingency tables.

[2] B. Broeksema, T. Baudel, and A. Telea. Visual analysis of multidimensional categorical datasets, 2013.

[3] D. B. Coimbra, R. M. Martins, T. T. Neves, A. C. Telea, and F. V. Paulovich. Explaining three-dimensional dimensionality reduction plots.

[4] R. R. O. da Silva, P. E. Rauber, R. M. Martins, R. Minghim, and A. C. Telea. Attribute-based visual explanation of multidimensional projections.

[5] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics.

[6] S. Feiner and C. Beshers. Worlds within worlds - metaphors for exploring n-dimensional virtual worlds.

[7] P. E. R. Renato R.O. da Silva and A. C. Telea. Beyond the third dimension: Visualizing high-dimensional data with projections.

[8] T. Schreck, T. von Landesberger, and S. Bremm. Techniques for precision-based visual analysis of projected data.

[9] L. I. Smith. A tutorial on principal components analysis.

Vector graphics primitives: An overview of three techniques

Joël Grondman, Klaas Kliffen

Abstract—A raster image is represented by a grid of pixels which is limited by the resolution used during drawing. To solve this problem a different approach in image representation has been developed, called vector graphics. Several vector graphics primitives exist which can mostly be categorized under elemental gradients, gradient meshes and diffusion curves. Each primitive has its own strengths and weaknesses in terms of limitations and use of resources.
For each of these three primitives we discuss their way of representing a vector image. There is no objective "best" primitive, so we focus on what kind of image can be represented by each primitive more easily. Finally we compare each primitive method by its complexity, flexibility and resource usage in the process of creating an image.
Each primitive category has its own varying methods which are discussed and reviewed briefly, including examples. Primitives in different categories rely on different methods of rendering an image. Their methods are therefore not compared, but the effort required in using the methods as well as the resources needed are our main way of comparing different primitives. We found each category of primitives to have their own strengths and weaknesses which means none of them are redundant in general and their usefulness depends on the intended application.

Index Terms—Vector graphics primitives, gradient meshes, diffusion curves.

1 Introduction

Digital images are represented in two ways: raster images and vector images. Raster images consist of a grid of pixels. The grid can be seen as a canvas which can be modified by artist using different tools. These tools work in a similar way as physical painting tools, consisting of adding color by painting them. After painting it is hard to adjust the color and can only be changed by replacing the old color with a new one.

While raster images are limited by the resolution at which they are drawn, vector images can be scaled without aliasing artifacts such as losing sharp edges. However, vector images do need to be processed before they can be displayed on a screen or printed, since these consist of pixels or dots respectively. This process is called rasterization and converts the shapes described by the vector image to a raster image.

In vector graphics the artist has more control over elements, also called primitives, after painting. These primitives are mostly represented by Bézier curves representing either line segments, or shapes. Primitives can be easily changed after creating an image by adjusting the control points of the primitive.

Primitives can be colored using a single color or with a gradient. We focus mainly on gradients in this paper rather than single color shapes, because they allow for more expressive images. There are multiple definitions for a gradient which depends on the context. In [5] two definitions are given: Gradients are patterns of changing colors or a sequence of colors that is mapped to an image. The first definition is the result of the mapping and can be seen in the final image. The latter is represented by a number of so called stops which specify a color. Between the stops colors are interpolated. The interpolation can be linear, cubic or even a higher order function depending on the desired smoothness.

Several vector graphics primitives exist which can mostly be categorized under elemental gradients, gradient meshes [2] and diffusion curves [13]. For each of these three primitives we discuss their way of representing a vector image. There is no objective "best" primitive, so we focus on what kind of image can be represented by each primitive more easily. Finally we compare each primitive method by its complexity, flexibility and resource usage in the process of creating an image.

In Section 2 related work on comparing vector primitives is discussed. Then we discuss each of the compared techniques in more detail: elemental gradients in Section 3, gradient meshes in Section 4 and diffusion curves in Section 5. Afterwards we give a comparison of the techniques in Section 6. A summary and additional discussion is given in Section 7. Finally possible future research is discussed in Section 8.

- *Joël Grondman is a MSc. Computing Science student at the University of Groningen, E-mail: j.h.l.grondman@student.rug.nl.*
- *Klaas Kliffen is a MSc. Computing Science student at the University of Groningen, E-mail: k.y.kliffen@student.rug.nl.*

2 Related work

Comparing primitives is subjective and depends on the type of image that is drawn. Some objective measures are needed to compare vector graphics primitives. In [2] two goals for vector graphics primitives are mentioned: accurate control over the primitive and using as few primitives as possible for the final image.

Accurate control over the primitive can be described as the expressive power of the primitive. This can be seen as the number of controllable points for the primitive. The larger the number is, the more precise the control can be. Using as few primitives as possible decreases the complexity of an image. This benefits the computational time it takes to rasterize the vector image for displaying or printing. These two measures are related, since more accurate control structures allow for representing complex images with fewer primitives.

We use these two objective measures for a similar scene and also compare the resulting images of each of the techniques.

These primitives are closely related to vectorization methods, algorithms that convert an image into a gradient mesh or diffusion curve representation. While it is important to convert existing raster images we will focus on the creation of new images. Several other sources cover this topic in [13, 2].

3 Elemental gradients

In this section we discuss elemental gradients. These are the most basic primitives using gradients.

3.1 Basic method

Elemental gradients consist of a single shape which is filled with a single color gradient. The boundary of the elemental gradient is usually constructed with Bézier curve line segments. Two possible gradients exist to fill the shape: linear and radial.

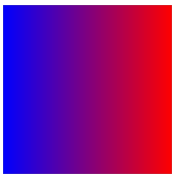


Fig. 1. Square with a linear gradient drawn in Inkscape.

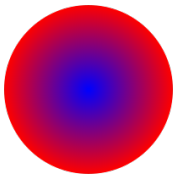


Fig. 2. Circle with a radial gradient drawn in Inkscape.

Fig. 3. Complex shape with a linear gradient with multiple stops drawn in Inkscape.

The linear gradient is controlled by a single straight line segment and shown in Fig. 1. Colors are interpolated along the line segment between stops. To calculate the color within the shape, points are projected onto the line segment. Points that cannot be orthogonally projected are projected to the line through the line segment and clamped to the color of the nearest end point of the line segment.

The radial gradient is controlled by a center point and one line segment representing the radius and can be seen in Fig. 2. Colors are interpolated along the line segment, where all points within the shape at the same radius have the same color. It is also possible to use two orthogonal lines to form an ellipse, instead of a circle. A variant of the radial gradient is the polar gradient, which uses a circle instead of line segment. Colors are then interpolated along the circle and all points on the line from the center to a point on the circle have the same color.

It is also possible to have multiple stops in a gradient. Such an example is shown in Fig. 3.

3.2 Benefits of elemental gradients

Elemental gradients are computationally efficient to calculate. Each primitive can be rasterized individually and pixels within the shape are colored using the projection onto the color gradient.

Elemental gradients can be used for simple images with shapes without small details. A boundary can be manually traced around an object in a reference image and a gradient can then be used to fill the shape. Elemental gradients are thus controlled by a boundary consisting of a number of Bézier curves and a single gradient with a fixed number of stops. This makes it very intuitive to draw with.

3.3 Weaknesses and limitations

For complex images, multiple elemental gradients are needed. Only a single gradient can be used per primitive. This allows only for smooth transitions of colors in the direction of the gradient. Sharp transitions between gradients can be worked around by reducing the color opacity or alpha value for colors near the end of the line segments of a gradient and using layering to stack multiple elemental gradients on top of each other.

4 Gradient meshes

In this section we discuss gradient meshes, which is a more advanced technique of using gradients in vector images.

4.1 Basic method

The gradient mesh technique uses a two-dimensional mesh of regular quads with sides represented by Bézier curves and interpolates colors between the vertices of a quad. The drawing process starts with the layout of the mesh over an shape to be drawn. Colors are then defined at the vertices of the mesh. The color of the pixels within a quad are then interpolated using a set of basis functions such as bi-linear or bi-cubic interpolation, depending on the smoothness desired.

4.2 Benefits of mesh gradients

Gradient meshes are just like elemental gradients computationally efficient to compute. Each quad can be rasterized individually and colors

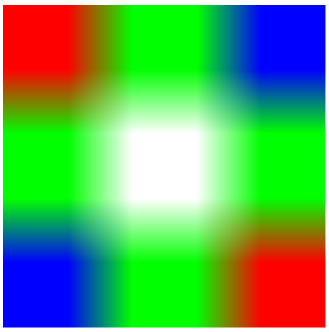


Fig. 4. A square filled with a simple gradient mesh drawn in Inkscape

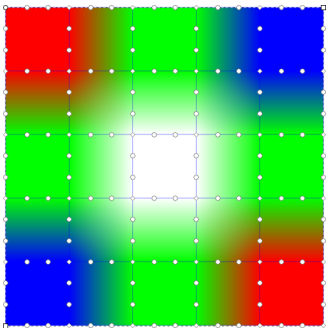


Fig. 5. Control points of the mesh from Fig. 4

for the pixels within the quad can be interpolated with the desired interpolation functions. A gradient mesh can be seen as a set of elemental gradients.

Gradient meshes are capable of having smooth transitions in more directions than elemental gradients. The colors are interpolated between four vertices instead of two points for the single gradient in the elemental gradients. Therefore gradient meshes are mostly used for smooth surfaces without hard edges.

Vertices can be placed and colored by hand, but it also possible to automate this process. When a raster image is used as a reference image, the colors of the vertices can be retrieved from the pixels of the raster image. The artists can create a course mesh over this image and mesh optimization [14] algorithms can be used to optimize the location of the vertices to match the original raster image as close as possible.

4.3 Weaknesses and limitations

While it is possible to create a gradient mesh from scratch, placement of the vertices is not very intuitive to get the desired result. Some editors, such as Inkscape [7], support gradient meshes only as a beta option and only allow to define the mesh at the start of the drawing and the mesh can not be changed later.

Some images require a large number of vertices to be represented. The number of vertices needed for the image depends on the number of transitions of the surface. This can be seen in Fig. 6, where regions containing sharp transitions beneath the nose are more dense compared to smooth regions in the skin. Since the mesh is regular, some smooth regions are also modeled by this part of the dense grid. This problem can be solved by using a course mesh for the smooth regions and create a new layer on top with a finer mesh to support the detailed regions in an image.

Another weakness of gradient meshes is its ability to represent

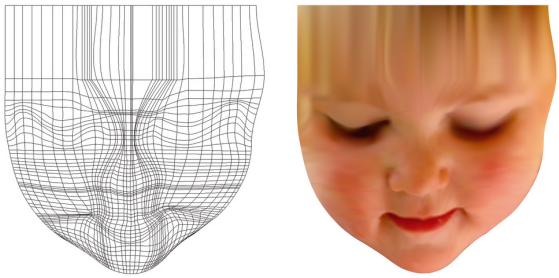


Fig. 6. A human face modeled with a gradient mesh. Regions with high detail need a lot of vertices to be represented. Image by roctopus: `https://www.flickr.com/photos/hownowbrowncow/2667521674`.

sharp edges in an image. Since colors are interpolated between vertices, this is only possible by putting two vertices on top of each-other. A more common approach is to use multiple gradient meshes and layering them on top of each-other.

5 Diffusion curves

The edges in an image are important features of an image. Diffusion curves mimick the properties of edges in images.

Diffusion curves (DC's) are curves which impose constraints on the image such as color. As in reality, edges often represent sharp color transitions which diffusion curves represent. From these edges a diffusion process fills in the rest of the space. Two implementations of DC's have been made which we describe shortly and compare.

Orzan et al. in [13] implemented such a DC method where the user can draw curves with a gradient on either side of the curve. Given these curves, a fitting image that abides by these constraints is computed through solving the Laplacian equation over the whole image. This solution yields a smooth color transition throughout the image except at the drawn DC's.

Finch et al. in [3] solved a higher order equation instead. This leads to a different kind of smoothness and generalizes the type of constraint a DC can represent, not necessarily two colors on either side but also color derivative constraints. This higher order approach leads to more tools and flexibility in exchange for performance.

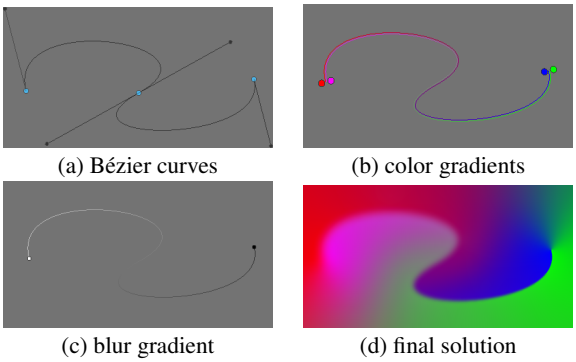
5.1 Laplacian solution

In this case a vector image is represented by a set of DC's. Each DC is a curve with two sides. Each side of the spline has its own color gradient. In addition blur control points Σ can be added that control the smoothness of color transition across the curve. This whole process is depicted in Fig. 7.

After the DC's have been drawn the image I is computed by solving the Laplacian equation $\Delta I = 0$ except near the DC's themselves to maintain sharp color transitions near the curves. The sharp color transition on the curves is smoothed out afterwards by adding blur. Fig. 7d shows how the result could look like.

Solving the Laplacian equation requires solving a large, sparse linear system which is efficiently solved by a GPU implementation of the multigrid solver [4]. The process is similar to heat diffusion, several iterations are done to propagate the color from the curves and spread them through the image.

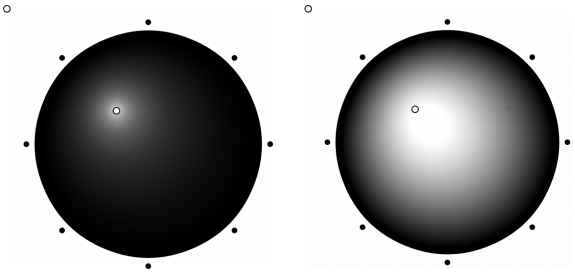
This is a very brief explanation of the DC solution offered by Orzan et al. which suffices for this research. For details on the implementation we refer to [13].



(a) Bézier curves (b) color gradients

(c) blur gradient (d) final solution

Fig. 7. The process of creating a DC. (a) Defining the curve, in this case a Bézier curve, (b) define color gradients on either side of the curve, (c) add optional blur along the curve, (d) final solution through solving a Poisson equation and applying the blur gradient afterwards. These figures were made using an editor by Orzan et al.[13].



(a) Laplacian solution (b) BiLaplacian solution

Fig. 8. Two circles drawn using different DC solutions. The BiLaplacian solution (b) interpolates smoothly but the Laplacian solution (a) has a discontinuity near the color constraints. These figures were made using an editor by Finch et al. [3].

5.2 BiLaplacian solution

This approach is similar to the Laplacian solution except a higher order equation is solved, the BiLaplacian equation $\Delta^2 I$.

Similarly to the previous method, the vector image is represented by curves. These curves have been generalized in the sense that these curves can not only contain a color constraint but also a derivative constraint due to the higher order equation to be solved. Note that this was not possible in the previous solution due to the lower order Laplacian equation.

The curves in this case introduce a weight around them which affects the final solution. Depending on the type of constraint a weight can be added to fix a certain color near the curve or a weight along the derivative tangent to the curve to achieve a contour along the curve. These weights are added resulting in a large sparse linear system which is optimized using least-squares resulting in the best fitting image solution given the constraints.

As said before the constraints imposed by DC's are not solely color constraints with this method. Several other constraints are possible that go by different names:

Tears remove the BiLaplacian constraint near the curve leading to a sharp color discontinuity. Creases do the same but add derivative constraints in the normal direction of the curve to not lose color discontinuity. Slopes add derivative constraints in the normal direction of the curve. Contours add constraints in the tangent direction of the curve. These constraints can be combined as well to create compound curves making this method a powerful tool to manipulate color and derivative continuity throughout the vector image.

For details on the linear system to be solved and more we refer to Finch et al. in [3].

5.3 Both solutions compared

Both solutions achieve smoothness throughout the vector image except where DC's are defined. The difference in smoothness is noticable, however, as can be seen in Fig. 8. As can be seen the Laplacian solution leads to a smooth color transition but has a tent-like behaviour near the constrained point in the center of the circle. The higher order BiLaplacian solution provides a better, smoother solution without tent-like artifacts.

In addition the Laplacian solution can not have derivative constraints because it already seeks zero derivative in all directions. With the BiLaplacian solution you can have these additional derivative constraints which leads to contour, slope, crease and directional constraints. These additional types of constraints lead to more tools for the artist which makes the BiLaplacian solution a more flexible DC tool.

In terms of performance, both methods suffer from the fact that each pixel in the image is not locally defined by its nearest primitives, e.g. in gradient meshes each pixel is contained in one of the many quads and thus defined by the interpolation of the colors of the four vertices. With DC's the pixel color value depends on all the defined constrains

in the vector image. This implies that DC's need more resources in order to draw the vector image compared to gradient meshes. One consequence of this is that performance is linked to the resolution of the vector image as the size of the linear system that needs to be solved grows when the resolution is increased. Both DC methods apply several performance optimizations in order to make the methods interactive. Still some latency can be noticed when using these tools.

Comparing both the Laplacian solution and BiLaplacian solution, the BiLaplacian solution is superior due to not suffering from tent-like effects and has more constraint possibilities aside from color constraints. This method however is more resource intensive. As said before the Laplacian solution uses the GPU implementation of the multigrid solver which the BiLaplacian solution can not use due to the complexity of its linear system.

5.4 Other solutions

There are more DC schemes which provide additional tools to the Laplacian solution. In this section we discuss three of them.

Poisson vector graphics [6] is an addition to the Laplacian solution. It allows the user to create Poisson regions and curves in order to create specular highlights and shadows.

The generalized diffusion curves [8] method generates two images using two Laplacian solutions, these two laplacian solutions are obtained by having two color gradients constraints on each side of a DC. A blending function then spatially blends the two solutions together to form a new image. This leads to a more natural color transition and removal of tent-like effects near constraints.

Shading curves [11] follows a different procedure than the other methods. First each region with different color tone is outlined and then filled with a constant color. The user specifies shading curves along edges which propagate through the image creating shadows or specular highlights.

6 Comparison

In this section we compare the methods and discuss possible applications. For our first comparison we use a reference image of an apple as can be seen in Fig. 9. The apple primarily features a smooth surface. We created a second scene featuring a cube with a shadow with both sharp and smooth edges. Finally we summarised the main results in Table 1.

6.1 Elemental gradients

Elemental gradients are the most basic type of primitive. Regions can be of any shape or size and filled in with only one color gradient. This constraint leads to its inability to define more complex color transitions in one region. To achieve a more complex color transition multiple regions need to be defined each with a different color gradient.

Since each region's color is only defined by its own gradient the color transition between regions is not automatically smooth and hence smoothness needs to be artificially created. This becomes unmanageable when adjusting a region's shape, size or color gradient leading to multiple edits to other regions to maintain smoothness. Elemental gradients should therefore only be used for creating images with simple one-dimensional color gradients.

The two other primitives solve the smoothness across regions problem by having interdependence between regions e.g. shared nodes between adjacent regions for gradient meshes and not being constrained to (semi-)closed regions at all for diffusion curves.

The apple example in Fig. 10 clearly shows the hard edges between the different elements. We tried to keep the number of shapes low, therefore the texture of the surface of the apple is lost.

Elemental gradients perform better on the cube scene in Fig. 14. The sharp edges between the different surfaces are easy to recreate with a single quad. For the shadow we used a single shape with a radial gradient, however the shadow might be improved by using a linear gradient with multiple stops.

Fig. 9. Reference image of an apple. Cut-out from the original image: `https://www.pexels.com/photo/apple-fruit-healthy-8208/` released under a CC0 license.

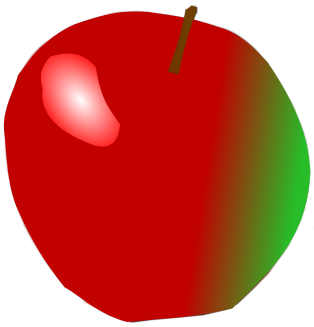


Fig. 10. Apple drawn with 3 elemental gradients in Inkscape.

Fig. 11. Apple drawn with 2 gradient meshes in Inkscape.

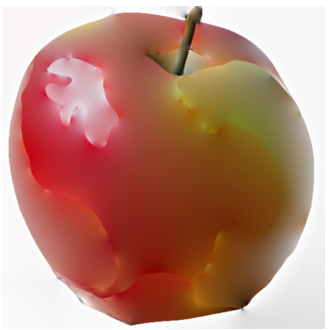


Fig. 12. Apple drawn with the Laplacian solution. Using the tool provided by Orzan et al. [13].

Fig. 13. Apple drawn with the BiLaplacian solution. Using the tool provided by Finch et al. [3].

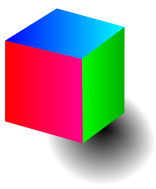


Fig. 14. Cube drawn with 4 elemental gradients in Inkscape.

Fig. 15. Cube drawn with 4 gradient meshes in Inkscape.

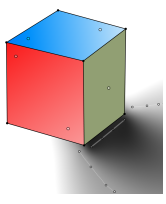


Fig. 16. Cube drawn with the Laplacian solution. Using the tool provided by Finch et al. [3].

Measure	Elemental gradients	Gradient meshes	Diffusion curves
Ease of drawing	Very easy to draw simple scene	Easy for course mesh, needs training for complex mesh	Very expressive primitive, needs training before use
Number of primitives	Large for complex scenes	Smaller than elemental gradients, more vertices than diffusion curves	Least amount of primitives
Editors available	Available in every vector image editor	Available in most editors, sometimes as a beta feature	Only in research applications
Flaws	Too simple for drawing complex scenes, no automatic smoothness between regions	Dense mesh needed for complex surfaces	Performance issue especially when used interactively
Other remarks			Additional application in rendering surface details or key framing

Table 1. Comparison of the three methods based on various measures.

6.2 Gradient meshes

Gradient meshes perform well on smooth surfaces, but require careful placement of the vertices at the right location. The colors of the vertices for the apple in Fig. 11 were automatically sampled from the reference image after we manually placed the vertices. For the stem a second mesh was used, placed on top of the first mesh.

The smooth surface transitions can be modeled quite well with the gradient mesh, however some aliasing from placing the vertices at suboptimal locations can be seen in the highlight of the apple.

Gradient meshes do not perform well on sharp edges and this can be seen in Fig. 15. We used 3 gradient meshes for representing the cube, essentially mimicking elemental gradients. A fourth gradient mesh was used for the shadow. The placement and coloring of the vertices was not intuitive and as such the shadow does not look as well as it could.

6.3 Diffusion curves

Compared to the other methods diffusion curves are not bound to closed regions. The user does not have to define closed regions and can define multiple color constraints, or other constraints such as contours, inside a region. This makes DC's the most flexible vector graphic primitive. This strength comes with a weakness, a small change may have an unforeseen side-effect in the whole image.

Due to the flexibility the interpolation technique is also more complex. Where the other two primitives consist of closed regions with a well-defined color interpolation process, DC's have no such thing and the whole vector image color propagation between curves is computed in one go by solving a large linear system. Several optimizations have been applied to this scheme to maintain user interactivity. One of these optimizations is solving a more coarse version of the image first before refining it which is noticeable while editing curves.

Some expertise areas may benefit from diffusion curves. Key framing in animation or film making is used to create in-between images given two-frames which yields in smoother animations. Due to the low amount of primitives in DC images, key framing can be applied effectively to create a smooth animation. In theory this can be done using gradient meshes as well but due to the higher amount of primitives requires more time.

Another application is rendering surface details with diffusion curves [9]. Traditionally textures are stored at a given resolution which results in staircase effects or blurry edges. DC's offer a way to store edge details that can be used for any resolution.

In Fig. 12 and Fig. 13 a vector image representing an apple was drawn by the Laplacian and BiLaplacian solution respectively. Note that the program used to draw Fig. 12 had tools to match drawn curves to contours and sample colors from an underlying bitmap image which we used. This resulted in a better approximation to the raster image in Fig. 9 but due to the curves not matching the contours of underlying mipmap perfectly caused some color being "leaked" from some regions closed by DC's.

Fig. 16 shows a better application of DC's. While the cube itself could be just as efficiently drawn with elemental gradients, it only took a few primitives to mimic a color gradient using DC's. The shadow was drawn more effectively using this method, a source of the shadow and two value, crease compound curves were used to guide the shadow on either side.

7 Discussion

In this paper we have looked at three types of vector graphics primitives with as main objective to compare them to each other. We have shortly described each primitive and can draw some general conclusions based on their implementation.

Elemental gradients are easy to use but as a result lack the needed complexity to draw more interesting color transitions. Gradient meshes allow more complex color transitions due to smooth color transition between quads but due to the regular mesh may result in dense meshes to represent a vector image. Diffusion curves are the most complex primitive with a wild range of different implementations available. Due to its non-closed form and intuitive representation, requires the least amount of primitives to draw a complex image. However, as a cost requires a lot of resources to use and its color propagation scheme is less controllable compared to the other primitives.

As an example we have attempted to redraw an apple given a raster image example. The apple contained a lot of texture which is an issue for all the discussed primitives. The thesis by Orzan [12] describes the process of applying texture to gradient meshes or DC's. We've chosen not to apply any texture and to go with a more artistic drawing of an apple. Given a small amount of time and artistic talent, DC's proved to be the most efficient method of drawing an image with reasonable complexity.

In the introduction we stated that each primitive has its own application. This is true as we have shown that simple image such as the cube was best drawn with elemental gradients followed by DC's. For the apple gradient meshes followed by DC's was the best method to be used. While these were only two examples, more applications exist. Elemental gradients are often used to create logos for companies and gradient meshes for more natural objects such as faces or fruit. Diffusion curves are not widely used yet, as the most professional editing software such as Adobe Illustrator and Inkscape have not adopted them yet. DC's, however, could adapt to both applications of elemental gradients and gradient meshes as DC's can create complex and simple color transitions at the same time without a great amount of redundant primitives to be drawn.

8 Future research

There are still many open problems concerning vector graphic primitives. In this section we discuss a few of them we encountered.

Gradient meshes provide a way of dividing an image into regions and smoothly interpolate in between them. But due to the regular mesh structure of gradient meshes, increasing the resolution of a quad

locally is currently not possible. Subdividing a quad may provide a solution at the cost of increased overhead [10].

Diffusion curves are a more recent area of research, the first method was defined in 2008 by Orzan et al. and republished in [13]. Several adaptations were created based on this basic DC method such as shadow curves and the higher order approach based on the BiLaplacian solution. It is likely that more approaches will arise in the near future.

Currently elemental gradients and gradient meshes have been adapted into vector drawing applications such as Adobe Illustrator[1] and Inkscape while no such professional application exists yet for DC's. This is most likely due to DC's requiring more resources which impedes interactivity and all the different kinds of DC's that currently exist. A closed-form method that is efficient to render combined with the ability to define gradient extrema constraints may provide the performance and flexibility that is needed. As mentioned by Barla et al.in [2]: A closed-form solution combined with gradient extrema primitives remains a challenging research direction.

Acknowledgments

The authors would like to thank J. Kosinka for reviewing and advising on the subject.

References

[1] Adobe. Adobe illustrator: commercial vector graphics editor. `http://www.adobe.com/products/illustrator.html`. Accessed: 2017-3-29.

[2] P. Barla and A. Bousseau. *Gradient Art: Creation and Vectorization, in Image and Video-Based Artistic Stylisation (P. Rosin and J. Collomosse Eds.)*, chapter 8, pages 149–166. Springer, 2012. Edited by Paul Rosin and John Colomosse.

[3] M. Finch, J. Snyder, and H. Hoppe. Freeform vector graphics with controlled thin-plate splines. *ACM Trans. Graph.*, 30(6):166:1–166:10, Dec. 2011.

[4] N. Goodnight, C. Woolley, G. Lewin, D. Luebke, and G. Humphreys. A multigrid solver for boundary value problems using programmable graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, HWWS '03, pages 102–111, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[5] J. v. d. Gronde. Seperating gradients from geometry. In *9th International Conference on Scalable Vector Graphics*, 2011.

[6] F. Hou, Q. Sun, Z. Fang, Y. Liu, S. Hu, H. Qin, A. Hao, and Y. He. Poisson vector graphics (PVG) and its closed-form solver. *CoRR*, abs/1701.04303, 2017.

[7] Inkscape. Inkscape: Open source scalable vector graphics editor. `https://inkscape.org`. Accessed: 2017-3-29.

[8] S. Jeschke. Generalized diffusion curves: An improved vector representation for smooth-shaded images. *Computer Graphics Forum*, 35(2):71–79, 2016.

[9] S. Jeschke, D. Cline, and P. Wonka. Rendering surface details with diffusion curves. *ACM Trans. Graph.*, 28(5):117:1–117:8, Dec. 2009.

[10] H. Lieng, J. Kosinka, J. Shen, and N. A. Dodgson. A colour interpolation scheme for topologically unrestricted gradient meshes. *Computer Graphics Forum*, pages n/a–n/a, 2016.

[11] H. Lieng, F. Tasse, J. Kosinka, and N. A. Dodgson. Shading curves: Vector-based drawing with explicit gradient control. *Computer Graphics Forum*, 34(6):228–239, 2015.

[12] A. Orzan. *Contour-based Images: Representation, Creation and Manipulation*. PhD thesis, INPG, june 2009.

[13] A. Orzan, A. Bousseau, P. Barla, H. Winnemöller, J. Thollot, and D. Salesin. Diffusion curves: A vector representation for smooth-shaded images. *Commun. ACM*, 56(7):101–108, July 2013.

[14] J. Sun, L. Liang, F. Wen, and H.-Y. Shum. Image vectorization using optimized gradient meshes. *ACM Trans. Graph.*, 26(3), July 2007.

Stochastic Gradient Optimization With Loss Function Feedback

Jos van de Wolfshaar & Siebert Looije

Abstract—Recently, a large share of the machine learning community has shifted its focus towards deep learning. This paradigm shift is due to significant successes in a wide range of sub-domains such as image classification, face verification and speech recognition, often outperforming humans at the task they are trained for. The introduction of new optimization algorithms for gradient descent have boosted the scientific progress. Gradient descent algorithms seek to minimize a cost function that formally describes the penalty given a model's prediction and a target. Until recently, the most popular form of gradient descent has been stochastic gradient descent (SGD). However, in the past few years, several extensions of vanilla SGD introduced exhibiting superior convergence guarantees and speeds. One particularly successful extension is named Adam. This algorithm adaptively tunes the learning rate, yielding state-of-the-art performance. More recently, a novel optimizer called Eve has been introduced that is shown to outperform Adam for image classification and language modeling. The Eve optimizer extends on Adam by adopting loss function feedback to scale the learning rate. For this paper, we have tried to validate the improvement of Eve over Adam in a range of experiments that test the applicability outside the domain of the experiments that the authors of the Eve algorithm assessed. By comparing the optimizers in this unexplored area of problems, we show that the Eve optimizer has some instability issues that can be overcome by proposing a simple clipping procedure. In addition, we investigate whether the use of feedback can be applied to the RMSProp gradient descent optimizer as well.

Index Terms—Gradient descent, stochastic optimization, adaptive moment estimation, loss function feedback.

1 Introduction

In recent years, gradient descent algorithms have become increasingly more popular. These algorithms can be used to optimize any differentiable loss function. Nowadays, most of such differentiable loss functions are those describing the error of machine learning models. The impressive results that have been obtained for challenges ranging from computer vision [13], language modeling [26], speech recognition [22, 1] and more have boosted the scientific efforts towards improving the standard gradient descent algorithms. With the introduction of several new optimizer algorithms since, these models are trained significantly faster than before.

The fact that deep neural networks (DNNs) can deal with real-life data and that they often outperform humans at the task they are trained for makes them a popular choice for a wide range of machine learning problems. Examples of applications are: face recognition that can be used for security [9], descriptive caption generation for images [24, 4] and leveraging the accuracy of cancer diagnostics systems [6].

In practice, the training time required for DNNs and other shallow architectures might impede the actual employment given limited time and computational resources. To that end, it is of vital importance to enhance the data efficiency of these models. In this text, data efficiency refers to the test error with respect to the amount of training examples that are used. In other words, the better the data efficiency, the less training iterations we need to converge to a proper solution of the optimization problem.

This paper elaborates on the performance of two state-of-the-art optimizers. The first optimizer is known as the adaptive moment estimation optimizer (Adam) [10]. This algorithm was introduced prior to the second optimizer that we consider, which is the Eve optimizer [11]. Both algorithms adaptively tune the learning rate, making them robust to different learning rate initializations, ultimately leading to better performance. The Eve optimizer extends on Adam by also including a loss function feedback coefficient that speeds up learning if the relative change in the loss is small and vice versa. Koushik et al. [11] show that the Eve optimizer outperforms the Adam optimizer in terms of data efficiency for several challenges such as image classification and language modeling. For further investigation of the possible limitations of both algorithms with respect to another, we explore their performance and limitations in several additional experiments.

We quickly noticed the danger of numerical instabilities for the default settings of the Eve optimizer in some experimental situations. For this reason, we explore how we can assure stability of the optimizer. Our results show that the proposed measure of ensuring stability yields a more robust optimizer.

In addition, we explore the use of feedback for the RMSProp optimizer [23]. Similar to Adam and Eve, this algorithm adaptively tunes its learning rate, albeit in a slightly different way. In our experiments we show that using feedback for the RMSProp might not necessarily yield an improvement over the standard RMSProp implementation.

Outline In Section 2 we elaborate on related research in the field of stochastic optimization through gradient descent. Section 3 discusses the details of the algorithms and models that were used. Then, we cover the data and experiment setup in Section 4 and the corresponding outcomes in Section 5. Finally, Section 6 reflects on our findings and proposes future directions for research.

2 Background

This section elaborates on related work regarding gradient descent algorithms that are relevant for our methods and experiments. We describe several gradient descent optimizers in chronological order. For a more thorough treatment of these algorithms, see the overview in [19].

2.1 Gradient descent

Gradient descent was first proposed by Cauchy [3]. This method was introduced as a generic function minimizer. In machine learning, the function to minimize is often referred to as the *loss function* or *cost function*. In this text, loss functions are denoted as $\mathscr{L}(\vec{x}, y; \theta)$. The semicolon is to emphasize the fact that the role of $\vec{x}$ and y are conceptually different from the role of θ. The vector $\vec{x}$ denotes the model's input and y denotes the models *target* (i.e. the desired output). These are not to be altered by the algorithm and are externally provided. For example, in image classification, $\vec{x}$ will usually be the image, and y may take the form of an image label such as 'cat' or 'dog'. The function should be minimized with respect to θ. To accomplish this, gradient decent methods consider the gradient of the function to find the local direction of steepest descent in the parameter space given by θ.

- *Jos van de Wolfshaar is a MSc. student at the University of Groningen, E-mail: j.van.de.wolfshaar@student.rug.nl.*
- *Siebert Looije is a MSc. student at the University of Groningen, E-mail: s.looije@student.rug.nl.*

This boils down to iteratively updating θ as follows:

$$\theta \leftarrow \theta - \eta g_t, \tag{1}$$

where

$$g_t = \nabla_\theta \mathscr{L}(\vec{x}, y; \theta), \tag{2}$$

and $\eta \in (0,1)$ is the *learning rate* which characterizes the magnitude of the updates with respect to the gradients.

The loss function $\mathscr{L}(\vec{x}, y; \theta)$ should characterize the error of the model with respect to the task it is trained for. For the sake of simplicity, we restrict ourselves to the case of regression, where the loss function is usually of the form:

$$\mathscr{L} = \frac{1}{2}\sum_i^N (f(\vec{x}^{(i)}; \theta) - y^{(i)})^2, \tag{3}$$

where N is the number of examples in the data set and $f(\vec{x}; \theta)$ is the model's *prediction* and $\frac{1}{2}$ is added for mathematical convenience. Evaluating this term repetitively can become computationally expensive in case of large data sets. Moreover, minimizing this term for a train set will not guarantee adequate performance on some unseen *test set*. Ultimately, the model should be able to generalize over unseen data. If one naively uses the exact gradient given by Equation (3), it might lead to an *overfitted* model. Overfitting refers to the problem of minimizing the error on train data to such extent that we overcompensate for noisy patterns that are not part of the underlying distribution that has generated the data. This leads to impeded performance on unseen data, which is typically contained in a *test set*. Moreover, following the local gradient so accurately, will potentially steer the weights θ into zero-gradient regions that are suboptimal such as saddle points, local minima and plateaus.

2.2 Stochastic gradient descent

Stochastic gradient decent (SGD) [2] was introduced to partially overcome the problems that were introduced near the end of the previous section. SGD *approximates* the error gradient by only considering a subset of the training data:

$$\mathscr{L} = \frac{1}{2}\sum_i^M (f(x^{(i)}; \theta) - y^{(i)})^2, \tag{4}$$

where $M < N$. Originally, the case in which $M = 1$ was referred to as SGD. Nowadays, when $1 < M < N$, it is common to refer to the method as being stochastic batch gradient descent or just SGD. The method is stochastic in the sense that the error gradient is approximated instead of being fully evaluated. By doing so, the algorithm no longer follows the exact shape of the error surface. This can lower the probability of ending up in zero-gradient regions and it also reduces the chance of overfitting. Furthermore it is important to mention that the examples are randomly selected. Moreover, the method is significantly more efficient, as we only need to evaluate a subset of the data for each update of θ.

2.3 Momentum

Usually, there are areas in the parameter space of the loss function that are substantially steeper in some directions than in other directions [21]. This causes disproportionate updates in which certain dimensions dominate the parameter updates. This can lead to oscillating trajectories in the parameter space and impeded performance. To overcome this issue, Qian et al. introduced the momentum method [18]. This method makes SGD more robust for such disproportinate dimensions by smoothing the trajectory, effectively dampening the oscillations. This is accomplished by adding some fraction of the previous update step into the current update step. Equations (5) and (6) display the gradient descent update rule with the added momentum:

$$\theta \leftarrow \theta - p_t, \tag{5}$$

where

$$p_t = \gamma p_{t-1} + n g_t, \tag{6}$$

and γ is the fraction of the previous update step and the p_t is the update step.

2.4 Nesterov Accelerated Gradient

One potential risk of using the momentum term is that the added momentum might become too large, such that it does not have time to slow down properly in the vicinity of a minimum. As a result, the optimizer might overshoot a proper solution. To avoid this issue, the Nesterov accelerated gradient (NAG) was proposed in [15]. In NAG, an approximation of the values of the gradient in the next iteration is calculated first. This is accomplished by using the momentum to determine the current direction, which means that we perform an update step before even evaluating a gradient. Then, by evaluating the gradient at this new point we fine-tune the direction by using the gradient of our approximated future position, instead of our actual position at the beginning of the update iteration. A mathematical description of the NAG is shown in Equation (7). A fraction γ of the previous update step p_{t-1} is subtracted from the input of the objective function:

$$g_t = \nabla_\theta \mathscr{L}(x, y; \theta - \gamma p_{t-1}). \tag{7}$$

2.5 AdaGrad

AdaGrad was among the first few gradient-based optimization algorithms that use adaptive learning rates. It was introduced in 2011 by Duchi et al. [7]. Adopting adaptive learning rate implies that the learning rate is updated according to the development of the gradients through time. If the parameters update are infrequent and/or small, then the adapted learning rate increases, otherwise the adaptive learning rate decreases. The history of gradients is used to get a more reliable estimate of whether the gradients have been small or large. Formally, the AdaGrad update rule is given by:

$$\theta \leftarrow \theta - \frac{\eta}{\sqrt{G_t + \varepsilon}} g_t, \tag{8}$$

in which G_t is the sum of squares of the past gradients for all parameters separately. This G_t is used to adapt the learning rate as shown in the update step of the AdaGrad in Equation (8). The ε is the so-called *fuzz factor* that is usually set a small non-zero constant, e.g. 0.0001, which guarantees that the denominator is not going to approach 0, thus eliminating numerical instability issues. The main drawback of using the AdaGrad method is that the G_t term always increases due to fact that it is an accumulation of squared values. Eventually, this might severely impede the learning speed of the optimizer.

3 Methods

This section elaborates on the algorithms that we used for our experiments. In particular, we look at several adaptive optimizers and we provide a detailed specification of the algorithmic steps.

3.1 Adam optimizer

The Adaptive Moment Estimation optimizer, or simply Adam, was recently introduced by Kingma et al. [10]. This optimizer computes the adaptive learning rate in a similar fashion as the AdaGrad [25] or the RMSprop [23] methods. The RMSprop and AdaGrad algorithms use an exponentially decaying average of the past squared gradient v_t for calculating this learning rate. The Adam optimizer extends on this idea by also incorporating an exponentially decaying average of the past gradient m_t.

The running average of the first-order moment m_t is calculated as:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{9}$$

where β_1 is the decay rate which is usually set to 0.9 [10] and g_t is the computed gradient at iteration t. So according to the average of the past gradient will mostly be influenced by m_{t-1}.

The running average of the second-order moment v_t is calculated by:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2 g_t^2) \tag{10}$$

The decay rate β_2 is set to 0.999, based on the general advice in [10]. This means that the trajectory of the second-order moment has a greater time-window when compared to the first-order moment.

The Adam update θ_{t+1} is calculated by :

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon}\hat{m}_t \tag{11}$$

with

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \tag{12}$$

and

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}, \tag{13}$$

where we also encounter a fuzz factor ε that is usually set to 10^{-8} [10].

3.2 Eve optimizer

Roughly two years after the introduction of Adam, Koushik et al. proposed the Eve optimizer [11]. They suggest to also adapt the learning rate by using the loss function's history. They propose to divide the learning rate by the relative change of the loss function. The loss function at time t will be denoted f_t from hereon. Their motivation is that if the relative change of the objective function is large, then the learning rate should be reduced and if the objective function is small, then the learning rate should be increased.

It is important to stress that the Eve optimizer is an extension of Adam in the sense that it merely adds a loss feedback coefficient d. So just like the Adam optimizer, the Eve optimizer makes use of running averages of the first and second order moment of the gradient. It then computes a lower bound δ_t and an upper bound Δ_t as follows. If we assume that the objective function at time $t-1$ is smaller than or equal to the tracked objective function at time $t-2$, $f_{t-1} \leq \hat{f}_{t-2}$ then

$$\delta_t = \frac{1}{k+1},\ \Delta_t = \frac{1}{K+1} \tag{14}$$

otherwise (so if $f_{t-1} > \hat{f}_{t-2}$):

$$\delta_t = k+1,\ \Delta_t = K+1, \tag{15}$$

where k is set to 0.1 and K is set to 10 by default.

The next step is to determine the *change factor* c_t. This change factor is clipped using δ_t and Δ_t as follows:

$$c_t = \min\left(\max\left(\delta_t, \frac{f_{t-1}}{f_{t-2}}\right), \Delta_t\right). \tag{16}$$

This clipped change factor is used to determine an *estimated loss* at time $t-1$:

$$\hat{f}_{t-1} = c_t \hat{f}_{t-2}. \tag{17}$$

The algorithm then determines a certain ratio between the minimum of both the new estimated loss and the previously estimated loss with respect to the absolute change between them:

$$r_t = \frac{|\hat{f}_{t-1} - \hat{f}_{t-2}|}{\min\{\hat{f}_{t-1}, \hat{f}_{t-2}\}}. \tag{18}$$

It then uses the running average of this ratio as the feedback coefficient:

$$d_t = \beta_3 d_{t-1} + (1-\beta_3) r_t \tag{19}$$

where $\beta_3 \in [0,1]$. By default Koushik et al. set the value of β_3 to 0.999, which means that time window of d_t is relatively large [11]. After d_t is calculated as in Equation (19), it is used in the learning rate by dividing the learning rate by the value of d_t. The parameter update for the Eve optimizer is shown in Equation (20). It is clear that it only differs slightly from Adam's update rule, as the only addition is the feedback coefficient d_t:

$$\theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{d_t \sqrt{\hat{v}_t} + \varepsilon}. \tag{20}$$

We chose to separate the loss function feedback mechanism and the default Adam code into two algorithms that are listed in Algorithm 1 and 2. Adding the feedback coefficient to the update rule results in larger updates when the change in loss is relatively small and vice versa.

Algorithm 1 UpdateWithFeedback

Require: An optimizer
Require: Decay rate for relative change β_3
Require: $d_0 = 1$
Require: $t = 0$
while stop condition is not reached **do**
  $t \leftarrow t+1$
  if $t > 1$ **then**
    if $f_{t-1} \geq \hat{f}_{t-2}$ **then**
      $\delta_t \leftarrow k+1$
      $\Delta_t \leftarrow K+1$
    else
      $\delta_t \leftarrow \frac{1}{K+1}$
      $\Delta_t \leftarrow \frac{1}{k+1}$
    $c_t \leftarrow \min\left\{\max\left\{\delta_t, \frac{f_{t-1}}{\hat{f}_{t-2}}, \Delta_t\right\}\right\}$
    $\hat{f}_{t-1} \leftarrow c_t \hat{f}_{t-2}$
    $r_t \leftarrow \frac{|\hat{f}_{t-1} - \hat{f}_{t-2}|}{\min\{\hat{f}_{t-1}, \hat{f}_{t-2}\}}$
    $d_t \leftarrow \beta_3 d_{t-1} + (1-\beta_3) r_t$
  else
    $\hat{f}_{t-1} \leftarrow f_{t-1}$
    $d_t \leftarrow 1$
  optimizer.update_with_feedback(f_{t-1}, d_t)

Algorithm 2 Adam.update_with_feedback(f_{t-1}, d_t)

Require: Learning rate α
Require: Decay parameters β_1, β_2
Require: Fuzz factor ε
Require: 1st and 2nd order gradient moments m_t and v_t
while stop condition is not reached **do**
  $g_t \leftarrow \nabla_\theta f(\theta_{t-1})$
  $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
  $\hat{m}_t \leftarrow \frac{m_t}{1-\beta_1^t}$
  $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
  $\hat{v}_t \leftarrow \frac{v_t}{1-\beta_2^t}$
  $\theta \leftarrow \theta_{t-1} - \alpha \frac{\hat{m}_t}{d_t \sqrt{\hat{v}_t} + \varepsilon}$

3.3 Adding feedback clipping on Eve optimizer

During the experiments we quickly discovered that when we used the default settings of the Eve optimizer on the MNIST dataset, we would obtain numerical overflow for d_t after about 500 minibatches, despite the clipping as seen in Equation (16). We found that we could overcome this by directly clipping the value of d_t. So essentially, we propose to replace Equation (19) with:

$$d_t = \min\left\{\max\{\beta_3 d_{t-1}, d_\perp\}, d_\top\right\} \tag{21}$$

$d_\top$ is defined as the upper threshold for the clipping and $d_\perp$ is defined as the lower threshold. Different values of the upper threshold will be assessed in the experiments to see what the effects are on training the model.

In addition, we need to make sure that the denominator of equation (18) does not come close to zero. We accomplish this by adding a fuzz factor to this equation, such that r_t is now computed as

$$r_t = \frac{|\hat{f}_{t-1} - \hat{f}_{t-2}|}{\min\{\hat{f}_{t-1}, \hat{f}_{t-2}\} + \varepsilon_{r_t}}, \tag{22}$$

where $\varepsilon_{r_t} = 10^{-8}$ by default.

3.4 RMSProp

The RMSProp algorithm [23] adapts its gradient updates according to a the root of a running average of the square gradient. This means that

the gradient updates are given by:

$$E[g^2]_t = \beta_1 E[g^2]_{t-1} + (1-\beta_1) g_t^2, \tag{23}$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t + \varepsilon}} g_t, \tag{24}$$

Where $E[g^2]_t$ is the running average of the squared gradient, β_1 is the corresponding decay parameter, g_t is the gradient at time t and ε is the fuzz factor that is required for numerical stability.

3.5 RMSProp with feedback

To test whether the use of loss function feedback could yield improved performance for other adaptive optimizers, we explore the extension of the RMSProp algorithm with feedback. The feedback is applied in a similar way, so again, we can make use of the generic algorithm listed in Algorithm 1. The only difference with respect to the Eve optimizer is the optimizer that provide is provided. The RMSProp optimizer with feedback is described in Algorithm 3. Note that we omitted the computation of a momentum term as the default momentum for RMSProp is zero and we did not consider this to be the most important parameter to investigate. The feedback is applied once more by multiplying it with the square root of the denominator, similar to the Eve algorithm.

In our experiments, we have also explored the difference in performance when d_t is both clipped and when it is not clipped at all.

Algorithm 3 RMSProp.update_with_feedback($f(\theta_{t-1}), d_t$)

Require: Learning rate α
Require: Decay parameters β_1, β_2
Require: Fuzz factor ε
Require: Mean square m_t
1: **while** stop condition is not reached **do**
2: $g_t \leftarrow \nabla_\theta f(\theta_{t-1})$
3: $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t^2$
4: $\theta \leftarrow \theta_{t-1} - \alpha \frac{g_t}{d_t \sqrt{m_t + \varepsilon}}$

4 Data and Experiments

This section discuss the data sets and the exact experiment setup.

4.1 Data

For the experiments we are using three different data sets. We consider the MNIST [14], Cifar10 [12] and OxfordFlower17 [16] datasets. The Cifar10 data set is also used in the comparison of [11], so here it is used to verify whether their results are reproducible. The OxfordFlower17 and MNIST data sets are added to evaluate whether the Eve algorithm also performs well outside the domain of the previous experiments in [11]. Table 1 provides and overview of the data sets.

Table 1: Overview of the Mnist, cifar10 and oxflower 17 datasets. The number of samples and classes are specified in this table.

Name	Number of samples	Number of classes
Mnist [14]	70K	10
Cifar10 [12]	60K	10
Oxfordflower 17 [16]	1360	17

4.2 Experiments

This section covers the model's architecture that was used throughout the experiments and the exact experimental setup in terms of the hyperparameters.

4.2.1 Model architecture

In our experiments we use a convolutional neural network (CNN). A CNN is a deep neural network that exploits the spatial correlations in images, yielding state-of-the-art performance in a wide range of image classification tasks [13, 14]. The CNN is trained and tested on the image data sets. The CNN that is used here is relatively small as it has no more than 4 parameterized layers. This architecture is used for all image classification tasks that follow. It is the same architecture that Koushik et al. used for their experiments on Cifar10 [11]. This model consists of a 6-layered CNN (5 hidden), there are two blocks of convolutional layers with 3x3 kernels. The first convolutional layer has 32 filters and the second has 64. Each convolutional layer is followed by 2x2 max pooling with 2x2 strides and 0.25 dropout. Max pooling is a downsampling procedure in which a patch shifts over a spatially arranged set of neurons. The patch maps the values under it to the maximum of these neuron activations. This provides translation invariance, meaning that small distortions of the image will not have a notable effect on the hidden activations. This can be found in many deep learning architectures, e.g. [13, 14]. Dropout is a technique in which neurons are randomly turned off during training w ith a certain probability p_d (in our case $p_d = 0.25$) [20]. However, during testing all neurons are active and so the test network acts as if it averages the predictions of all random permutations of neurons that were encountered during training. This reduces the risk of overfitting.

After the convolutional block the feature maps are fed into a fully connected layer with 512 nodes and a softmax layer for which the number of neurons is equal to the number of classes. All layers are followed by a rectified linear unit (ReLU) that computes its activation as $f(x) = \max\{0, x\}$. This activation function has become the default choice in deep learning architectures since their introduction by Glorot et al. [8], mainly because it typically has larger gradients than the sigmoid activation function where $f(x) = 1/(1+\exp(-x))$.

4.2.2 Experiment setup

The hyperparameters that were used are listed in Table 2. For some of these hyperparameters, multiple values are listed. In that case, the setting marked with bold font indicates the default setting and the other values might have been the varying parameter in our experiments. For each data set that we considered, we measured the performance on a 5-fold cross validation, meaning that we shuffled the data randomly after which we split the data evenly in 5 sets. For the k-th cross validation iteration, we left out the k-th set as a test set and train on the other sets. We report the average accuracies on the test set. The Adam and RMSProp optimizers were taken from the Keras library [5].

Table 2: Overview of hyperparameters

Hyperparameter	Value
α	0.001 (MNIST and Cifar10), 0.0001 (OxFlower)
β_1	0.9
β_2	0.999
β_3	0.999
ε	10^{-8}
$d_\perp$	$-\infty$, **0.1**
$d_\top$	**10**, 100, 1000, ∞
ε_{r_t}	10^{-8}

5 Results

This section discusses the main results of our experiments and we try to characterize the differences between the optimization algorithms.

5.1 Eve optimizer with feedback clipping

Figure 1 illustrates the averaged test results of the 5-fold cross validated CNNs with different values for the feedback clipping of the Eve optimizer. We can see that the default setting of the clipping parameter ($d_\perp = \infty$) causes an unstable optimizer, leading to chance-level performance. Presumably, the instability arises from the fact that the denominator in equation (18) attains a value that is close to zero. The observation of the fact that such issues do not arise when $d_\perp \in \{10, 100, 1000\}$ supports this notion. For these values, the optimizer is able to find an adequate solution within reasonable time. To be able to distinguish between the former three cases, we provide a more detailed view in Figure 2. Here we can see that $d_\perp = 10$ results

in the highest average accuracy followed by $d_\perp = 100$ and finally we see that $d_\perp = 1000$ results in the lowest accuracy. As we will see in Section 5.2, these results are considerably better than the default Eve optimizer as described in [11].

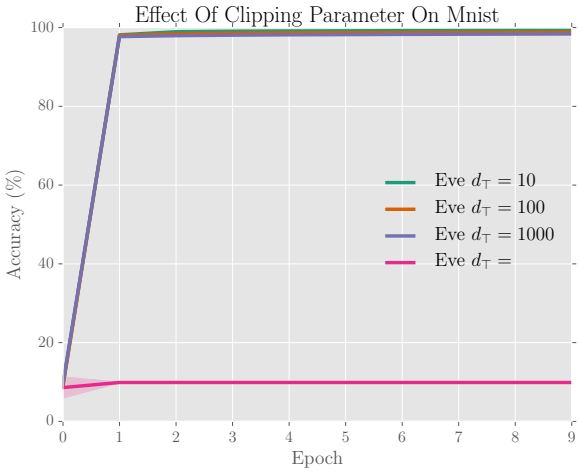


Fig. 1: Mean accuracy (%) vs. epoch on MNIST with different values for the feedback coefficient upper clipping $d_\top$ in which the shaded areas indicate the standard deviations. The optimizer that is used here is the Eve optimizer. The $d_\top = \infty$ corresponds to the Eve algorithm as in [11].

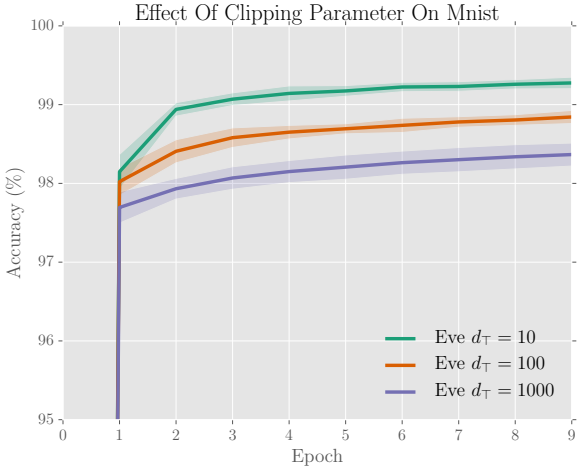


Fig. 2: Mean accuracy (%) vs. epoch on mnist in which the shaded areas indicate the standard deviations. This is view is to visualize the differences among the various settings of $d_\top$. To see how these settings compare to $d_\top = \infty$, see Figure 1.

5.2 Comparing Adam, Eve, RMSEve and RMSProp on MNIST

Figure 3 shows the classification accuracy for Adam, Eve and RMSEve on the MNIST dataset. Note that in these experiments, $d_\perp$ was set to 10 which is motivated by the results that we discussed in the previous section. We can clearly see that the Eve optimizer has a steeper accuracy curve than the other optimizers. This means that the Eve optimizer learns faster on average on this particular dataset and with the settings as described in Table 2. The Adam optimizer appears to outperform the RMSEve optimizer too. Perhaps we can observe notable differences if we were to continue learning for even more epochs, but we have not included such results due to time constraints.

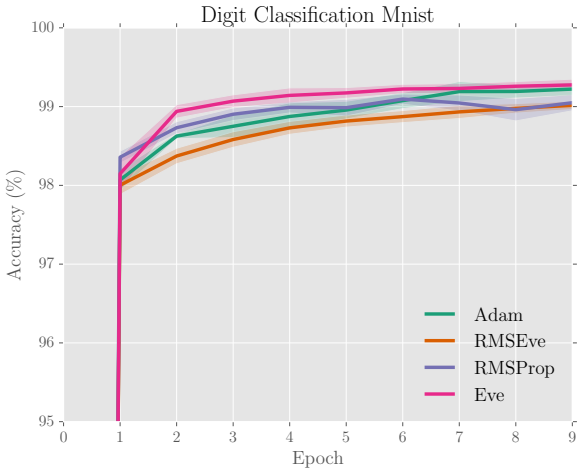


Fig. 3: Mean accuracy vs. train step on MNIST in which the shaded areas indicate the standard deviations. The figure depicts the resutls for the Adam, Eve, RMSEve and RMSprop optimizers.

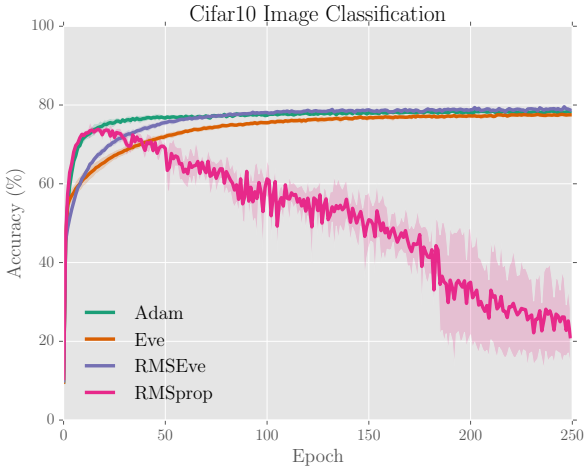


Fig. 4: Mean accuracy (%) vs. epoch on cifar10 in which the shaded areas indicate the standard deviations. The figure displays the results of the Adam, Eve, RMSEve and RMSprop optimizers.

5.3 Comparison on Cifar10 dataset

Figure 4 displays the results on the Cifar10 image classification task. We no longer display Koushik et al.'s version of the Eve optimizer due to time constraints. Nevertheless, we can observe that the RSMEve algorithm has no problem to converge to a proper solution within 250 epochs. Moreover, the RMSProp algorithm seems unstable when compared to RMSEve. Apparently, the use of the feedback coefficient sufficiently slows down learning to avoid oscillation and other unpredictable trajectories. Most surprisingly, the Eve optimizer does not seem to outperform Adam. Perhaps this is due to the fact that the loss was close to zero, causing the d parameter to be clipped and thereby forcing an abrupt slowdown of the learning rate. Also, these results do not reproduce the findings in [11], because in our research, the Eve optimizer does not perform better than the Adam optimizer, whereas in the work by [11], it does. This is presumably because in [11] the authors tried several different learning rates and report their best results among these different learning rates. Moreover, we have slightly modified the Eve algorithm to be more stable. However, because of the lower clipping of d_t, the learning speed might not be as high as it would have been if we would not use any clipping.

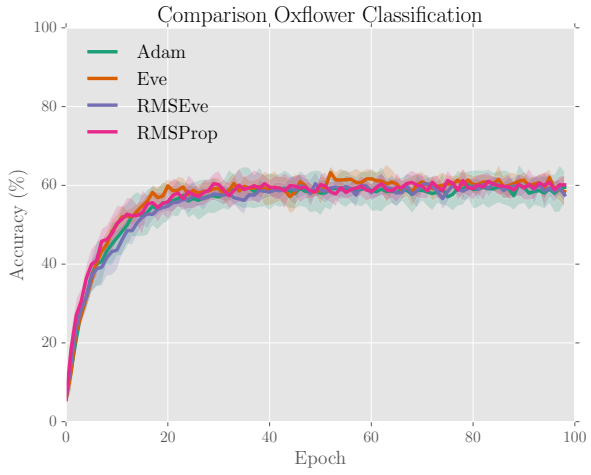


Fig. 5: Mean accuracy (%) vs. epoch on oxflower in which the shaded areas indicate the standard deviations. The figure shows the results for the Adam, Eve, RMSEve and RMSprop optimizers.

5.4 Comparison on OxFlower

Figure 5 depicts the average classification accuracies vs. the number of epochs for the OxfordFlower17 dataset. In this figure, we can see that all optimizers exhibit comparable performance, both in terms of speed and in terms of the final performance. Perhaps the bottleneck for this problem is not the optimizer itself, but the architecture that was used. This might also explain why the results are considerable lower than the results that were presented in [17], in which the authors used a different classification method.

6 Discussion

In this paper we have considered a range of experiments with optimizers that use loss function feedback to scale their parameter updates. We have found that numerical instabilities can severely hinder if not completely disable the actual performance. The most significant contribution of this paper is the introduction of the feedback clipping that resolves this problem as is clearly demonstrated in the case of MNIST. Based on the results that were obtained here, it is difficult to argue whether the Eve optimizer is superior to the Adam optimizer. To make a more reliable claim about this, we would need to do a more research across other domains in machine learning and e.g. assess different learning rates. Perhaps such research will show significant differences between the optimizers in terms of robustness and convergence speeds. Another important observation is that with our settings we find an effect that contradicts the findings by [11] as we can see that in the case of Cifar10, the Eve optimizer performs worse than the Adam's optimizer.

Apart from the parameter searches that are discussed in this paper, there could be a more thorough parameter search for the remaining parameters. One particularly interesting parameter to vary would be the lower bound for the feedback factor $d_{\perp}$, as we now only considered two different values. One could also choose to set a lower bound on the feedback factor by simply adding a small constant, which will have a similar limiting effect, but it would mean that this bound is asymptotically approached, instead of being surpassed abruptly.

For future work, it would be interesting to consider even more challenging optimization problems, particularly for recurrent neural networks, as these were the cases in which the improvement of the Eve algorithm with respect to the Adam algorithm were the most significant in [11]. Moreover, it would be interesting to consider per-neuron errors instead of errors that are describing the error of the whole classifier. Using the error per neuron this way might yield a more weight-specific learning rate adaptation.

References

[1] P. Bell, P. Swietojanski, and S. Renals. Multitask learning of context-dependent targets in deep neural network acoustic models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(2):238–247, 2017.

[2] L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998.

[3] A. Cauchy. Méthode générale pour la résolution des systemes déquations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

[4] X. Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2422–2431, 2015.

[5] F. Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

[6] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 411–418. Springer, 2013.

[7] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[8] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Aistats*, volume 15, page 275, 2011.

[9] A. K. Jain and S. Z. Li. *Handbook of face recognition*. Springer, 2011.

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] J. Koushik and H. Hayashi. Improving stochastic gradient descent with feedback. *arXiv preprint arXiv:1611.01505*, 2016.

[12] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/k2). In *Doklady an SSSR*, volume 269, pages 543–547, 1983.

[16] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454, 2006.

[17] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 722–729. IEEE, 2008.

[18] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[19] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[20] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[21] R. S. Sutton. Two problems with backpropagation and other steepest-descent learning procedures for networks. In *Proc. 8th annual conf. cognitive science society*, pages 823–831. Erlbaum, 1986.

[22] A. Thanda and S. M. Venkatesan. Multi-task learning of deep neural networks for audio visual automatic speech recognition. *arXiv preprint arXiv:1701.02477*, 2017.

[23] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 2012.

[24] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 14, pages 77–81, 2015.

[25] M. D. Zeiler. Adadelta: An adaptive learning rate method. arxiv preprint. *arXiv preprint arXiv:1212.5701*, 2012.

[26] R. Zhao and K. Mao. Topic-aware deep compositional models for sentence classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(2):248–260, 2017.

Assessing the Novelty of the Extreme Learning Machine (ELM)

Fthi Abadi, Remi Brandt

Abstract—Extreme learning (Haung et *al.* in 2004) is a learning algorithm for single hidden layer feedforward neural networks. In the ELM, the weights from the input layer and bias to the hidden layer neurons are set randomly. The weights from the hidden layer to the output layer neurons are determined analytically. According to its authors, the ELM provides superior generalization and learning efficiency compared to the prior learning algorithms which use an iterative strategy to determine neural network parameters.

It has been claimed that the essence of the ELM was proposed prior to Haung et *al.* by Schmidt et *al.* in 1992, Pao et *al.* in 1992 and Broomhead and Lowe in 1988. This has led to a debate concerning the novelty of the ELM and hence the necessity and justification to introduce a new name: "Extreme Learning Machine".

The ELM as well as prior work of which it is supposed that the essence is equal to that of the ELM (Schmidt et *al.* in 1992, Pao et *al.* in 1992 and Broomhead and Lowe in 1988) will be discussed. The similarities and differences of the algorithms proposed in said papers with respect to learning strategy, approximation and unknown determination will be discussed. Subsequently, arguments regarding the uniqueness of the methodologies in the ELM compared to the methodologies proposed by the prior work will be presented. Finally, it is concluded that the ELM cannot be considered novel because of it's strong similarities with related earlier works.

Field of research: Artificial neural networks (ANN's). **Topic:** Extreme Learning Machine (ELM). **Focus** / **Research question:** To what extent can Extreme Learning Machine be considered novel? **Results:** According to the authors of this paper, the ELM cannot be considered novel because of it's strong similarities with related earlier works.

Index Terms—Feedforward neural network, Multilayer perceptron network, Radial basis function network, Random vector functional-link network, Extreme Learning Machine, ELM.

1 Introduction

Artificial neural networks (ANNs) are an approach to solve computational problems. In principle, they have the power of a universal approximator, i.e. they can realise an arbitrary mapping of one vector space onto another vector space [9]. Real world applications of ANNs include risk management, stock market forecasting and speech recognition.

As described in [9], the design of ANNs was inspired by that of biological neural networks which are present in brains. ANNs are networks of formal neurons. These neurons are organized in layers, i.e. input layer, output layer and layers in between: hidden layers. Neurons in the input layer and bias neurons provide a (constant) output signal and receive no input signal. Neurons in hidden layers receive input signals which are input to an activation function that determines an output signal. Neurons in the output layer give as output signal the weighted sum of their input neurons. Each neuron in an ANN is connected with at least one other neuron, and each connection is weighted by a numerical value that indicates the importance of the connection. These weight parameters may be set manually or learned by an algorithm.

A vast amount of ANN types and classes have been proposed, two of which are Feed forward neural networks and Single layer feedforward neural networks [9]: Feed forward neural networks are ANNs of which connections between the neurons do not form a cycle. Single layer feedforward neural networks (SLFN) are feed forward neural networks which contain one hidden layer. Other ANN types and classes have been proposed as well which will be discussed in detail in Section 2.1.

- *Fthi Abadi is an MSc Computing Science student at the University of Groningen, E-mail: f.a.abadi@student.rug.nl, S-number: 3074641*
- *Remi Brandt is an MSc Computing Science student at the University of Groningen, E-mail: r.brandt@student.rug.nl, S-number: 2509644*

As detailed in [5], the modeling of a mapping of one vector space onto an other vector space using a SLFN can be defined as

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = t_j,\ j = 1,2,...,N, \tag{1}$$

where N denotes the number of samples (x_i,t_i), $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^\mathsf{T} \in R^n$, $t_i = [t_{i1}, t_{i2}, ..., t_{im}]^\mathsf{T} \in R^m$, $\tilde{N}$ denotes the number of hidden neurons, $g(\cdot)$ is an activation function, $w_i = [w_{i1}, w_{i2}, ..., w_{in}]^\mathsf{T}$ is the weight vector connecting the ith hidden neuron and the input neurons, $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{in}]^\mathsf{T}$ is the weight vector connecting the ith hidden neuron and the output neurons, b_i is the bias of the ith hidden neuron and $w_i \cdot x_j$ denotes the inner product of w_i and x_j. Note that this notation will be used as well in the remainder of this paper.

A way to learn SLFN weight parameters from N samples is to use the Extreme Learning Machine (ELM) algorithm [5]: In the ELM, the weights from the input layer and bias to the hidden layer neurons in a single hidden layer neural network are set randomly and the weights from the hidden layer to the output layer neurons are determined analytically. According to its authors, the ELM provides superior generalization and learning efficiency compared to the regular learning algorithms which use an iterative strategy to determine feedforward neural network parameters.

It has been claimed that the essence of the ELM was proposed prior to Haung et *al*. by Schmidt et *al*. in 1992 [8], Pao et *al*. in 1994 [7] and Broomhead and Lowe in 1988 [1]. This prior work was not cited in the paper proposing the ELM. This has led to a debate concerning the novelty of the ELM and hence the necessity and justification to introduce a new name: "Extreme Learning Machine".

Plagiarizing is the act of using other peoples work without authorization and giving credit. It hence concerns both similarity and origin. It is important that it is evaluated whether or not the ELM plagiarizes similar work for a number of reasons:

It will function as an example of a work which is in the long term not falsely classified as plagiarism or non-plagiarism. Such an example will regardless of the outcome encourage to not commit plagiarism.

Awarded funding for research in the ELM could be redirected into research into the plagiarized work if the ELM will be evaluated as

plagiarized. Such a redirection will be an encouraging example of work being rewarded fairly.

In this paper, prior work of which it is supposed that the essence is equal to that of the ELM will be discussed and compared to the ELM. Note that we will compare these works to the original version of the ELM [5]. We will only assess the similarity between the works and we will not speculate about whether or not possible similarities are the result of unauthorized copying. Subsequently, the uniqueness of the ELM compared to said related work will be evaluated based on similarities between the learning algorithms. Lastly, conclusions made through the viewpoint of the authors of this proposal will be given which indicate whether or not the ELM is considered novel.

This paper is organized as follows. Section 2 gives definitions of the compared algorithm characteristics. Section 3 gives an overview of the compared algorithms. Section 4 details an analysis of similarity between the ELM and the similar works. Results will be discussed in Section 5. A conclusion will be presented in Section 6. Finally, future work will be discussed in Section 7.

2 Comparison criteria

In this section the characteristics which have been considered during the comparison of the different algorithms will be defined: how the approximation problem is solved, learning strategy and the way in which unknowns in overdetermined systems (a system of equations with more equations than unknowns) are determined.

2.1 Approximation problem

In principle, neural networks have the power of a universal approximator, i.e. they can realise an arbitrary mapping of one vector space onto another vector space [9]. A variety of different types of ANNs have been used in the compared learning algorithms which allow for the respective networks to act as a universal approximator.

2.1.1 Single hidden layer feedforward neural network (SLFN)

The ELM uses SLFNs. In the ELM a mathematical model is described in Equation (7) that proved itself to be a universal approximator when output weights are adjusted incrementally by adding a new hidden neuron at every iteration of the training [11]. Outputs weights can also be updated using the Moore-Penrose generalized inverse or sequentially as a new data arrives in real-time applications [11].

Schmidt et *al.* in 1992 [8] also implement Moore-Penrose minimum norm inverse to compute the output weights just like the ELM does with a small difference on bias. Unlike the ELM, the algorithm proposed by Schmidt et *al.* has a bais to the output layer. An illustration of the type of ANN (SLFN) used by the ELM is illustrated in Figure (1).

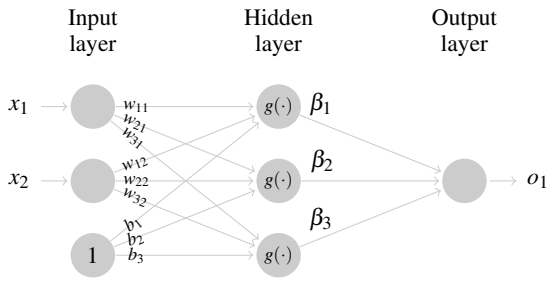


Fig. 1. Single hidden layer feedforward network used by ELM. Note that an arbitrary number of input, hidden and output neurons was chosen. Used symbols are defined in Section 1 and 3.1. Derived from [5].

2.1.2 Radial basis function network (RBF)

Broomhead and Lowe [1] showed that randomly selecting hidden radial basis function centers is sufficient to allow universal approximation. Radial basis function networks are artificial neural networks that use a radial basis function $\phi(\cdot)$ as activation function and the output of the network is the weighted sum of the results from these radial functions.

A radial basis function $\phi(\|x_j - y_i\|)$ is a real valued function where $x_j \in R^n$ and $\|...\|$ denotes a distance measure imposed on R^n which is in this paper euclidean. The vectors $y_i \in R^n, i = 1,2,...,m$ are the centers of the basis functions where m denotes the number of input neurons.

RBF networks can be used to approximate a function by learning through techniques like training the basis function centers, Pseudo-inverse solution or gradient descent training. Broomhead and Lowe [1], as well, implemented the Pseudo-inverse solution for computing the output weight that are between the hidden layer and the output layer. An illustration of RBF is given in Figure (2).

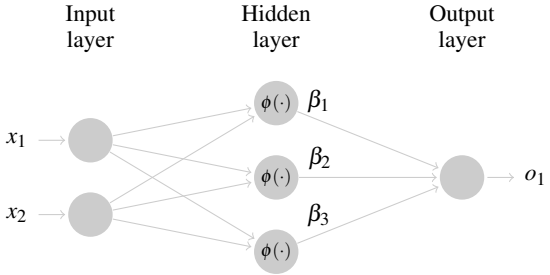


Fig. 2. Radial basis function network. Note that an arbitrary number of input, hidden and output neurons was chosen. Used symbols are defined in Section 1 and 3.3. Derived from [1].

2.1.3 Random vector functional-link network (RVFL)

Pao et *al.* [7] proposed the random vector functional-link (RVFL) network. The learning algorithm proposed by Pao et *al.* implements Moore-Penrose generalized inverse like the ELM does except it allows a direct link from the input neurons to the output neurons [11]. Igelnik and Pao [6] also proved the RVFL neural network with single hidden layer, is a universal approximator both when learning the weight parameters to minimize overall system error and in generalization performance. An illustration of RVFL is given in Figure (3).

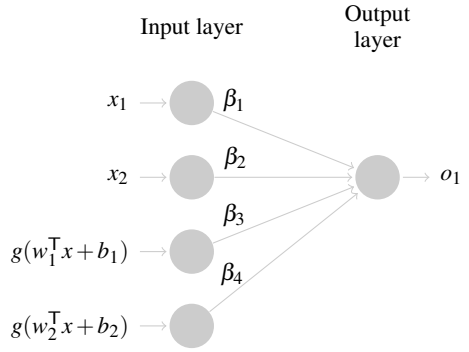


Fig. 3. Random vector functional-link network. Note that an arbitrary number of input neurons was chosen. Used symbols are defined in Section 1 and 3.4. Derived from [7].

2.2 Learning strategy

Gradient-based learning [2] is an iterative technique used to learn the adaptive parameters of a neural network. A basic gradient learning algorithm updates a learnable parameter p of a neural network using the general rule

$$p_{new} = p_{old} - \alpha_p \frac{\partial E}{\partial p}, \tag{2}$$

where α is a learning rate and E represents the error of the whole training set or a single training instance depending on the choice of training strategy. The ELM as well as the discussed similar works do not implement such approach because, according to authors of the ELM, gradient-based learning or error propagation has the following issues [5]:

First, how fast the algorithm converges depends on the value of the learning rate . The presence of a learning rate leads to unstable algorithms. Second, Backpropagation learning algorithms tend to surface on a local minimum even when the algorithm is located far above a global minimum. Third, gradient based learning leads to worse generalization performance which requires suitable stopping methods to mitigate the problem. Finally, gradient-based learning is a very time consuming strategy when applied in most applications.

The ELM as well as the discussed similar works initialize certain parameters of the neural network randomly to overcome the negative qualities of gradient based learning. Because of said random initialization, the problem of learning the adaptive parameters of a neural network is changed to one which can be solved analytically.

2.3 Unknowns determination in overdetermined systems

The compared learning algorithms determine network parameters analytically. A number of methods have been proposed to determine unknowns in overdetermined systems.

2.3.1 Minimum norm least-squares solution of general linear system

As stated in [5], least square solutions are methods used to estimate unknowns in a linear regression model with the goal of minimizing the sum of the squares of the difference between the observed value and the value predicted by the model. For a general linear system $\mathbf{A}x = t$ in Euclidean space, where $A \in R^{m \times n}$ and $t \in R^m$, $\hat{x}$ is a least-squares solution if

$$\|\mathbf{A}\hat{x} - t\| = \min_x \|\mathbf{A}x - t\|, \tag{3}$$

where $\|...\|$ is a norm in Euclidean space. A solution is said to be minimum norm least-squares solution of a line system $\mathbf{A}x = t$ if it has the smallest norm among all the least squares solutions. [5]

2.3.2 Moore-Penrose generalized inverse

A matrix $\mathbf{G}$ of order $n \times m$ is the Moore-Penrose generalized inverse of matrix $\mathbf{A}$ of order $m \times n$ if [5]:

$$\mathbf{AGA} = \mathbf{A},\ \mathbf{GAG} = \mathbf{G},\ (\mathbf{AG})^\mathsf{T} = \mathbf{AG},\ (\mathbf{GA})^\mathsf{T} = \mathbf{GA}. \tag{4}$$

As stated in [5], if there exists a matrix $\mathbf{G}$ such that $\mathbf{G}$t is a minimum norm least-squares solution of a line system $\mathbf{A}x = t$, then $G = \mathbf{A}^\dagger$. Where $\mathbf{A}^\dagger$ is the Moore-Penrose generalized inverse of matrix $\mathbf{A}$.

2.3.3 L_2 criterion

Schmidt et *al*. [8] optimizes the output weights using the L_2 criterion, also called the least square procedure. The L_2 criterion is commonly used to estimate the regression coefficients [10]. Schmidt et *al* used the L_2 criterion to compute the output weights by fitting them to a model generated by the input weights and the corresponding input data. The resulting weight vector is applied to a Fisher Equation (16) to search for a minimal. This is discussed in detail in Section 3.2.

Given the randomly selected input to hidden layer weights as M and the hidden layer to output weights as W, the following equations describe how the paper used the L_2 criterion to optimize the output weights.

$$E^2 = \sum_{i=0}^{N} (t - \mathbf{W}^\mathsf{T} O_{hidden})^2, \tag{5}$$

where t is the desired output and W is the output weights needed to be optimized. O_{hidden} is defined as

$$O_{hidden} = ([F(M_1^\mathsf{T}X), F(M_2^\mathsf{T}X), ..., F(M_N^\mathsf{T}X)])^\mathsf{T}. \tag{6}$$

Given the set of chosen input weights (i.e. input layer to hidden layer weights) we get a constant O_{hidden} which makes equation (5) a linear one. The result of the L_2 criterion will be fed to the Fisher derivation technique to get the final output weight vector as explained in detail in Section 3.2.

3 Description of methods

In this section we will detail the compared algorithms. The algorithms will be explained in terms of how the approximation problem is solved, learning strategy and the way in which unknowns in overdetermined systems are determined. Based on these properties an answer to our research question can be derived.

3.1 Huang, Zhu and Siew

The Extreme Learning Machine is a learning algorithm for single hidden layer feedforward neural networks proposed by Huang, Zhu and Siew in 2004 [5].

As detailed in [5], a standard SLFN can approximate N samples with zero error given that $\tilde{N} \leq N$. Therefore, there exists β_i, w_i and b_i such that

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = t_j,\ j = 1, 2, ..., N. \tag{7}$$

These N equations can be written compactly as

$$\mathbf{H}\beta = \mathbf{T} \tag{8}$$

where

$$\mathbf{H} = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & ... & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ . & ... & . \\ . & ... & . \\ . & ... & . \\ g(w_1 \cdot x_N + b_1) & ... & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \tag{9}$$

$$\beta = \begin{bmatrix} \beta_1^\mathsf{T} \\ . \\ . \\ . \\ \beta_{\tilde{N}}^\mathsf{T} \end{bmatrix}_{\tilde{N} \times m} \quad \textit{and}\ \mathbf{T} = \begin{bmatrix} t_1^\mathsf{T} \\ . \\ . \\ . \\ t_N^\mathsf{T} \end{bmatrix}_{N \times m} \tag{10}$$

The ith column of $\mathbf{H}$ is the ith hidden neurons output vector with respect to inputs $x_1, x_2, ..., x_N$.

The ELM is designed around the understanding that the input weights w_i and the hidden layer biases b_i do not have to be learned and $\mathbf{H}$ can hence be set randomly.

Said understanding allows to define training a SLFN to be equivalent with finding a least-squares solution $\hat{\beta}$ of the linear system $\mathbf{H}\beta = \mathbf{T}$:

$$\|\mathbf{H}\hat{\beta} - \mathbf{T}\| = \min_\beta \|\mathbf{H}\beta - \mathbf{T}\|. \tag{11}$$

The smallest norm least squares solution of the above linear system is $\hat{\beta} = \mathbf{H}^\dagger\mathbf{T}$, where $\mathbf{H}^\dagger$ denotes the Moore-Penrose generalized inverse of matrix $\mathbf{H}$.

To summarize, the Extreme Learning Machine algorithm is defined as follows [5]:

Data: Training set $\sigma = \{(x_i,t_i)|x_i \in R^n, t_i \in R^m, i = 1,2,..,N\}$
Activation function $g(\cdot)$
Hidden neuron number $\tilde{N}$.
Result: Single hidden layer neural network with tuned weights
Assign arbitrary input weight w_i and bias $b_i, i = 1,...,\tilde{N}$.;
Calculate $\mathbf{H}$.;
Calculate the output weight $\hat{\beta}$ as the smallest norm least-squares solution

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{T}, \tag{12}$$

where $\mathbf{H}$, $\hat{\beta}$ and $\mathbf{T}$ are defined as Equation (9) and (10).
Algorithm 1: Extreme Learning Machine algorithm.

3.2 Schmidt, Kraaijveld and Duin

Schmidt, Kraaijveld and Duin have proposed a learning algorithm for single hidden layer feedforward neural networks in [8], which will be outlined in this section.

The learning algorithm proposed by Schmidt et *al.* uses the same SLFN as the ELM uses with some exceptions: The bias neuron is connected to the output neuron directly and not to the hidden layer. The output is defined by

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j) + \beta_{\tilde{N}+1} = t_j,\ j = 1,2,...,N, \tag{13}$$

where $w = [w_1, w_2, ..., w_{\tilde{N}}]^{\mathsf{T}}$, $x_i = [x_{i1}, x_{i2}, ..., x_{i\tilde{N}}]^{\mathsf{T}}$ and β is defined as before with the exception that $\beta_{\tilde{N}+1}$ denotes bias weight.

The Fisher method is used to determine weights. Weights are optimized by using the L_2 criterion

$$E^2 = \sum_{j=1}^{N}(t_j - \sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j) + \beta_{\tilde{N}+1})^2 = \sum_{j=1}^{N}(t_j - \beta^{\mathsf{T}}\mathbf{O}_{hidden})^2, \tag{14}$$

where $\mathbf{O}_{hidden}$ is defined as $\mathbf{O}_{hidden} = [g(w_i^{\mathsf{T}}x_j), ..., g(w_{\tilde{N}}^{\mathsf{T}}x_j), 1]^{\mathsf{T}}$.

The Schmidt, Kraaijveld and Duin algorithm is build around the assumption that input weights w may be set to uniform [-1..1] random values. Therefore, $\mathbf{O}_{hidden}$ is fixed. The parameters β are then optimized for the chosen set of weights w. Because the optimization problem is reduced to finding the unknowns in a set of equations, neural network parameters can be determined analytically.

The weight vector $\hat{\beta}$ for which E^2 as defined in Equation (14) is minimal is calculated analytically using Equation (16).

$$\frac{\partial E^2}{\partial \beta} = 2\mathbf{R}\hat{\beta} - 2\mathbf{P} = 0 \rightarrow \tag{15}$$

$$\hat{\beta} = \mathbf{R}^{\dagger}\mathbf{P}, \tag{16}$$

where

$$\mathbf{R} = \sum_{j=1}^{N} \mathbf{O}_{hidden}\mathbf{O}_{hidden}^{\mathsf{T}}, \tag{17}$$

$$\mathbf{P} = \sum_{j=1}^{N} t_j\mathbf{O}_{hidden}^{\mathsf{T}}, \tag{18}$$

and $\mathbf{R}^{\dagger}$ denotes the Moore-Penrose minimum norm inverse of $\mathbf{R}$. Note that $\mathbf{R}$ and $\mathbf{P}$ are the input correlation matrix and input-target correlation vector respectively of the output unit.

To summarize, the Schmidt et *al.* learning algorithm for SLFNs is defined as follows [8]:

Data: Training set $\sigma = \{(x_i,t_i)|x_i \in R^n, t_i \in R^m, i = 1,2,..,N\}$
Activation function $g(\cdot)$
Hidden neuron number $\tilde{N}$.
Result: Neural network with tuned weights
Assign uniform [-1..1] random values to input weights w;
Determine matrix $\mathbf{R}$ and $\mathbf{P}$ using Equation (17) and (18) respectively.;
Determine $\hat{\beta}$ by computing

$$\hat{\beta} = \mathbf{R}^{\dagger}\mathbf{P}. \tag{19}$$

Algorithm 2: Schmidt et *al.* learning algorithm for SLFNs.

3.3 Broomhead and Lowe

Broomhead and Lowe have proposed a learning algorithm for single hidden layer radial basis function networks in [1], which will be outlined in this section.

Output of the radial basis network is defined as

$$\sum_{i=1}^{\tilde{N}} \beta_i \phi(\|x_j - y_i\|) = t_j,\ j = 1,2,...,N, \tag{20}$$

where t_j is the expected output of the radial basis network for input x_j. $y_i \in R^n, i = 1,2,...,\tilde{N}$ are the centers of the basis functions. $\phi(\cdot)$ is a generally non-linear activation function. $\|...\|$ is a norm imposed on R^n where $x \in R^n$.

Broomhead and Lowe have proposed that y_i may be set randomly. Said understanding allows to define training a Single hidden layer RBF network to be equivalent with finding a least-squares solution $\hat{\beta}$ of the linear system $\beta = \mathbf{A}^{\dagger}\mathbf{T}$, where β and $\mathbf{T}$ are defined as in Equation (10) and $\mathbf{A}$ is defined as

$$\mathbf{A} = \begin{bmatrix} \phi(\|x_1 - y_1\|) & ... & \phi(\|x_1 - y_{\tilde{N}}\|) \\ . & ... & . \\ . & ... & . \\ . & ... & . \\ \phi(\|x_N - y_1\|) & ... & \phi(\|x_N - y_{\tilde{N}}\|) \end{bmatrix}_{N \times \tilde{N}} \tag{21}$$

The smallest norm pseudo inverse is used to find $\hat{\beta}$: of all the vectors β which minimize the sum of squares $\|\mathbf{A}\beta - \mathbf{T}\|^2$, the one which has the smallest norm and hence minimizes $\|\beta\|^2$ is given by $\hat{\beta} = \mathbf{A}^{\dagger}\mathbf{T}$.

To summarize, the learning algorithm for RBNs is defined as follows [1]:

Data: Training set $\sigma = \{(x_i,t_i)|x_i \in R^n, t_i \in R^m, i = 1,2,..,N\}$
Radial basis activation function $\phi(\cdot)$
Hidden neuron number $\tilde{N}$.
Result: Neural network with tuned weights
Assign random values to hidden layer centers y;
Determine matrix $\mathbf{A}$ according to Equation (21);
Calculate the output weight β as the smallest norm least-squares solution

$$\hat{\beta} = \mathbf{A}^{\dagger}\mathbf{T}, \tag{22}$$

where $\mathbf{A}$ is defined as in Equation (21), and β and $\mathbf{T}$ are defined as in Equation (10).
Algorithm 3: Broomhead and Lowe learning algorithm for RBNs.

3.4 Pao, Park and Sobajic

Pao, Park and Sobajic have proposed a learning algorithm for vector Functional-link networks in [7], which will be outlined in this section.

Vector Functional-link networks contain as input neurons the input $X = [x_1, ..., x_N] \in R$ as well as enhancement neurons $[g(w_1^{\mathsf{T}}x + b_1), ..., g(w_J^{\mathsf{T}}x + b_J)]$ where b_j is the bias parameter for the neuron j

and $w_j^{\mathsf{T}} x$ is the input to the hidden layer neuron j. The output is defined by

$$\sum_{j=1}^{N+J} \beta_j g(w_j^{\mathsf{T}} x + b_j). \tag{23}$$

The learning algorithm proposed in [7] initializes weight vectors $\{w_j\}$ randomly with the constraint that the activation functions will not be saturated most of the time. Only the (N+J) weights β_j need to be learned.

Learning is by minimization of the system error defined as

$$E = \frac{1}{2N} \sum_{j=1}^{N} (t_j - \mathbf{B}^{\mathsf{T}} \mathbf{d}_j)^2, \tag{24}$$

where $\mathbf{B}^{\mathsf{T}}$ is the vector of weight values $\beta_j,\ j = 1,2,...,(N+J)$, and $\mathbf{d}$ is the enhanced input vector.

The remaining parameters can subsequently be solved with use of a pseudo-inverse if feasible.

To summarize, the Pao et *al.* learning algorithm for RVFLs is defined as follows [7]:

Data: Training set $\sigma = \{(x_i, t_i) | x_i \in R^n, t_i \in R, i = 1,2,..,N\}$
Activation function $g(\cdot)$
Result: Neural network with tuned weights
Assign random values to input weights a;
Calculate the output weight β with the use of a pseudo inverse.;
Algorithm 4: Pao, Park and Sobajic learning algorithm for RVFLs.

4 Results

This section lists the results of comparing Schmidt et *al.*, Broomhead et *al.* and Pao et *al.* with Huang et *al.* (ELM). In Table (1) the characteristics of the compared algorithms are summarized.

5 Discussion

As was discussed in Section 4 in detail, each of the three characteristics of the ELM have been compared with those of the other algorithms. It is clear that all three of the characteristics of ELM have either complete or partial similarity with those of the compared algorithms.

All of the compared algorithms use a linear output function. They all make use of randomization of the input weights. The ELM and Schmidt et *al.* use a single layer feedforward neural network. The ELM is the only algorithm which does not implement activation functions that use bias. The Moore-Penrose inverse or similar Pseudo-inverse computation is used by all compared algorithms to solve for unknowns in a system of equations.

Wang and Wan reacted to the ELM paper in a reply paper [11]. They claim that if all the similar previous works had been properly cited, discussed and compared to the ELM in the paper proposing the ELM, the paper proposing the ELM would not have been published at all. Wang and Wan justify their claim on the same grounds as the authors of this paper by pointing out the fact the ELM only added few links or removed few links in the neural network structure which is not enough to make the ELM paper novel.

Guang-Bin Huang, the main author of the ELM paper, in his reply paper [3] to the comments given by Wang and Wans [11] has discussed his point of view. The first thing the author pointed out to indicate novelty of his paper is regarding how Lowe's paper only selects centers of the RBF but not connection weights. The second point he mentioned is about the fact the prior works would not approximate well as the ELM does. Huang also mentioned that removing a link and adding a different one in the neural network does bring significant difference in approximation performance.

It is clear that the ELM does indeed have strong similarity with the works investigated. The papers mentioned and discussed here were published before the ELM paper was published.

The scientific way to conduct a study and publish your findings would be to study the prior literature and demonstrate that the findings are better than the latest finding. On the top of that the researcher is expected to give credits to the previous works and also reference them properly. Sadly, the ELM paper did not mention the papers we discussed and compared with in the previous sections.

6 Conclusions

In this paper we have given an answer to our research question: "To what extent can Extreme Learning Machine be considered novel". In this paper we have compared works similar to the ELM to derive an answer to our research question. In the opinion of the authors of this paper the ELM cannot be considered novel because of its strong similarities with related earlier works.

The fact that in the paper proposing the ELM, similar earlier works were not mentioned leads its readers to believe that the ELM is novel. However, considering the found similarities with other papers we can conclude that all aspects of the ELM are present in earlier works one way or another.

The authors of this paper stress, however, that this need not mean that the ELM is the result of plagiarism. Furthermore, it is only claimed that the ELM is not novel in the opinion of the authors of this paper, not that the ELM is factually not novel.

The name "Extreme Learning Machine" is in the opinion of the authors of this paper not necessary because of said lack of novelty. In [4] Huang argues that the ELM deserves its name despite its similarities with related works because he thinks "The ultimate goal of research is to find the truth of natural phenomena and to move research forward instead of arguing for being listed as "origins".". He points out that e.g. "feedforward neural networks" would also not deserve their own name and should have been called "perceptrons". The authors of this paper believe that there is indeed a need for new terms. However, we believe that only novel works deserve to be awarded a new term because this will help to encourage researchers to help the state of the art forward by not encouraging them to work on (variations of) existing works. We acknowledge that "novel" is an ambiguous term and also recognize that most works build on top of existing findings. We define a "novel" work as one which of which a significant segment of its elements are not present in prior works.

In this paper we have assessed novelty and not whether or not lack of novelty is the result of plagiarism. The impact of our results would have been greater if it had shown that the ELM is a result of plagiarism.

Overall, our findings imply that the name "Extreme leaning machine" is not needed because ELM lacks novelty in the opinion of the authors of this paper. Our findings will furthermore function as an example to show that works will not be not be falsely considered novel in the long term. Such an example will encourage researchers to work on truly novel works. Awarded funding for research in the ELM could be redirected into research into the discussed similar related works. Such a redirection will be an encouraging example of work being rewarded fairly.

7 Future work

We have only assessed the similarity between the works and we have not speculated about whether or not possible similarities are the result of unauthorized copying. Future work may focus on the origin aspect of plagiarism concerning the ELM.

Acknowledgements

The authors wish to thank Michael Biehl for his valuable feedback.

	Approximation problem	Learning strategy	Unknowns determination
Huang, Zhu and Siew	SLFN - Contains links from bias node to hidden layer nodes. Linear output layer function.	Bias and input weights are determined randomly. Hidden layer to output layer weights are subsequently determined analytically	Least squares solution is used for optimization. System of equations is solved using Moore-Penrose generalized inverse.
Schmidt, Kraaijveld and Duin	SLFN - Contains direct link from bias node to output layer. Linear output layer function.	Input weights are determined randomly. Bias as well as hidden layer to output layer weights are subsequently determined analytically.	L_2 criterion is minimized. System of equations is solved using Moore-Penrose generalized inverse.
Broomhead and Lowe	RBF - Linear output layer function.	Radial basis function centers are determined randomly. Weights from hidden layer nodes to output layer nodes are subsequently determined analytically.	Least squares solution is used for optimization. System of equations is solved using Moore-Penrose generalized inverse.
Pao, Park and Sobajic	RVFL - Contains direct links from input neurons to output neurons. Contains links from bias node to hidden layer nodes. Linear output layer function.	Input to hidden layer weights are determined randomly. Input to output layer weights as well as bias weights and hidden layer to output layer weights are subsequently determined analytically.	System error is minimized. System of equations is solved using a pseudo-inverse.

Table 1. Comparison of algorithms with respect to characteristics. Each row represents an algorithm and each column represents a characteristic.

REFERENCES

[1] D. S. Broomhead and D. Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, DTIC Document, 1988.

[2] M. L. Forcada. Gradient-based algorithms. `http://www.dlsi.ua.es/~mlf/nnafmc/pbook/node27.html`. Accessed: 2017-02-26.

[3] G.-B. Huang. Reply to comments on the extreme learning machine. *Transactions on neural networks*, pages 1495 – 1496, August 2008.

[4] G.-B. Huang. What are extreme learning machines? filling the gap between frank rosenblatts dream and john von neumanns puzzle. *Cognitive Computation*, 7(3):263–278, 2015.

[5] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985–990. IEEE, 2004.

[6] B. Igelnik and Y. H. Pao. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Trans.*, 6(6):1320–1329, Nov. 1995.

[7] Y.-H. Pao, G.-H. Park, and D. J. Sobajic. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing*, 6(2):163–180, 1994.

[8] W. F. Schmidt, M. A. Kraaijveld, and R. P. Duin. Feedforward neural networks with random weights. In *Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*, pages 1–4. IEEE, 1992.

[9] D. Svozil, V. Kvasnicka, and J. Pospichal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62, 1997.

[10] P. Tominc and L. B. Tominc. Some aspects of differences between l1 and l2 criteria in the linear switching regression. *New Approaches in Applied Statistics*, 2000.

[11] L. P. Wang and chunru R. Wan. Comments and replies on extreme learning machine. *IEEE Trans. on neural networks*, 19(8):1–2, Aug. 2008.

Unveiling storytelling and visualization of data

S.Arevalo Arboleda and A. Dewan

Abstract—Effective visual storytelling has the power of making a connection with the audience and here lies the importance of its study. The art of creating visual stories from scientific data is not an easy task, due to the elements that are combined in the process. It can be divided into three components to facilitate individual analysis: scientific data, visualization and a narrative representation. These components can support each other in transforming data into knowledge, in a manner that facilitates coherent flow of information without compromising on clarity. In the process of understanding storytelling, we present an example using Gapminder that will help to illustrate data visualization in an interactive way. In this paper we are interested in validating that scientific insights can be effectively imbued through storytelling. We describe different tools that enable storytelling to find its relevance in the field of visualization and tell simple stories with complex data.

Index Terms—Storytelling, Data visualization, Gapminder.

1 Introduction

"Once upon a time" is a phrase we are used to listening to since childhood. From an early age we learn about good deeds from fables and fairy tales, then, when we go to school we learn about history and the process does not stop there, we keep on listening to stories as the time goes by, only to become storytellers of our own experiences. Storytelling has been handed over from generation to generation, we are wired for communicating and learning from stories. Because stories are memorable, they can travel far and wide and more than that they can inspire.

Humans are visual creatures, we depend a lot on what our eyes tell us. Have you heard the phrase "let me picture this for you?" or "An image is worth a thousand words". These phrases tell how important it is to visualize things. Therefore statistics and numbers are more understandable and engaging when presented in a visual manner. Nowadays, there are a countless number of tools that present data visually through charts and graphs. Visualization tools have matured over time in level of sophistication and automation rendering a new way of seeing and interacting with scientifically collected data.

Scientific data tends to be complex in nature and meaning, therefore, presenting it visually has brought different challenges. Knaflic[8] highlights how we have all been victims of bad hit-and-run presentations. It has become easier to generate visual charts but difficult to communicate the most important information bits. This is primarily because information shared without a tactful manner falls out as an ordinary message.

The timeless art of storytelling is taking an important stand in the field of scientific data now, where data explorers have become the narrators of the stories that data is trying to tell. The power of narrative in scientific data lies behind transforming information into knowledge that provides better understanding of complex matters. This enables scientific data explorers to achieve the goal of creating engagement and raising awareness of the message that is being communicated. Yoder-Wise & Kowalski [17] present storytelling as "an art focused on a desire to connect with the users in a meaningful and purposeful way". This conceptualization involves one of the main purposes of data visualization, transmitting relevant information.

In this paper we focus on providing information about visualizing scientific data through storytelling and highlighting its relevance in conveying a message. We start by giving a detailed account of the components that make a visual data story. In the accounts that follow, we first narrow down the scope of our research to the practical implementation of storytelling. We describe the process stages involved in creating and delivering the story. Then, the research continues with a wider scope of the most used tools and techniques that have achieved the purpose of effective data visualization through storytelling. It is important to mention that a number of factors must be taken into consideration while creating and delivering visual data stories, hence, we describe the influence of these factors in the backdrop of the most common practical situations. In addition, examples are the best way of conveying a message, thus, we show an example on how data visualization is being used to tell a story and communicate data in a participative way.

This paper is organized as follows: Section 2 describes the key concepts and individual key components of storytelling, then, Section 3 explains the techniques involved and the influential factors in the subject. Section 4 shows a practical example using the most popular tool for presenting visual stories, Gapminder. Section 5 presents the conclusions of this paper, and Section 6 presents how the future of storytelling can be envisioned.

- *S.Arevalo Arboleda is a Msc.Human-Technology Interaction student at Tampere University of Technology , E-mail: stephanie.arevaloarboleda@student.tut.fi .*
- *A. Dewan is a MSc. Computing Science student at the University of Groningen, E-mail: a.dewan@student.rug.nl.*

2 Concept

In order to provide a better understanding on the subject, we introduce some concepts for setting the ground for the following study. In computer science, data visualization is a term often used for business intelligence and data analysis matters, thus, at first sight these terms might not seem new or hard to understand. Conversely, storytelling, is a term that is not always in the computer jargon, but stranger to the field. And here lies the importance for those terms to be clarified.

2.1 Data Visualization

Data visualization can be introduced as presenting information in a visual manner, however, its importance goes beyond that simplistic definition. SAS discussed in a global forum that data visualization enables decision making, since it can provide verifiable and reliable statistical measures to help gain understanding of complex concepts, or identify trends and predict results[12].

Now, the importance of visualizing data can be stated from different perspectives, Aparicio & Costa[1] state that data visualization turns information to an experiential level, implying a cognitive abstraction of the term. In fact, we as humans process information visually, therefore, providing organized information that can be easily understood, supports the channel of communicating a message. On the other hand, Gratz et al.[4] state the discovery of new insights as the main goal of data visualization. Combining these two approaches in data visualization, finding the appropriate medium for presenting information seems to be vital for conveying a message.

2.2 Storytelling

”Storytelling is the cornerstone of human experience”[9]. We use stories to provide information about different types of events. We do not just tell a story, we share it with our peers and we intend to produce a certain response from them. Therefore, a story can be conceptualized from different angles.

Gershon & Ward[3] present storytelling as a way of ”conveying information”, this interpretation summarizes the concept and the main purpose of the term. Specifically, they highlight that stories are more compelling and ease the understanding of facts. Added to that, Kosara & Mackinlay[9] define it as ”a sequence of steps”, where timeline is the dominant element. In fact, time provides sense and causality to the story, that is why the way the timeline is represented depends on the style being used, and the effect that is intended. For instance, a story can be told in chronological order, or backwards to present the reasons that lead to a result that was first presented. An example of this is the map and line chart showing Napoleon’s retreat from Moscow, remarking the temperatures during the retreat[16]. The style of telling a story depends on the person who is telling it, and the expected impact in the audience.

2.3 Narrative in Data Visualization

The need to present complex data in a more digestible way to reach different types of public has become of great interest. Presenting charts and graphics as separated pieces of information is no longer the best way for reaching an audience. The new trend is turning data into a story, which brings a whole new set of opportunities and challenges to be discussed when considering the art of storytelling.

When telling a story Mackinlay et al.[9] insist upon the importance of adding a little drama when presenting a story. The reason for this is making a bigger impact, because of the intriguing factor, unlike just showcasing data and facts. Moreover, they mention that stories explain reasons when data presents a description of an event. In this context, combining data with stories just makes sense in order to convey a message.

Furthermore, Mackinlay et al. present a structure for successful stories that goes beyond the regular structure that we all have learned at schools. Figure 1 presents an arc to illustrate the flow of a story that consists of rising action and conflict, presented logically and fluently to lead to a conclusion. [9]

Besides stating a general structure, Segel & Heer[14] add two different approaches to storytelling: ”author-driven and reader driven”. The first one does not contain interactivity, follows a linear structure and sends a specific message, while the second one is more dynamic allows data exploration and invites to different interpretations. These approaches present different schemas of visualization divided in: a martini glass structure, interactive slideshow and a drill-down story. First, a martini glass structure presents findings at the beginning and then opens up to discussion as it develops. Then, the interactive slideshow allows exploration across the whole process of data visualization. Finally, the drill-down story presents a general topic and allows specific exploration of a section in the data.

Choosing an approach for telling a story through visualizing data remains vital for transmitting a message and reaching the audience in the way we expect.

3 Realization

In this section we will give an outline of the methodologies for implementing the ideas of creating and delivering visual stories. That includes the traditionally adopted approach, as well as the most novel techniques. Based upon these tools and techniques we describe external factors that influence the implementation and the outcomes at every stage.

3.1 Process

Journalists, wanting to bring a change and have an influence on public opinions were the early adopters of storytelling through visualization. They would share their researched facts in a relevant and sequential way for the readers, to highlight important connections between the facts and reach a wider audience. [5] Data journalism literature inspired Lee et al. [10] to formulate a working model for the main roles and activities involved in turning raw data into a visually shared story. The model has three stages: exploring data, authoring a story, and presenting a story. The process may occur as linearly ordered or may occur with some looping between the stages. The flexibility to control the order of sequence will be discussed later. For the sake of simplicity, we describe the stages in a linear order.

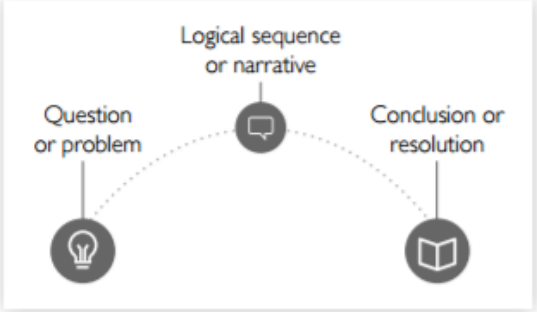


Fig. 1: Mackinlay et al.[9] representation of the story arc

Exploring data involves analyzing the data and summarizing the main characteristics of a dataset. The data analyst uses standard statistical tools and techniques to pinpoint trends, correlations and patterns in data sets. It is important to mention that an analyst is often not the same person as the one who makes decisions, therefore, analysts are required to prepare charts based on a collection of the chosen data excerpts.[9]

Authoring involves weaving a narrative around the findings from data exploration. A sequence and a plot are the significant parts of making a story. The activities span out in the form of ordering, establishing logical flow of ideas and highlighting important messages. The process may be implemented sequentially or through multiple iterations.[10] In most cases, the author has to switch back to the exploration stage to gather more insights or evidences. The result of authoring is an overall plot which connects sections of the story in terms of time, cause and effect and patterns.

Presentation involves selling the data story to everyone in the audience. The editor completes the story material provided by the author and builds a presentation that is ready to be showcased. The presenter delivers this information through the presentation in different settings and with different audiences. Each scenario has different requirements for the the way the presentation is conducted. We elaborate these requirements and scenarios when we discuss the external factors. In their paper, Kosara & Mackinlay [9] look at the effectiveness of a presentation as an evaluation parameter. They provide an overview of features that can make a presentation memorable, thereby enhancing its effectiveness. The features include audience interaction and embellishments such as annotations, highlighting, arrows, etc.

3.2 Tools and techniques

The visual data storytelling process described above is implemented with many of the existing tools meant to help users deliver a compelling story and present data with more sophisticated interactions. Gapminder, exemplified by Hans Rosling, stands out as having the most effective interface. A graph produced from Gapminder can store a wide variety of indicators along a timeline.[2] The narrative starts with the simpler indicators on the X axis, then moves on to more complex indicators along Y axis, and finally, uses bubble sizes as the third indicator. The presenter is able to bind the animation with his narrative wherein he can play/stop anytime, move back-and-forth to examine a specific period of time, zoom in to mark observations about general trends over time.

More research is being done to enable authors with little or no programming skills to create visual stories. Hullman et al.[6] focus on linear, ’slideshow-style’ presentations to provide a deeper understand-

ing. The slideshow's controls allow the user to move back and forth between steps, and the content is structured like a dialog. In a project at UW Interactive Data Lab, Satyanarayan et al. [13] developed Ellipsis, a system that combines a domain-specific language (DSL) for storytelling with a graphical interface for story authoring. It is a significant and relevant advancement along the lines of what Lee et al. [10] suggest in their paper regarding the need of developing more sophisticated methodologies that support the entire visual data storytelling process. Another tool is Tableau Story Points[11], a framework where users can tell stories with data in the same tool that is being used to analyze data. It also allows navigating through data visualizations sequentially in series of interactive dashboards.

We mentioned earlier how the order of sequence of stages can be controlled with more powerful techniques. The tools enlisted so far focus more on back and forth transitions between exploration and authoring, than on allowing other transitions as shown in Figure 2. The lack of a back-link from the final curated story to the initial exploration stage and the underlying data makes it difficult to reproduce and verify the findings explained in a presented story. Gratzl et al.[4] present a more recent technique to control the sequence of stages. They introduce the idea of provenance data and also propose the CLUE (capture, label, understand, explain) model to bridge the gap between data exploration and presentation of stories. Provenance of the state of a visual exploration process refers to all information and actions that lead to the final story. The CLUE framework rests upon a provenance graph that contains all actions performed during the exploration, including exploration paths that led to findings as well as the dead ends encountered by the analyst. Existing visual exploration tools can use the CLUE library framework to reproduce the original analysis and to launch new explorations. Figure 3 shows that the CLUE model enables process development in any sequence desired by the user depending on the objective. Any of the three stages can be the entry point and the numbers indicate the order sequence in which the stages are visited by the user.

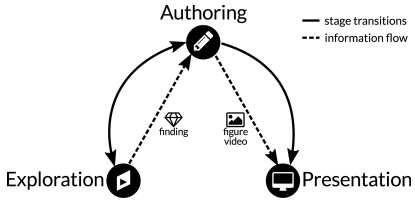


Fig. 2: Traditional workflow with solid edges for transitions between stages and dashed edges for information flow. [4]

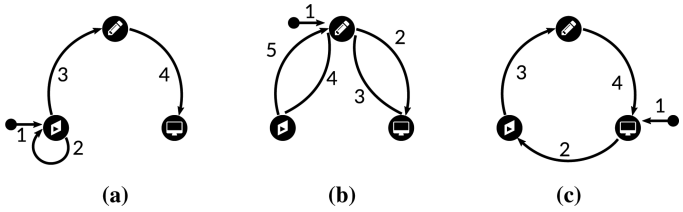


Fig. 3: Examples with different entry points in the CLUE model.[4]

Furthermore, Gratzl et al. give a detailed account of the components of the CLUE framework, its provenance and story view. Provenance refers to the historical record of every process involved in delivering the current state of narrative. The provenance view provides visualization of the graph of origin in a way that it can be zoomed in to any degree of detail, at any point in the history. The provenance graph has four node types: state, action, object, and slide as shown in Figure 4. Actions are associated with operations such as create, update and remove. They transform one state to another by the means of these operations. A state consists of all objects that are active at this point of the exploration. An object can be any of these types: data, visual, layout, logic, and selection. A slide points to a state along with annotations and descriptions explaining the state. The Story View visualizes the elements of the story. Exploration, authoring and presentation modes focus on different perspectives, therefore, only the relevant information and visual elements relevant to the current mode are shown in the provenance and story views. We link the work done by S. Gratzl et al. with the kind of research future scope suggested by Lee et al.[10] in their paper.

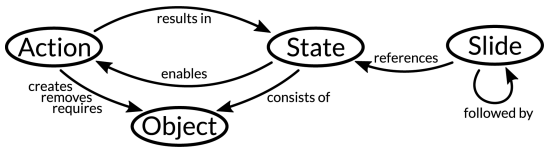


Fig. 4: Components of the provenance graph model[4]

3.3 Choosing the appropriate tool for telling visual data stories

A set of external factors may impact the steps in the process of telling a story. As the tools and methods become more mature and holistic, we need to start focusing on basing the presentation structure upon different settings and with different audiences. Every scenario holds different requirements as to how much and which data is shown, the techniques used, the objective of presentation, the amount of interaction anticipated, etc.[9] We will first describe the practical scenarios, their requirements and a suitable choice of tool/technique. Towards the end of the section we enlist the factors separately once again.

Scenario 1: The presenter may be showing findings to a room full of people who do not have previous knowledge of the information that is displayed. This is the case when a story is required to raise awareness and create interest in a topic that the audience may not otherwise be aware of. These stories can be entirely self-running (videoclip), or require the user to click through (slideshow), or provide limited means of interaction with the audience. The focus is on explaining the highlighted points in sufficient detail for the viewer to understand the story based on real facts and data. Much time is spent on achieving a sequence which provides an engaging teaser without requiring much interaction, and also allows a deeper and meaningful connection with the story.[9] This scenario is an exceptional case wherein static images may be used as much as animations and visualizations, because static views at the initial stages avoid the situation of any possible information dump for the novice viewers. Once a state of familiarity with the topic is achieved, animations can be used to allow the viewer to dig deeper into the data, or find out how to relate to it. Often, a presentation of this nature is created once without the need of switching back to exploration or authoring modes. A linear slide-show presentation style is a suitable choice for this scenario.[6]

Scenario 2: The presenter may be giving a routine presentation in front of an audience that has some background knowledge. This calls for presenting with an accurate judgment of the extent of trade off between audience engagement and focus.[9] In the same paper, Kosara et al. highlight the trade-off between interaction and focus. They consider focus as one of the evaluation parameters and evaluate effectiveness of different presentations with varying degrees of interaction. When the presenter pauses the story development for responding to questions, a certain section of audience is likely to get distracted from the story. Therefore, presentations in such scenarios are focused on building smaller blocks of information, so that the effects of any possible distractions are minimized. Gapminder charts are the most suitable choice for these scenarios.

Scenario 3: The presenter may be convincing senior leaders or peers about newly drawn critical points on a subject matter that everyone in audience understands well. This kind of presentation is often limited to a smaller group of individuals who could be decision makers or policy evaluators. Consequently, this scenario demands a maximum level of interaction between the presenter and the audience. The presentation tool must be more flexible to answer questions that

come up during the presentation. The presenter may receive pertinent or sometimes impertinent questions about things that may or may not be a part of the narrative. A well-presented story will lead to less questions or clarifications from the audience. A story of this nature serves two-fold purposes: it becomes a medium for disseminating information and it collects additional insights on the presented data.[9] The advanced CLUE model is a suitable choice for such scenarios as it provides more flexibility in an holistic manner.

Based on the different possible situations discussed, a set of factors must be must be kept in mind. Across every stage of exploration, authoring and presentation, it is a matter of utmost importance to consider the targeted audience, and sometimes the size of the audience. The next concern should regard the setting and context to determine the way a visual story will be presented. This has mostly to do with the size of the presentation venue and allotted time duration.

Furthermore, the medium is another factor that influences how the story material will be created, presented and consumed. For instance, the pace and flow of a narrative will be different for a static image than for an interactive visual animation. Except for a narrative for a novice audience, static images and videos are not a suitable choice for visual data stories. Static images do not convey much information about the exploration findings and videos are difficult to create, edit and broadcast more information than required in a shorter time frame.

4 Unfolding an example through Gapminder

After gaining awareness of the most common tools used for data visualization, we present an example using Gapminder. We intend to examine the different elements in the tool and how storytelling can be used to present information.

Gapminder presents itself as a "fact tank" that provides information that can be easily understood. It presents a combination of statistics gathered from different reliable sources and showcases the information in an animated timeline, its main goal is to "promote a fact-based worldview everyone can understand"[2]. Gapminder communicates facts in a dramatic and engaging manner, keeping an audience interested while providing knowledge. It highlights the importance of creating an impact and raising interest in topics of global concern. Presenting statistics visually gives numbers a meaning and expands the understanding of information. For instance, let us take a look at "Life expectancy vs Income per person". We can present a bar chart with that title, it can be divided by country and it will look something like what is presented in Figure 5. It looks like a standard chart with some information about how life expectancy has evolved, however, there is nothing that impacts or engages the audience. It can be said that it just looks like some data presented in bars.

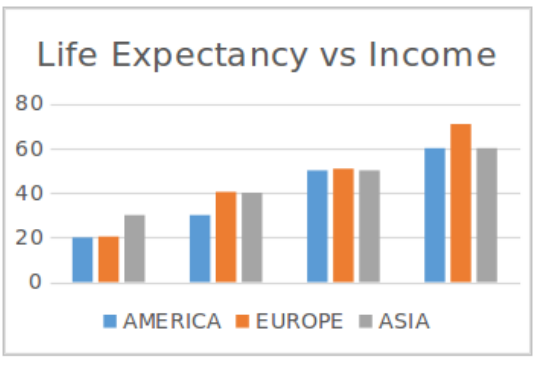


Fig. 5: A representation of life expectancy in a bar chart. The Y axis shows the life expectancy in years and the X axis the income. Each bar represents the continent that is being analyzed and how it changes along time.

Now, let us make some modifications in this representation of data. To begin with, the title can be changed into something more dramatic that will captivate the audience. It can be presented as: "Can living in a wealthy country make you live longer?" Elaborating a *question* for introducing information not only follows the structure presented by Mackinlay, but also adds some intriguing factor to it, which evokes emotions in the audience. And it is with the introduction of a story that we are creating some type of drama from the very beginning.

Once we have the interest of the audience, it is important to maintain it by presenting data in a more dynamic and appealing way. Figure 6 shows how Gapminder presents data. The data is collected from different sources: the human mortality database from the University of California, the United Nations population prospects, the Human-Life table database, national statistics agencies, experts' information and scientific literature.[7] As can be seen, Gapminder graph accommodates a great amount of information about life expectancy, income and time, all condensed in one image which creates an impact that goes beyond numbers.

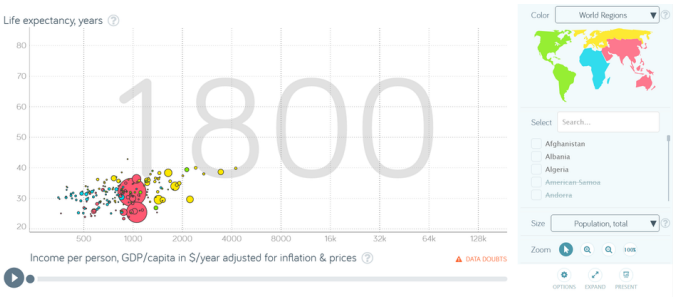


Fig. 6: Life expectancy vs income data expressed in Gapminder, the colors represent each continent and the bubbles represent each country, the size of the bubble shows how big the population is and the start for the information year is 1800. X axis presents the income and the Y axis shows the life expectancy in years [2]

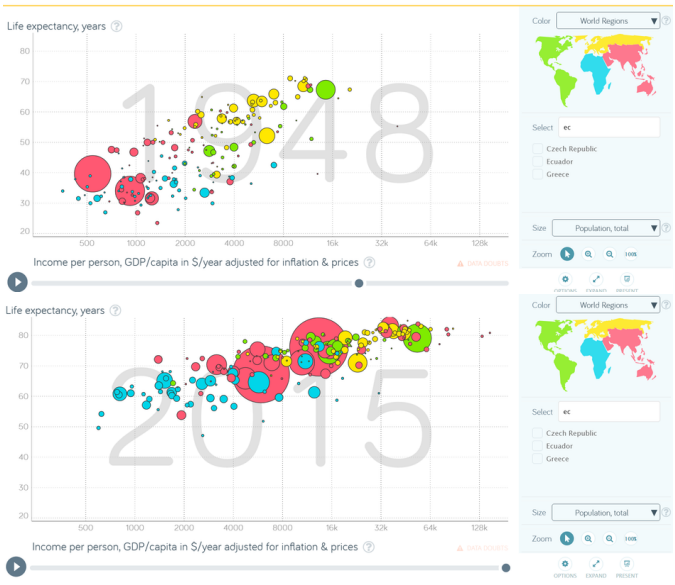


Fig. 7: Representation of data in a timeline with Gapminder[2]

Moving on to Figure 7, it is imperative to describe some of its elements. The information displayed on the right side allows users to select a specific country. The play bar below the chart creates an interactive flow between 1800 and 2015. The timeline adds the second element from Mackinlay, a *logical sequence.*

Finally, the last element is a conclusion, which can be expressed as follows: *Apparently humanity has come a long way to extend their time on earth, we have achieved to extend our life expectancy*

through economic wealth. However, we have also increased inequality amongst us. This invites people to reflect on the information that was presented and draw their own conclusions. That is just one way how the information can be interpreted in the images.

All in all, in general terms it can be seen that Gapminder presents information in a way that catches the attention of the public and communicates the message in a visually attractive depiction of data.

4.1 Data exploration through Gapminder

Allowing users to explore data is the most powerful tool for diversifying the narrative. It is often said that beauty lies in the eye of the beholder, in this case, stories lie in the eye of the beholder.

By way of illustration, following the example of "Life expectancy vs Income per person" in Figure 8. What was going on in 1941 in those countries? Well, the world was struck by WWII and Hitler was focused on exterminating the Soviet Union as a military power. Thus, in December 1940 Germany invaded the western Soviet Union, actual territories of Ukraine, Belarus and northwest Russia, leaving millions of casualties as a consequence of the conflict.[15] Thereupon, the life expectancy reduced considerably in those countries when compared to the rest of the world.

It can be seen that special circumstances triggered an abnormality in the data, that can be explained by determining events in history, such as war.

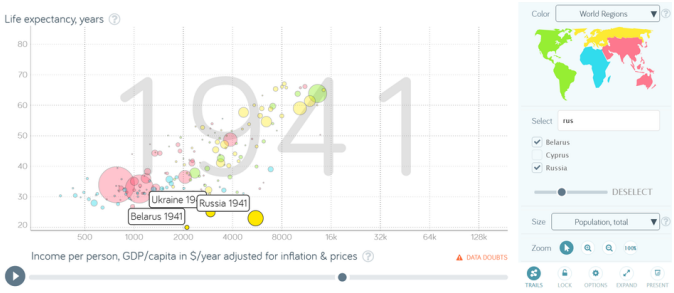


Fig. 8: Life expectancy and income in 1941, focusing on Russia, Ukraine and Belarus.[2]

5 Conclusion

Telling stories in a visual manner intends to captivate audiences and invite them to tell their own stories using the showcased data. It encourages the audience to draw their own conclusions based on some specific parts of data. Moreover, it can be said that it empowers the audience with the presented information in order to transform the public into storytellers.

On the other hand, there are different tools that can be used for visual storytelling, each one of them specializes in a specific purpose depending on the message that is intended to be communicated.

Storytelling goes beyond creating engagement and presenting information in an appealing manner, it sets the first step towards a course of action, it invites the audience to take the next step when the message has been delivered. All in all, through this paper we showed a number of ways to incorporate the ideas that lie in the intersection zone of technical, social and interpersonal needs of showcasing information.

6 Future Work

The future of storytelling seems promising, and it can include more elements that will help to convey a message, create awareness and engage the audience into a new experience of analyzing data. Therefore, different elements can be added into storytelling, one of them can be augmented reality. It can add realism to the data presented and show possible future scenarios, which can result in transmitting a clearer message and creating a connection with the audience. Added to that, augmented reality can bring interactivity to the already existing storytelling tools, plus it can invite people to relate to the data, and finally spread the message.

It is often the case that the scientific computing pipeline starts with the modeling phase, followed by a simulation and visualization phase. Therefore, visualization scientists are required to share the set of final results back with a larger community. We think for further research in this direction, there is a scope of common problem-solving environment (PSE). This would not interfere with rigorous activities of other components which operate in their independent development environments. For example, if a Gapminder graph highlights such results that address a certain problem, then, in a common PSE forum the results can be used to address a type of problem that appears to be universal across other components of modeling, simulation and visualization. Therefore, as in the real world, the stories can travel far and wide in the scientific computing world too.

References

[1] M. Aparicio and C. J. Costa. Data visualization. *Commun. Des. Q. Rev.*, 3(1):7–11, Jan 2015.

[2] G. Foundation. gapminder.. `http://www.gapminder.org/`. Accessed: 01-Mar-2017.

[3] N. Gershon and W. Page. What storytelling can do for information visualization. *Commun. ACM*, 44(8):31–37, Aug 2001.

[4] S. Gratzl, A. Lex, N. C. N. Gehlenborg, and M. Streit. From visual exploration to storytelling and back again. *Comput. Graph. Forum*, 35(3):491–500, Jun 2016.

[5] J. Gray, L. Chambers, and L. Bounegru. *The Data Journalism Handbook*. OReilly Media, 2012.

[6] J. Hullman, S. Drucker, N. H. Riche, B. Lee, D. Fisher, and E. Adar. A deeper understanding of sequence in narrative visualization. *IEEE TVCG (Proc. InfoVis)*, 19(12):2406–2415, 2013.

[7] K. Johansson and L. Mattias. Documentation for life expectancy at birth (years) for countries and territories. *Gapminder Found.*, page 10, 2014.

[8] C. N. Knaflic. *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley, Nov. 2015.

[9] R. Kosara and J. Mackinlay. Storytelling: The next step for visualization. *Computer (Long. Beach. Calif).*, 46(5):44–50, May 2013.

[10] B. Lee, N. H. Riche, P. Isenberg, and S. Carpendale. More than telling a story: A closer look at the process of transforming data into visually shared stories. *IEEE Computer Graphics and Applications, Institute of Electrical and Electronics Engineers*, 35(5):84–90, 2015.

[11] J. Mackinaly, R. Kosara, and M. Wallace. data storytelling using visualization to share the human impact of numbers. Technical report, Tableau Software, 2014. Accessed: 28-Feb-2017.

[12] SAS. Data visualization: What it is and why matters. `http://www.sas.com/en_us/insights/big-data/data-visualization.html`. Accessed: 27-Feb-2017.

[13] A. Satyanarayan and J. Heer. authoring narrative visualizations with ellipsis, 2014.

[14] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1139–1148, Nov 2010.

[15] D. Stahel. operation barbarossa and germanys defeat in the east, 2009.

[16] E. R. Tufte. *The Visual Display of Quantitative Information*, volume 4. Graphics Press, Cheshire, USA, 2001.

[17] P. S. Yoder-Wise and K. Kowalski. The power of storytelling. *Nurs Outlook*, 51(1):37–42, 2003.

The ideal architecture decision management approach

Alexandra Matreata and Petar Hariskov

Abstract—The management of architectural design decisions represents a critical part in the life cycle of any product. While an erroneous decision is easily reversible during the architecting phase, it can lead to disastrous effects at a later moment and could even cause the failure of a project. Architecture decision management deals with this issue by helping architects trace back their decisions, justify them in relation to possible alternatives and evaluate them. Furthermore, it allows the tracking of these decisions to specific, concrete parts and components of the system. However, the effort, of using architecture knowledge and more specifically, architecture decision management during the design phase of a system is usually perceived as too high in comparison to the benefits it brings. In this paper we aim to compare several different approaches(theoretical and experimental techniques, tools for tabular or structural visualization, methods for concurrent or after the fact documentation) for Architecture Decision Management(ADM) and provide specifications which would be required for an ”ideal” ADM approach. We further extracted a description of an approach for architecture decision management able to balance the two contradictory forces described above: cost of implementation vs. usage and benefits. The results also provide a list of common liabilities in the compared approaches indicating points for future research.

Index Terms—Architecture decision management, architecture design decision, architecture knowledge management

1 Introduction

The last decade has brought a shift in the way in which software engineering is perceived and in what are considered to be the main responsibilities of a software architect. According to this new view, at the core of Software Architecture, lies the principle of considering the architect as a decision maker instead of someone drawing boxes and lines [1]. Architects have grown to have considerable more responsibilities and their decisions determine the overall structure, behavior and quality of a software system [2].

Due to their relatively newly discovered importance, architecture decisions have been the focus of many studies in recent years and considerable effort has been concentrated towards capturing, sharing and explaining the rationale behind them. A definition of architectural design decisions is given in [3]: ”A description of the choice and considered solutions that (partially) realize one or more requirements. Solutions consist of a set of architectural additions, subtractions and modifications to the software architecture, the rationale, and the design rules, design constraints and additional requirements.”.

Despite the numerous advantages brought by a thorough and detailed representation of the design decisions taken in the architecting phase of a project(e.g. increase in maintainability, evaluation of decisions with respect to alternatives), their documentation is often incomplete and out of date [1]. The reason for the lack of practical implementation of architecture decision management despite its theoretically proved value is mostly justified by the highly time consuming nature of the process.

Figure 1 displays a graphical overview of the architecting process from the perspective of design decision making. Given the complexity and high amount of iterations needed for this process, ADM is perceived as overhead and is frequently omitted.

In order to facilitate the work of architects and decrease the amount of time needed for documenting and justifying their decisions, several solutions have been proposed over the last years. These solutions are divided into two main categories: theoretical techniques(e.g. quality attribute scenarios, architectural patterns, view-based documentation techniques and scenario-based evaluation methods) and experimental techniques(e.g. simulation, prototyping and scenario-based methods with explicit stakeholder involvement) [5].

In this paper, we aim to perform a comparison between different existing tools and approaches for architecture decision management and determine which of their attributes would most likely be associated with an ideal method. The paper is structured as follows: in section 2, the method and criteria used for the selection, comparison and extraction of significant attributes will be described, section 3 will list and explain the approaches, tools or methods which were compared, section 4 will outline the results obtained, namely the most significant characteristics common to a majority of the studied approaches and what liabilities each of them presents, section 5 will present the conclusions derived from the comparison study.

2 Method

This study compares 4 different approaches for architecture decision management. In order to be able to spread our comparison study on a broader section of this domain, we chose one approach from every main category or division, while taking into consideration the uneven distribution of research into these categories. The divisions taken into consideration for this comparison study are: experimental versus theoretical techniques; documentation synchronized with the architecting phase or done in a recursive manner; different types of models for documentation.

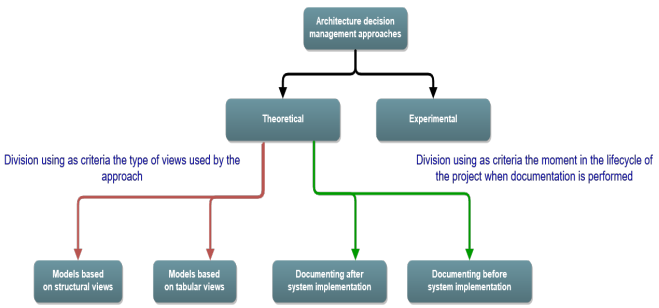


Fig. 2. Divisions of the selected architecture decision approaches into categories

Figure 2 displays the division of architecture decision management approaches in the main categories taken into consideration for this study. It is to be noted that the division criteria used for this classification are not the only criteria available. Decision management approaches cover a very broad area of diversity and can be categorised in multiple ways. The different criteria for dividing the approaches which have been taken into consideration for this study correspond to what the authors perceived as important and often used criteria. For further research, we suggest a comparison study taking into consideration more criteria and a broader scope of observation.

A first division of the approaches compared in this study, as can be observed in figure 2, categorises them into theoretical and experimental. As described in [5], this line of division is an important criteria for decision management approach categorisation. As observed in figure 2, the theoretical branch can be further divided with respect to the way in which the information is presented to the user(graphs, views, tables, etc.) or by taking into consideration the moment in the life cycle of the

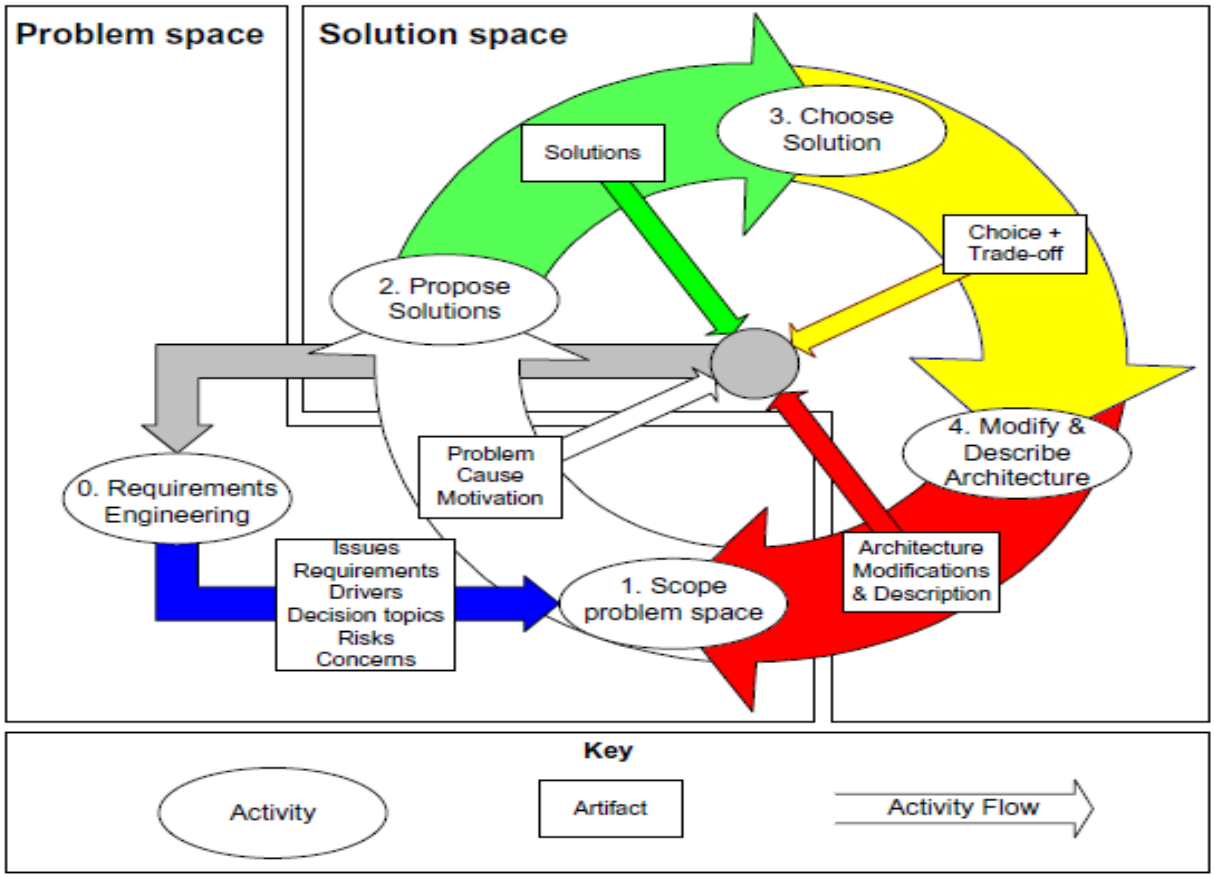


Fig. 1. The architecting process from an architectural design decision perspective [4]

project when the documentation is performed(documenting before or after the fact).

After choosing the criteria, we proceeded with choosing relevant approaches which would fit into one or more categories presented in the figure above. The approaches were chosen by taking into consideration two main aspects: their relevance with respect to one or more categories obtained from applying the division criteria presented above and their popularity in their area of expertise(e.g. measured in number of citations). The selected approaches and their connections to different categories are listed below:

1. **Decision architect [2]: theoretical technique; used for documentation before the fact; uses both tabular and graph-oriented views**
2. **Architectural prototypes [5]:** experimental technique
3. **ADDRA(Architectural Design Decision Recovery Approach) [4]:** theoretical technique for documenting after the fact
4. **Graph-oriented model [6]:** theoretical technique exclusively oriented towards structural views

Each approach was evaluated in parallel by the two authors of this paper. The selected approaches were compared based on a set of 10 criteria extracted from [7]. Since the questions proposed by [7] form the base of a comparative study which focuses on architectural knowledge management in general, a few adjustments were needed in order to better model the criteria to our more specific domain of interest.

Table 1 shows a list of the personalised queries used for this study. Three possible answers were taken into consideration for each question when analysing each approach: 'yes', 'no' and 'to some extent'. Each response was given a score of 1, 0 and 0.5 respectively. The final score of an approach was determined as the average value of the summed scores obtained from the answers given by each author.

ID	Question
Q1	Does the approach support and manage dependencies between different types of architectural entities (such as decisions, requirements and tactics)?
Q2	Does the approach support change impact analysis?
Q3	Does the approach support architects to use and produce well-described design decisions during architecture synthesis stage?
Q4	Does the approach support architecture evaluation?
Q5	Does the approach support architecture maintenance?
Q6	Does the approach support the personalization of knowledge representation based on different users preferences and profiles?
Q7	Does the approach support the integration with other software engineering and knowledge engineering tools and knowledge repositories?
Q8	Does the approach support collaboration between distributed software development teams?
Q9	Does the approach support a thorough analysis of the design decisions?(such as providing solutions for capturing detailed description, traceability, alternatives)
Q10	Does the approach allow for an easy communication of design decisions captured in an early phase towards the implementation phase?

Table 1. Quality assessment criteria

The scores and justified answers given for each approach are available in tables 2 to 5.

3 Approaches

According to the main categories displayed in figure 2, we selected four approaches for architectural decision management as follows:

1. **Decision architect:**

   Decision Architect is a tool developed as a collaboration project between RuG and ABB and described in the article [2] with the purpose to surpass the limitations of already available tools for capturing Architectural Decisions. The tool is build upon five viewpoints for each of the stakeholder concerns. The *decision relationship* provides information about which technological decisions influence the usage of which tools. The *decision detail* is concerned with solutions and arguements for a decision. The *decision chronology* shows decisions as they are taken from a time line point of view, *decision forces* shows the influences over the architect for choosing alternatives and *stakeholder involvement* involves stakeholders through the decision-making process.
   As the paper's [2] outcome suggests, the tool has been found to be useful for architects as it improves the documentation quality and productivity of architects. Due to the tool's ease of use it has not been perceived by the studied architects as an obstacle but rather as an actual practical tool in contrast with most alternative tools.

2. **Architectural prototypes:**

   As defined in [5], the understanding of the term "Architectural Prototype" is a process of using code to outline architectural concerns in any system. In order to identify a prototype as architectural, [8] has given five specifications. A prototype is architectural if it is focused around the idea of *learning*, the prototype has to outline the *architectural risks*, present the implications of different decisions derived from the *quality attributes* while not providing any *functionality* and lastly address *knowledge transfer*. Constructing a decision tree for each quality attribute could provide invaluable information for future changes to an architecture, the trade-offs that will result in the change, all the following changes to the architecture that will follow as well as the size and complexity of such a change.

3. **ADDRA(Architectural Design Decision Recovery Approach):**

   Paper [4] proposes an approach for modeling and documenting design decisions from an already implemented system and architecture. This process, defined as architectural decision recovery, is a complex and iterative process, which ADDRA divides into five main steps: "define and select releases, recover detailed design, recover software architecture views, determine architectural delta, recover architectural design decisions [4]".

   Using Nonakas paradigm for knowledge creation [9], the steps cited above help recover tacit architectural knowledge and dinamicaly transform it into appropriate organizational knowledge.

   The paper also discusses important limitations of the approach and points out issues in need of future research. The biggest limitation discussed is represented by the assumption that the architect responsible for building the architecture in the first place is also in charge of the recovery process.

4. **Graph-oriented model:**

   Paper [6] proposes a solution comprised of an UML meta-model and a graph model combined in a theoretical technique for architectural decision documenting and reuse. The solution was tested for a documentation process simultaneous to the architecting phase.

   The UML meta-model, described in more detail in one of their previous works, [10], revolves around three main processing steps: decision identification, decision making, and decision enforcement. The meta-model proposes a simple, basic mandatory architecture for the users to follow when documenting decisions. It is also categorized as "machine readable and translatable into other specifications, e.g., into Web services contracts and relational database schemas" for enabling tool support in decision modeling.

   This meta-model is accompanied by a decision tree built in the form of a directed acyclic graphs with several types of nodes and edges which help better demonstrate the relationships and dependencies between different elements and better visualize the architectural organization.

   Their research is validated by analysis, implementation, experiment and industrial case studies. The dependency modeling proposed is also verified against an existing taxonomy [11].

4 Results

In this study we have compared four different approaches for architectural decision management. Each approach can be mapped onto one of the categories obtained by applying the different division criteria described above. Given the broad scope of the domain, the approaches compared cover a large and diverse area and are highly dissimilar to one another. While keeping this in mind, we tried to produce a list of common liabilities between the approaches from the answers given to the list of quality assessment criteria.

As can be observed from tables 2 to 5, all approaches received a similar, relatively high total score except for ADDRA, which received a significantly lower score. The reason for this could be explained from the point of view of the moment in the life-span of the system for which these approaches were conceived. Although all approaches compared in this study can be used for documenting decisions "after the fact", with the exception of the ADDRA approach, all the case studies presented by their authors correspond to a testing of the approaches in the context of a documentation process simultaneous to the architecting phase.

This suggests that a documentation of the decisions taken during the architecting phase itself produces better results in the capturing of architectural knowledge and helps provide a better base for architecture evaluation and the future maintenance process.

An interesting result which can be observed from the score tables is that question Q10 was fully satisfied only by the architectural prototyping approach. Thus, an experimental technique would seem to be better at displaying or communicating the knowledge captured, even for the case of inexperienced users. On the other hand, question Q2 was satisfied fully or at least to some extent by all theoretical approaches while the experimental one failed for this test. It would appear, from these findings, that theoretical techniques are better at managing changes taking place during the architecting phase and their overall impact on the system's performance. A combination of both techniques would appear to be beneficial and to bring a balanced improvement in the decision management process by combining a thorough documentation and evaluation of architectural knowledge with a powerful method for knowledge sharing.

With regards to the quality of the traceability of the decisions(questions Q1 and Q9) and their documentation(Q3), Decision architect and the graph oriented model obtained the highest scores, while ADDRA performed the worst. A reason for this would be that traceability is easier to be managed in an early phase of the project due to the effects brought by knowledge vaporization in later stages. Also, a structural view performs better at representing relations between architectural entities, while tabular views are more oriented to describe specific attributes of the decisions in larger amounts of detail. A combination of both types of views is thus suggested and would seem to perform best in describing and tracing back the decisions, as observed with Decision Architect.

Considering the support for evaluation of decisions(questions Q2, Q4), ADDRA received once more the lowest score, while Decision Architect proved to address this issue very well. Also, when consider-

ing experimental against theoretical techniques, we can observe that theoretical techniques are better at evaluating the impact of decisions on the whole system, while experimental techniques help test a singular decision in a narrow context, while providing more precise results.

Finally, regarding the compatibility of the approaches with other different tools and their support for collaborative processes(questions Q6, Q7, Q8 and Q10), the scores obtained by the approaches were similar, slightly lower for the case of ADDRA. We thus suggest that further research should be directed in this sense in order to achieve a more easily integration of the approaches with existing techniques.

5 Conclusions

The results obtained in this study show some interesting findings and help prove and strengthen the validity of some findings presented in similar studies regarding the attributes and qualities that an ideal decision management approach should poses.

First, documentation and knowledge capturing processes are best to be performed simultaneously to the architecting phase and not after. This helps increase several important attributes of the systems and their architectures such as maintainability, traceability, etc.

Second, although most of the approaches compared and analysed in this study received a relatively high total score, each of the categories are lacking points in important criteria and show that there is need for further research in the domain in order for the tools and approaches provided to fully meet the expectations of users.

With regard to the techniques analysed by this study, we finally propose as an ”ideal” approach a combination of both theoretical and experimental techniques used for a documentation process simultaneous to the architecting phase.

References

[1] R. Capilla, A. Jansen, A. Tang, P. Avgeriou, and M. A. Babar, “10 years of software architecture knowledge management practice and future,” 2015. [Online]. Available: http://www.cs.rug.nl/ paris/papers/JSS16c.pdf

[2] C. Manteuffel, D. Tofan, P. Avgeriou, H. Koziolek, and T. Goldschmidt, “Decision architecta decision documentation tool for industry,” 2015. [Online]. Available: http://www.cs.rug.nl/paris/papers/JSS16b.pdf

[3] J. S. van der Ven, A. G. J. Jansen, J. A. G. Nijhuis, and J. Bosch, “Design decisions: The bridge between rationale and architecture,” 2006. [Online]. Available: https://www.researchgate.net/publication/239546223$_{1}$ $_{D}$ $_{e}$ $_{s}$ $_{i}$ $_{g}$ $_{n}$ $_{D}$ $_{e}$ $_{c}$ $_{i}$ $_{s}$ $_{i}$ $_{o}$ $_{n}$ $_{s}$ $_{T}$ $_{h}$ $_{e}$ $_{B}$ $_{r}$ $_{i}$ $_{d}$ $_{g}$ $_{e}$ $_{b}$ $_{e}$ $_{t}$ $_{w}$ $_{e}$ $_{e}$ $_{n}$ $_{R}$ $_{a}$ $_{t}$ $_{i}$ $_{o}$ $_{n}$ $_{a}$ $_{l}$ $_{e}$ $_{a}$ $_{n}$ $_{d}$ $_{A}$ $_{r}$ $_{c}$ $_{h}$ $_{i}$ $_{t}$ $_{e}$ $_{c}$ $_{t}$ $_{u}$ $_{r}$ $_{e}$

[4] A. Jansen, J. Bosch, and P. Avgeriou, “Documenting after the fact: Recovering architectural design decisions,” 2007.

[5] H. B. Christensen and K. M. Hansen, “An empirical investigation of architectural prototyping,” 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1663921

[6] O. Zimmermann, J. Koehler, F. Leymann, R. Polley, and N. Schuster, “Managing architectural decision models with dependency relations, integrity constraints, and production rules,” 2009.

[7] A. Tang, P. Avgeriou, A. Jansen, R. Capilla, and M. A. Babar, “A comparative study of architecture knowledge management tools,” 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121209002295

[8] J. Bardram, H. Christensen, and K. Hansen, “Architetural prototyping: an approach for grounding architectural design,” 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1310686/

[9] I. Nonaka, “A dynamic theory of organizational knowledge creation,” p. 1437, 1994. [Online]. Available: http://pubsonline.informs.org/doi/abs/10.1287/orsc.5.1.14?journalCode=orsc

[10] S. Abrams, B. Bloom, P. Keyser, D. Kimelman, E. Nelson, W. Neuberger, T. Roth, I. Simmonds, S. Tang, and J. Vlissides, “Architectural thinking and modeling with the architects workbench,” 2006. [Online]. Available: http://domino.research.ibm.com/tchjr/journalindex.nsf/e90fc5d047e64ebf85256bc80066919c/862b3f047fea67fa852571bc007f4eb4!OpenDocument

[11] P. Kruchten, P. Lago, and H. van Vliet, “Building up and reasoning about architectural knowledge,” pp. 43–58, 2006. [Online]. Available: http://link.springer.com/chapter/10.1007/11921998$_{8}$

A Appendix - Result scores

Approach	Question	Score	Justification
Decision architect	Q1	1	The decisions are described in relation to requirements and the relationship viewpoint provides traceability for related decisions.
	Q2	1	The chronology viewpoint allows the documentation of changes in the decisions taken over time and by following the relation between a certain change and its corresponding entries in the other 4 views, the architect is able to evaluate the impact of that specific change on the system.
	Q3	1	The decisions are described using 5 different viewpoints.
	Q4	1	The forces viewpoint provides an evaluation of each decision and its alternatives according to different concerns and forces.
	Q5	0.5	The chronology viewpoint allows a description of the evolution of decisions over the architecting phase which can provide a good traceability and help with the maintenance process.
	Q6	0.5	The user needs to follow the guidelines for the 5-view model. However, none of the views are mandatory and any combination of them is allowed.
	Q7	1	The approach supports integration with any modeling tool providing a plug-in architecture.
	Q8	0	Information Q8 (distributed collaboration) is missing so it has been given the score of 0
	Q9	1	Decisions are thoroughly described through the use of the 5 different views.
	Q10	0.5	The approach supports a documentation of decisions both during the architecting phase and in retrospection. In addition, the chronology view helps keep track of the changes and evolution of the decisions from early phases towards implementation. However, a thorough documentation of the decisions using all 5 views demands a relatively large amount of effort.

Table 2. Quality assessment scores - Decision architect

Approach	Question	Score	Justification
ADDRA	Q1	0	The approach offers no dependency management between decisions and other types of architectural entities, only focusing on traceability of changes in the decisions themselves.
	Q2	0.5	The traceability between different versions of the same decision(architectural deltas) helps deduce the change impact inflicted by those particular decisions.
	Q3	0	The approach only deals with recovering information regarding decision after the architecting phase.
	Q4	0	The approach does not propose any evaluation methods, it only allows the recoverability of knowledge.
	Q5	0.5	Maintenance support is provided by the traceability of decisions from early stages of the system life-span.
	Q6	0	The user needs to follow the iterations and the conceptual model base presented in [4].
	Q7	0	Integration with other tools and repositories is described as ”cumbersome” [4]. The approach is described to be difficult to configure for specific systems, existing tools are often language specific and current research does not seem to be able to capture dynamic or distributed behaviour.
	Q8	0.5	The approach offers a means of knowledge recoverability from a fully-developed system, therefore aiding in collaboration activities such as knowledge sharing.
	Q9	0.5	Although a detailed description of the decisions taken and traceability between different variants of the same decision is achievable, the approach does not offer solutions for mapping decision alternatives.
	Q10	0	As stated in [4], ”The recovery steps of ADDRA are far from trivial. Steps 13 (and partially 4) are not completely solved yet and remain under research by the design recovery community”. Further research is needed for a more user-friendly variant of this approach.

Table 3. Quality assessment scores - ADDRA

Approach	Question	Score	Justification
Architectural prototypes	Q1	0.5	Depending on the size and focus of the architectural prototype, a different number of dependencies can be captured by the model.
	Q2	0	This approach does not support change management or change impact analysis. Due to the nature of the approach, only a single variant of the system(an instance of a particular sub-part of the system) is modelled at a time. In order to represent changes in the design decisions, multiple prototypes need to be built.
	Q3	1	Using this approach, architects are able to model and represent decisions at a high level of detail by implementing a specific sub-part or sub-context of the whole system and test its performance.
	Q4	1	Architectural prototypes help test the system's response to different quality attributes given a specific context which allows for a thorough evaluation of the architecture and the decisions made during the architecting phase.
	Q5	0.5	Building an architectural prototype allows architects to observe the behaviour of their system given a certain set of circumstances, allowing them to test the capabilities of the system to be implemented. This may facilitate the maintenance process by giving detailed insight on what problems might surface in later stages and allow the testing of different solutions to those problems on a smaller scale.
	Q6	1	Depending on the specific context and part of the system which the user aims to model, the prototype allows for a relatively broad area of representation and a high degree of freedom towards implementation.
	Q7	0.5	Although the prototype itself is a separate entity and can be perceived as a self-suficient representation of the system, it can be used in combination with different theoretical techniques.
	Q8	0.5	The approach was tested with systems being built by different types of teams, including distributed teams. Although the aproach does not offer a specific way to deal with the team distribution, it is possible to adapt this approach to such a case.
	Q9	1	The approach allows for a thorough analysis of design decisions and their alternatives by providing a testing environment for sub-parts of the system with regard to different quality attributes. According to the results obtained by testing the prototypes, certain decisions will be more easily justified compared to their alternatives.
	Q10	1	Since this is an experimental approach, the way in which decisions are represented is suggestive even to parties without backgrounds in software architecture or specific technologies(this approach is mostly used for communication with the stakeholders).

Table 4. Quality assessment scores - Architectural prototypes

Approach	Question	Score	Justification
Graph oriented model	Q1	1	The graph model is particularly useful for describing relations and dependencies between different architectural entities. As described in [6], the model was tested with a representation of decisions, issues which led to them(requirements) and alternatives. The joint UML model also described in [6] allows the user to add more details regarding the specificities of the decision, implementation issues, etc.(tactics).
	Q2	1	With the help of the decision tree graph model, changes and dependencies are well represented, therefore it is easy to assert the impact of a specific change regarding a design decision.
	Q3	1	The graph and UML model presented in [6] help represent a detailed and well-described evolution of architectural decisions.
	Q4	0.5	The approach does not offer a specific tool or method for evaluating the architecture, however, due to the well-represented dependencies and good traceability it offers, it can be considered helpful for this process.
	Q5	0.5	The maintenance process is helped to a certain extent by the suggestively representative solution provided by the graph model.
	Q6	0	The user has to follow the model given.
	Q7	0.5	Paper [6] offers information on using this model with a specific type of knowledge management repository.
	Q8	0.5	Paper [6] suggests a combination of this approach and wiki repositories for helping support collaboration.
	Q9	1	The two combined models presented by this approach help the user in giving a thorough description and perform a detailed analysis in the decisions taken.
	Q10	0.5	The graphical representation using the decision tree is a suggestive approach and helps perform an easy communication of the architectural entities depicted, however, the user needs to have first studied and understood the meta-model used for the representation.

Table 5. Quality assessment scores - Graph oriented model

Languages for Software-Defined Networks

Aida Baxhaku Timo Smit

Abstract— Software-defined networking (SDN) is gaining a lot of attention over the past years as the more future-proof approach of organizing networks in terms of both software and hardware. The notion of SDN originates from two main concepts: a more extensive feature set, to allow for more dynamic and scalable solutions, as well as to decouple software and hardware, which allows software to adapt for changing requirements independently from used hardware.
In recent years many advances have emerged, particularly in the field of information technology, data science and networking. However, the traditional technologies behind networks are often low-level and lack features to support the nowadays common modular approach of networking. Where networks used to be declared statically and heavy in terms of maintenance, this approach involves high availability and scalability, by allowing extra instances to spawn and connect on-the-fly to relieve the workload on other nodes in the network. With this development, the overall complexity and problems that come with network management have grown, e.g. monitoring network traffic, specifying and composing packet forwarding policies, and updating these policies in a consistent way.
In this paper we will discuss the available languages for network programming to date, as well as find out what their limitations are. We do so by analysing their specifications, performing literature research on the selected set of languages and finally comparing the results of the analysis. We evaluate and compare the available languages based on their requirements, their feature set and their performance.

Index Terms—Network programming language, SDN (software-defined network), OpenFlow, Frenetic, FML, NOX, Flog

1 Introduction

Due to the changing environment in which IT services operate, the way they are organized is changing as well. Whereas networks were previously declared statically, they are nowadays expected to scale on demand and thus need to adapt to be flexible in their set-up. While networks were programmed deeply into their hardware interfaces, it is now more convenient to do this on a software level.

Internet structure and computer networks are composed of different components such as router, switch and middle-boxes that provide functionality such as topology discovery, routing, traffic monitoring, and access control. In order to manage and configure network devices, a set of specific and predefined line commands based on embedded operating system is used. These networks could be very large, highly complex and error prone. Managing these networks having scalability and security in mind is challenging, therefore different methods have been developed to cover for this need for scalable networks, which are called software-defined networks (SDNs).

The origins [12] of software-defined networking began shortly after Sun Microsystems released Java in 1995. One of the first SDN projects was AT&T's GeoPlex, which leveraged the network APIs and dynamic aspects of the Java language as a means to implement middleware networks. There have been many improvements of SDN ever since.

Several programming languages have been developed for this purpose, such as Frenetic [9], Nox and FML [6]. In this paper we analyse these languages and compare them in terms of performance and abilities. We do this by performing a literature study on the set of languages that is available to date. The most widely used languages are described and examples illustrate the specifics of the related language. The languages are assessed on their usability and feature set.

We first describe what software-defined networking entails and how OpenFlow abstracts the networking from hardware components on a software level. In the later sections we discuss the languages available to use for programming SDNs. In our paper we have chosen to analyze Frenetic, Flog, Nox and FML and compare them. Based on this comparison, we come up with a final conclusion regarding the best languages to use for specific purposes. We conclude our research with possible future research questions and fields which may be explored.

- *Aida Baxhaku, E-mail: a.baxhaku@student.rug.nl.*
- *Timo Smit, E-mail: contact@timosmit.com.*

2 Software-defined networking

SDN is defined [3] as a reactive system in which a logically centralized controller manages the packet processing functionality of a distributed collection of switches. It can both simplify existing applications and also serve as a platform for developing new ones.

Whereas traditional networking involves the use of statically defined configuration, software-defined networking (SDN) declares its nodes and their relations using a protocol defined on a software level. This abstraction has the benefit of reducing the dependence of the network logic on the hardware that is used, allowing for easier maintenance, but also increasing the scalability of the application.

SDNs are constructed in way that a logically centralized controller manages the packet processing functionality of a distributed collection of switches. SDNs make it possible for programmers to control the behaviour of the network directly, by configuring the packet forwarding rules installed on each switch. Despite the centralized functioning, the controller is replicated and distributed for scalability and fault tolerance.

Masoudi and Ghaffari [8] in their paper introduce three planes that divide the notion of SDNs, on the lowest level a data plane, the highest level an application plane and a control plane in between, as depicted in fig. 1. The data plane is made up of the core electrical integrated circuits (ICs) that distribute the data, whereas the control plane is partially hardware and software. The control plane of the network has on the one side the CPU and memory which is in charge of managing the aforementioned ICs, while the software is composed of an operating system that sends calls to the CPU. At the highest level of the chain is the application that implements the protocols on top of the available functions through the control plane interface.

One of the most important functions of the data plane is packet forwarding. Recent research is highly focused also on the programmability of data plane, which enables many different functions such as network appliances, cache and transcoding. The data plane's programmability is also able to satisfy networking tasks such as anomaly detection and traffic engineering.

The control plane in SDN is called the controller because it acts like a middle layer between the two other layers. This layer provides an abstraction of the overall network. The control plane layer is built from two basic components which are the applications and network operating system. The application component is primarily concerned with monitoring while the network operating system acts as a controller for SDN.

It is also the interface between the data plane and the control plane

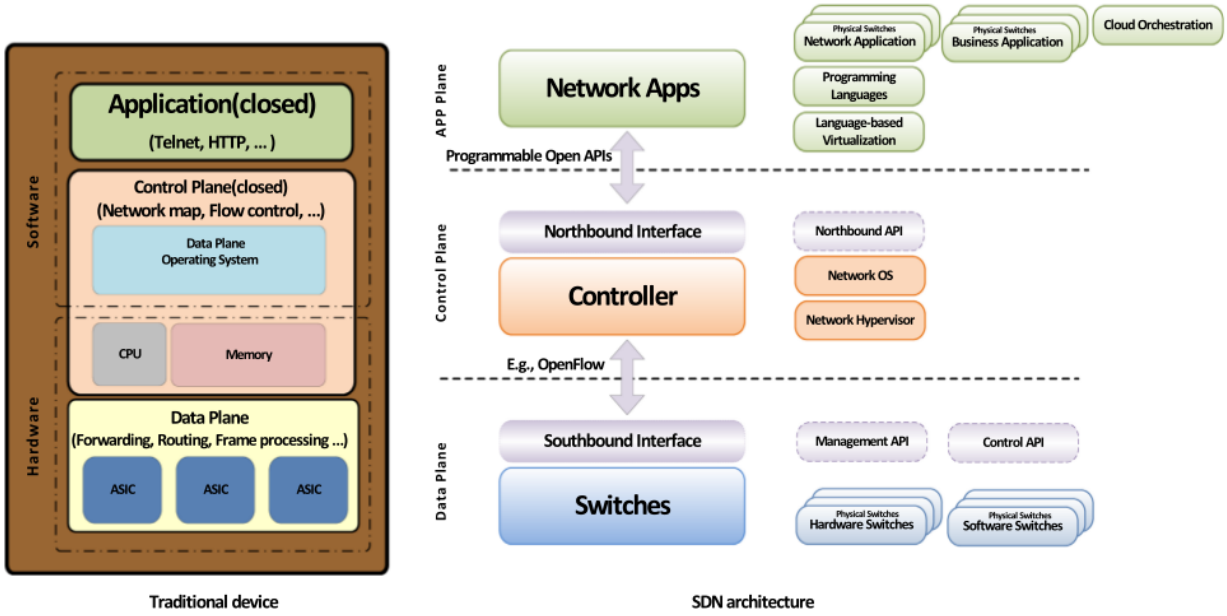


Fig. 1. Architecture [8]

that is interesting for software-defined networks. This allows the network programmers to control on a low level the way packets are distributed over the network. This for example allows for a shortest path routing to minimize latency, or minimize the load on certain hardware components available in the network to save energy. [3]

Another advantage of SDN is that the overall structure of the network is distributed among all the nodes of the network. This increases the reliability and the consistency of the structures, since these structures are now consistent and do not rely on the underlying architecture. It is for this reason that they can be reused inside the controller logic for many different applications, such as network leader election.

The application layer is the highest component of SDN. It includes all the applications that use services provided by the control layer in order to execute different network-related tasks.

3 OpenFlow

OpenFlow [10] [11] is a communication protocol that enables network controllers to determine the path of network packets across a network of switches. It is an open standard to deploy innovative protocols in production networks and its uses vary from applications such as virtual machine mobility to high-security networks and IP-based mobile networks.

In a classical router or switch, the fast packet forwarding (data path) and the high level routing decisions occur on the same device. An OpenFlow switch separates these two functions. The data path portion still resides on the switch, while high-level routing decisions are moved to a separate controller, typically a standard server. The switch will only contain a flow table, which determines how packets should be routed through the network. The switch and controller communicate via the OpenFlow protocol which defines messages such as packet-received, send-packet-out, modify-forwarding-table, and get-stats. The routing table on the switch, communication channel between the switch and the controller and the protocol that is used on this channel make up for the three components of an OpenFlow switch. [9]

The switches can be divided into two types: dedicated OpenFlow switches and OpenFlow-enabled switches. The first set of switches are simple switches that contain a routing table which determines how packets should be routed (e.g. matching a header, IP or MAC address, or a single TCP connection). The latter set of switches consists of third party switches made by manufacturers to support the OpenFlow protocol, but may also work independently. These switches implement the three components (the routing table, channel and protocol) on top of their own hard- and software.

The main purpose of the controller is to configure the OpenFlow switches and update their routing table. Such a flow table is composed of a set of rules consisting of multiple fields: a *pattern* that a packet has to match against, and a set of *instructions* which determine the route that a packet should be forwarded to. The routing table could consist of simple, static flows, but may also change during runtime as nodes are added to or removed from the network.

Regardless of the nature, the goal is to match as many packets as possible against one of the flows. Packets are typically routed through the network by the logic on the available switches. However, the main controller is used as fallback for flows that are not recognized. Hence performance is significantly better when the logic can be distributed among the switches rather than being handled by a single controller, which is less scalable.

4 FML

Hinrichs et al. [6] presents a flow-based management language (FML), which is a declarative policy language. It is developed to replace the traditional methods for declaring routing policies. FML is designed to be a basis for NOX, which is a network-wide controller and will be discussed hereafter.

4.1 Syntax and semantics

FML is based on Datalog, a declarative logic programming language. Rules are of the form *if-then*, since packets have to match some rule and a corresponding action should be taken to route the packet to its destination. One of the major differences between Datalog and FML is that variables in the body (consequent) of the rule need not occur as a positive literal in the body once, but rather occur once in the head (antecedent or constraint) somewhere in a rule. This simplifies the creation of rules and also improves performance, as the variables do not have to be cross-checked.

A set of rules define a policy, in which the ordering of those rules does not alter the behaviour of the policy. The benefit of this approach is that combining policies to set up one firewall will not have unexpected outcome. On the other hand, FML does allow a set of rules to be conflicting; that is, one rule may allow a certain packet, while another one denies it.

4.2 Conflict resolution

For this reason, conflict resolution has to take place. To implement this, keywords will have priority over each other; from highest priority to lowest: *deny*, *waypoint* and *avoid*, *allow*. Hence, a policy that would both allow and deny a packet, would simply deny it.

Another mechanism that has been built for resolving conflicts is a *cascade*, which is a set of policies in a specific order. Suppose $P_1 < P_2 < \ldots < P_n$ is a cascade, then P_2 would have priority over P_1, and P_n would have priority over all policies.

4.3 Example

The described semantics can be observed in the following example:

$$deny(U_s, H_s, A_s, U_t, H_t, A_t, Prot, Req) \Leftarrow blacklist(U_s)$$
$$allow(U_s, H_s, A_s, U_t, H_t, A_t, Prot, Req) \Leftarrow superuser(U_s)$$
$$superuser(alice)$$
$$superuser(bob)$$
$$blacklist(bob)$$

In this example, *superuser* will have no restrictions on the channels it wants to address. The arguments that are passed to *allow* are variables and correspond to the eight fields of a flow. *superuser* takes as a parameter a user, and the lines of the form $superuser(x)$ denote that any x is also a *superuser*, hence *alice* is a superuser. Additionally, *blacklist* denies access to the flows. Since *bob* is both a superuser and is blacklisted, he will not have access, as *deny* has precedence over *allow*.

5 NOX

Built upon OpenFlow, NOX is a network-wide controller [5] that involves the use of several switches and servers. Each of the NOX servers will execute a controller service that OpenFlow introduced. In addition to this, NOX adds a network view that contains the results of the executed logic and network observations as performed by the controller services. This network view is updated during runtime as the switches make decisions about packet routing.

Like FML, NOX uses flows to control the packet routing. This is done to maintain scalability, since routing each packet using a routing table will quickly pose performance issues for large applications. The scalability of NOX is further increased because the main operations can all be run in parallel.

Only the network view is required globally across the network, since all of the components need this to determine the packet routing logic. However, this view is only updated on the order of tens per second, whereas packets arrive on a busy link (10Gbps or more) millions of times per second. [5] The latter can be run in parallel because this logic is executed on different switches.

5.1 API

NOX implements an event-based API that can be used to write controller logic. Events such as *switch (dis)connected* and *packet received* can be hooked up to by event handlers, which are then able to kill the event or let it continue and subsequently, let the next handler do its logic. The event interface provides both default OpenFlow events as well as NOX-specific events.

The API however does not provide any security to malicious controller logic. It is thus possible to drop packets or connections randomly, overwrite memory or let the system run into a deadlock state. [5]

6 FRENETIC

Frenetic is a type of SDN controller, a language for programming SDNs. The initial aim of Frenetic was to increase the level of abstraction of the SDNs. Nowadays the key principles in which Frenetic is based [9] are **modularity** and **composition**. Frenetic [3] is based on querying network state, expressing policies, reconfiguring the network. In the subsections below the basic principles of Frenetic are explained in detail and examples are given in order to show how they actually improve the overall performance of SDNs.

6.1 Querying network state

Frenetic offers a high-level query language which controls or monitors the received information through different operators. Traffic monitoring is an important tasks of SDN. In Frenetic this is done through the counters that are associated with switches. These switches have rules, which maintain a counter per each rule, in order to keep track of the number of packets and bytes processed by that rule. It is difficult to install the rules in advance since the switches do not provide enough space for them. The query language offers the programmers the facility to decide which ports or packet forwarding they want to monitor. The way in which this data is acquired is [3] the responsibility of the runtime system. This approach increases the modularity of the system, which would otherwise write the rules individually for each module, hence increasing complexity and the time of execution. Frenetic simplifies not only the specification of a single query, but also of multiple queries without worrying about their interactions the runtime system selects rules in order to satisfy all of the queries. It also allows programmers to specify predicates by leaving out the details of how to construct and optimize switch-level rules to the runtime system.

The query can be registered by using operators such as `Select(bytes)` and `GroupBy([srcip])` which allow the runtime system to dynamically generate appropriate rules. Therefore all traffic is sent to the controller and upon receiving the first packet from a specific source, the runtime system installs a rule that matches future traffic from that host. For the second packet it generates another rule that matches that specific source and it follows the same procedure for every packet. In this way, the counter maintains the necessary information in order to implement the query which is used for processing future traffic monitoring. Another helpful command that Frenetic uses is `Limit(1)`. The controller and switch do not communicate instantaneously therefore multiple packets arrive at the controller before the rules are installed. In order to automatically handle these unexpected packets, Frenetic allows a query to specify the number of packets it wants to receive to one, by using the aforementioned operator `Limit(1)`.

6.2 Example

Through this simple example we would like to illustrate the power of Frenetic in practice and how it supports the querying of networks. The following example is written in query language which highly resembles SQL. We will briefly describe how this short script works.

MAC Learning An ethernet [3] switch performs medium access control (MAC) learning to identify what interface to use to reach a host. MAC learning can be expressed in Frenetic as follows:

```
Select(packets) *
GroupBy([srcmac]) *
SplitWhen([inport]) *
Limit(1)
```

The `Select(packets)` clause implies that there are packets ready for the program to receive them. The `GroupBy([srcmac])` groups the set of queried packets into subsets based on the src- mac header field, resulting in one subset for all packets with the same source MAC address. The `SplitWhen([inport])` clause has a similar function as `GroupBy`, hence it subdivides the set of selected packets into subsets; however, whereas `GroupBy` produces one subset of all packets grouping them according to their given header fields, `SplitWhen([inport])` generates a new subset each time the header values change (e.g. when inport changes). Together, the `GroupBy([srcmac])` and `SplitWhen([inport])` clauses state that the program wants to receive a packet only when a source MAC address appears at a new ingress port on the switch. The `Limit(1)` clause limits the intake of the packets with only one, so it does not receive all packets from that source MAC address at the new input port. The result is a stream of packets that the program can use to

update a table mapping each MAC address to the appropriate ingress port.

6.3 Expressing policies

Another advantage of Frenetic is the high level policy language that it offers which makes it easy to specify the packet forwarding behaviour of the network. The [3] older versions of interfaces make it difficult to perform network multitasking by using different modules, since packet handling rules installed by one module often interfere with overlapping rules installed by another module. Frenetic's policy language has a number of features that are designed to make it easy to construct and combine policies in a modular way. Frenetic applies parallel composition which could be illustrated with the example of the combination of the repeater with the monitor where the modules act on the same stream of packets. The repeater module applies a forwarding policy and the monitoring module queries the traffic.

6.4 Reconfiguring the network

One of the biggest struggles that programmers have faced while working with SDN is the need to install and uninstall each packet forwarding rule manually on individual switches. Frenetic automates the process by adding the abstractions for updating the global configuration of the network. The runtime ensures that there is no mix of policies, therefore guaranteeing an error-free and less complex system where loop freedom, connectivity, and access control are not violated during transition periods between policies.

7 Flog

Flog is an event-driven logic programming language [7] that is based on FML. The logic programming aspect has been adopted from FML because of the nature of the problem, which is processing logic based on table collections with packet forwarding rules. Besides that, Flog is inspired by Frenetic, which states that the controller is divided into three components, a mechanism for querying network state, one for processing data collected from queries and one for generating the rules that determine the packet logic. [7]

7.1 Events

A typical live network undergoes several events, e.g. packets that arrive in the network, switches that are enabled, disabled, possibly crash or switches that connect to a certain application. As for the packets that arrive, Flog will route these to the controller by default. This controller will then generate a policy that will serve as an example for future arrival of the same kind of packets. This policy in its turn is distributed among the switches in the network. The benefit of this approach is that the logic is now shared and workload can thus also be distributed over different nodes of the network.

7.2 Policies

Policies are generated by the controller and are typically of the form `h1(F1), h2(F2), ... |> action, level(i)` where `h1(x)` is a constraint on field `F1` in the packet, such as a target address or port, that should be matched. When a match occurs, the rule with the highest priority determined by `level` will execute the network logic specified by `action`.

7.3 Uses

Flog's capability of generating policies based on historical information, can be used in applications such as firewalls. For example, incoming packets could be only allowed when a certain IP has been already seen in an outcoming packet to prevent unknown outsiders from connecting to the network. This can be described in Flog as follows:

```
# Network Events
flow(dstip=IP), inport=2 --> seen(IP).

# Information Processing
seen(IP) +-> allow(IP).
allow(IP) +-> allow(IP).

# Policy Generation
inport(2) |> fwd(1), level(0).

allow(IP) -->
  srcip(IP), inport(1) |> fwd(2), level(0).
```

8 Conclusion

Following the innovation of programming the networks in a software level rather than a deep hardware level, many different programming languages have been developed for programming software-defined networks. These new languages tend to improve the overall performance of SDN compared to the older versions that were previously used.

OpenFlow's interface is quite similar to the features of the underlying switch hardware, therefore it is a bit low level compared to the newer versions of the languages.

Based on the latest research, Frenetic is considered as one of the best, compared to the other programming languages for SDNs. Therefore, interest in it is rapidly growing. The Frenetic language is an innovation that offers a new approach which attempts to reduce the difficulties that network programmers have previously faced while using low level languages such as OpenFlow. The main idea of Frenetic is to reduce the manual work and to delegate the routine tasks to the runtime system while increasing modularity and composition. The language offers a set of abstractions for writing controller programs for SDNs. It comes with a compiler and a well written documentation.

Built upon OpenFlow is FML, a declarative policy language. This language introduces rules of the form *if-then* and resolves certain issues with regards to semantics of the language. It is easier to maintain policies and combine them, as the underlying rules do not have to occur in the same order. Instead, conflicts in the rules are resolved by priority in the operations, which is more intuitive from a programmer's perspective.

NOX improves on this by parallelizing more components of the network and combining Flog and Frenetic features. It has the same programming logic as FML and it uses the component division such as Frenetic. NOX supports the same low-level interface, which forces applications to be implemented using programs that manipulate the state of individual devices. It also makes use of an event-based API that allows the programmer to hook into any OpenFlow event, as well as additional NOX events.

Flog is network event driven and it focuses on packet flow. It is a simple language which combines characteristics from FML and Frenetic. However it is limitied to flow control. This paper addresses each language in terms of usability, features and their possible use cases. In the future, more research could be done on the runtime performance of each of the languages. One more point we would like to add to our future studies is the research of another language that has attracted a lot of interest lately: Pyretic. In further studies we will depeen our research in Pyretic.

References

[1] M. Casado, N. Foster, and A. Guha. Abstractions for software-defined networks. *Communications of the ACM*, 57(10):86–95, Oct. 2014.

[2] A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi. Hierarchical policies for software defined networks. In *Proceedings of HotSDN'12*, pages 37–42, Aug. 2012.

[3] N. Foster, A. Guha, M. Reitblatt, A. Story, M. J. Freedman, N. P. Katta, C. Monsanto, J. Reich, J. Rexford, C. Schlesinger, D. Walker, and R. Harrison. Languages for software-defined networks. *IEEE Communications Magazine*, 51(2):128–134, Feb. 2013.

[4] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker. Frenetic: a network programming language. In *Proceedings of the 16th ACM SIGPLAN international conference on Functional programming*, pages 279–291, Sept. 2011.

[5] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. Nox: Towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105–110, July 2008.

[6] T. L. Hinrichs, N. S. Gude, M. Casado, J. C. Mitchell, and S. Shenker. Practical declarative network management. In *WREN '09: Proceedings of the 1st ACM Workshop on Research on Enterprise Networking*, Aug. 2009.

[7] N. P. Katta, J. Rexford, and D. Walker. Logic programming for software defined networks. http://frenetic-lang.org/publications/logic-programming-xldi12.pdf. [Online; accessed 5 March 2017].

[8] R. Masoudi and A. Ghaffari. Software defined networks: A survey. *Journal of Network and Computer Applications*, 67:1–25, Mar. 2016.

[9] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, Apr. 2008.

[10] OpenFlow. Official website. http://archive.openflow.org/. [Online; accessed 5 March 2017].

[11] OpenFlow. Specification. http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf. [Online; accessed 5 March 2017].

[12] Wikipedia. Software-defined networking. https://en.wikipedia.org/w/index.php?title=Software-defined_networking&oldid=765017066. [Online; accessed 5 March 2017].

Multidimensional projections: Scalability, Usability and Quality

Fank N. Mol

Abstract—One of the challenges in modern science is to comprehend multidimensional data. Whenever the number of dimensions are high, i.e. larger than 5, we can speak of multidimensional data, where we are not able to present all the given data in to a plot, since we are limited to 3 dimensions. Multidimensional Projections (MP) are techniques which focus on mapping the m dimensional data to 2 or 3 dimensions, where we will focus on techniques which map to 2D in this paper. All the recent techniques are based on the Multidimensional Scaling (MDS) method and the Principal Component Analysis (PCA), where these two techniques lack the ability for most of the data sets to preserve the underlying non-linear data structure. We consider four state-of-the-art techniques which have a different approach to map multidimensional data to two dimensions. These techniques are the t-SNE, LAMP, Isomap and LLE. Comparing these techniques gives the pros and cons for each technique in terms of scalability, usability and quality.

Index Terms—Multidimensional projections, t-SNE, LAMP, Isomap, LLE.

1 Introduction

In many application fields large amounts of multidimensional data are produced. In fields such as engineering, medical sciences and business intelligence, data is produced where each data point has a large amount of attributes. Because of the large amount of variables (larger than 5) per data point, this data is considered high dimensional [4]. Whenever we use data sets which are high dimensional, we face the problem of deducting information from this data set, such as which data points form a group (a cluster), and which data points are outliers. Using tables, we are not able to deduct such information from the data, and our visual methods, such as scatter plots, can only present the data with a limited number of dimensions. One solution to this problem is to map the data to lower dimensional data, which preserves the underlying information. Multidimensional Projections (MP) makes use of techniques that can visualize data with a high number of dimensions, in either 2D or 3D.

In this paper, we restrict ourselves to MP techniques which map from m dimensions to 2 dimensions. LAMP [2], LLE [3], Isomap [5] and t-SNE [7] are state-of-the art techniques for the mapping. All face a number of challenges. One of these challenges is the comprehensiveness of the projection. Another challenge of the representation is that it needs to be able to visualize certain information on the given data, i.e. data patterns. Also, another challenge the technique faces is computational complexity.

Since the techniques differ in the above mentioned challenges, a user needs to select a technique that suits their goal. Hence, we give an overview of state-of-the-art MP techniques, which we will compare using the following three characteristics: scalability, usability and quality. We measure these characteristics as following:

- Scalability will be measured in terms of it's computational complexity. The interesting part of the complexity is effect which the number of observations and the number of dimensions has, i.e. a method will be better compared to another if the computational complexity is less increasing in n and m, the number of observations and the number of dimensions. Note that it is possible for two different methods that it is, in terms of scalability, favorable to choose one above the other for some high number of observations and a low number of dimensions, while the other is favorable if the observations are lower and the dimensions are higher.
- Usability will be measured in how well the projection can be explained to the user. Hence, this is a metric which aims at ranking the complexity of what the method does, and the complexity of the outcome.
- Quality will be measured by how well the projections maintain the data patterns. We will look in how well the projections are able to present outliers, clusters and distances.

One of the papers suggest that many different techniques exist that create a 2D projection [4]. These can be classified as follows: 'Distance versus neighborhood preserving', 'Global versus local' and 'Dimension versus distance'. We will use at least one technique of each classification in our analysis.

In Section 2 we will give an overview of the Multidimensional Projections, in Section 3, we will outline the four proposed techniques which we will discuss, where these techniques will be compared on their scalability, usability and their quality in Section 4. We will end by giving a conclusion and the opportunities for further research in Section 5.

2 Multidimensional Projections

One of the problems which certain areas of research have, is that they want to analyze multiple features for each observation. Storing these features can easily be done, for example in a table. However, when the number of observations are high and when we store a lot of features, for example 1,000 observations of 5 features each, we can not subtract the underlying data structure from the table. Hence, a multidimensional projections (MP) can be used as a solution to the above stated problem. MPs attempt to convert the multidimensional data in to a data set containing only 2 or 3 dimensions. In other words, MPs are attempting to make a mapping $\mathbb{R}^m \rightarrow \mathbb{R}^3$ or a mapping $\mathbb{R}^m \rightarrow \mathbb{R}^2$. For this paper, as was mentioned in the introduction, we are restricting ourselves to techniques which map to 2 dimensions. The remaining data can usually be depicted in a scatter plot. Most of the MP techniques are based on the principles of Multidimensional Scaling (MDS) and Principal Component Analysis(PCA).

The methods based on MDS [6] all store a n by n matrix named D, where n represents the number of observations, where distances between every observation in the m-dimensional space are stored, i.e. $D_{i,j}$ is the distance from observation i to observation j.

Also, one of the building blocks for MP is PCA [1]. PCA produces an orthogonal data set, where the remaining features are uncorrelated. Hence, this results in a reduction in the number of

• *Frank N. Mol is with Rijksuniversiteit Groningen, E-mail: frank.n.mol@gmail.com.*

Manuscript received 7 March 2017.
For information on obtaining reprints of this article, please send e-mailto:frank.n.mol@gmail.com.

features the data set has. Note, that if the initial data set consists of exclusively uncorrelated features, than the PCA method will have the initial data set as output and this implies that there is no reduction in the number of features.

Both the above mentioned techniques, MDS and PCA, are linear techniques, which have the advantages that they are both easily implemented and have a low computational complexity. However, as Tenenbaum et al. states, both techniques fail, in many cases, to preserve the data set it's underlying non-linear data structure [5].

3 Non-linear MP Techniques

In this paper, we restrict ourselves to MP techniques which map from m dimensions to 2 dimensions. We consider four different techniques, LAMP [2], LLE [3], Isomap [5] and t-SNE [7]. The reason for choosing these four techniques is since these techniques are state-of-the-art in multidimensional projections to 2 dimensions, and due to the diversity of the techniques. The latter is helpful since we aim to give the pros and cons of each technique compared to each other such that the reader can choose which of these techniques is most suitable for the context one is facing. In the following subsections, we will explain the above mentioned techniques.

3.1 t-SNE

The technique t-SNE (t-distributed Stochastic Neighbour Embedding) is a variation of the Stochastic Neighbour Embedding which considers a t-distribution on the underlying data density instead of a Gaussian distribution.
Conditional probabilities which represent the probability of data point i to choose data point j as its neighbor, if neighbors were picked on their underlying density based on the t-distribution.

A distinctive characteristic of the t-SNE method is that this method aims to preserve the neighborhoods, while the common approach, as we will see in the techniques presented in the next subsections, use the distances.

The computational complexity of the t-SNE method is $\mathcal{O}(n^2)$, which means that the complexity is quadratic in the number of observations. Hence, as the paper which proposes t-SNE states, this limits the scalability to around 10,000 data points [7].

To conclude for this technique, t-SNE is a technique which retains a good local structure without losing important the global information.

3.2 Local Affine Multidimensional Projection

Based on mapping theory, the Local Affine Multidimensional Projection (LAMP) technique has the aim to increase the flexibility for than other techniques such that is has better groundwork for visual-oriented goals, instead of the mainstream techniques which aim for either shorter computation times and higher accuracy.

The technique is based on orthogonal mapping theory. The steps which are taken by this procedure are as following

- At first, a subset of data points of the complete data set are chosen to be the control points, which are used to map to visual dimensions (2D).
- Then, the algorithm works as following, for each data point:
    1. The weights α_i are are calculated for an observation x, using the distance from x and another observation x_i, for $i \in [1, ..., n]$. Then the weights are obtained as following

$$\alpha_i = \frac{1}{||x_i - x||^2} \tag{1}$$

    2. compute the best affine transformation, which minimizes the the weighted and squared distances between the affine transformation of a data point subtracted by its counterpart.

The aim of the LAMP method is visualization based, which results in slower computational times but a big improvement in terms of usability. The latter holds since the method is flexible, effective and is not hard to implement [2].

The computational complexity of the LAMP consists of the computational complexity of doing a Singular Value Decomposition (SVD) for each data point. Since the SVD consists of a $m \times 2$-matrix only, we have that it can be decomposed in $\mathcal{O}c$, where c represents the number of control points. Hence, we have a computational complexity of $\mathcal{O}(cn)$ for this technique.

3.3 Isomap

Isometric feature mapping, or in short Isomap, is a multidimensional projections technique which uses local metric information, which is able to discover the nonlinear degrees of freedom of the underlying observations [5]. The technique combines the major algorithmic features of principal component analysis (PCA) and multidimensional scaling (MDS). Where the former and the latter technique find an embedding of inter point changes, where these distances are Euclidean. Hence, both PCA and MDS focus on linear structures. The advantage of Isomap is that it focusses on data sets that contain non-linear structures.

Isomap uses geodesic distances between the observations

1. Determining neighbors based on distances $d_X(i, j)$ between pairs of points i and j. neighborhood relations will be stored in graph G.
2. Calculating the geodesic distances $d_M(i, j)$. Which are calculated by the shortest path distances in graph G between point i and point j.
3. Apply multidimensional scaling to the distance matrix where these distances are the geodesic distances calculated in in step 2. This step is calculated by using the Partial Eigenvalue Decomposition.

Then, using a cost function which extracts the euclidean distance form the geodesic distance, will result in a two dimensional (euclidean) representation of the multidimensional input. [5]

To obtain the computational complexity, we add up the complexities of all the three steps mentioned above. For the first step, determining the neighbors, we have a computational complexity of $\mathcal{O}(m\log(k)n\log(n))$. For the second step, the shortest path in a graph we use the computational complexity of the Floyd-Warshall algorithm which is $\mathcal{O}(n^3)$. And the last step, the partial eigenvalue decomposition has a computational complexity of $\mathcal{O}(n^2)$.
In the general cases, we can easily assume that $m < n$, which then results in a computational complexity of the order of the shortest path algorithm, which is $\mathcal{O}n^3$. Note however, that this is not the case when the number of observations are lower than the number of dimensions, but for this paper we assume that in general $m > n$ is considered.

3.4 Locally Linear Embedding

Locally Linear Embedding (LLE) is a MP technique which uses local symmetries of linear reconstructions. Also, LLE obtains reconstruction errors. Using this local linear information, LLE constructs a lower dimensional data set which obtains the global geometry.

The steps of the LLE algorithm are as following

1. Determining neighbors based on distances $d_X(i, j)$ between pairs of points i and j.

2. Construct a weight matrix using the neighbors of each data point.

3. Apply multidimensional scaling to the distance matrix where these distances are the geodesic distances calculated in in step 2. This step is calculated by using the Partial Eigenvalue Decomposition.

To obtain the computational complexity, we add up the complexities of all the three steps mentioned above. For the first step, determining the neighbors, we have a computational complexity of $\mathcal{O}(m\log(k)n\log(n))$. The second step constructs a weight matrix by a $k \times k$ linear equation, where k represents the number of neighbors used. This results in a computational complexity of $\mathcal{O}(m \times n \times k^3)$. And the last step, the partial eigenvalue decomposition has a computational complexity of $\mathcal{O}(n^2)$.

4 Comparison

We will compare all the techniques described in Section 3 in three different characteristics: scalability, usability and quality.

4.1 Scalability

To compare the Scalability, we use the computational complexity for each of the techniques. An overview of the computational complexities is presented in Table 1. Note, that in this table we have that n represents the number of data points, m represents the initial number of dimensions, c represents the number of control points used by the Isomap technique and k is the number of nearest neighbors used by the nearest neighborhood search. Note that if you have a table, that n represents the number of rows, and m represents the number of columns. It is easy to see from the computational complexities presented in Table 1 that LAMP has the lowest computational complexity, since the number of control points is always lower than the number of initial data points, i.e. we have that $c < n$. At second best,w e have the t-SNE method, and depending on the size of m w.r.t. n and the number of neighbors k used, we have either that Isomap or LLE has the highest computational complexity. In the general case, we can assume that Isomap has the higher complexity, due to the high computational complexity of the shortest path algorithm.

Method	Computational Complexity
t-SNE	$\mathcal{O}(n^2)$
Isomap	$\mathcal{O}(n^3)$
LAMP	$\mathcal{O}(cn)$
LLE	$\mathcal{O}(m\log(k)n\log(n))$ $+\mathcal{O}(m \times n^2)$

Table 1. Computational complexities of the different MP techniques.

4.2 Usability

The usability metric is about how well the data is explained to the user. This metric is somewhat influenced by the quality of the techniques, which will be discussed in the next section. Below, we will discuss the usability for each technique

- When using t-SNE, the aim is to find neighborhoods. When plotting t-SNE in a scatterplot, it is easy for the user to read the clusters, and hence t-SNE performs well on the usability.
- LAMP is a method which aims to be more visual orientated. Hence, the scatterplot resulting from this technique is easy to read for the user, and also this technique performs well on usability.
- Isomap and LLE have somewhat the same type of scatterplot, where a good understanding of the technique is needed by the user to obtain the information from the plot. Hence, these two techniques do not score well on usability.

4.3 Quality

The quality metric is about how well the underlying data structures are preserved by the Multidimensional Projection technique. One interesting quality characteristic we find is that the t-SNE aims at preserving the neighborhoods, while the LAMP, Isomap and the LLE techniques aim at preserving the inter point distances. An advantage of the approach t-SNE uses, is that the local distances are ignored and hence we have a better distinction in clusters.

On the other hand, since t-SNE preserves the neighborhoods instead of distance, it does show the outliers less well as the techniques which aim to preserve the inter point distances. Hence, it depends on the context whether the user wants to use t-SNE or one of the other 3 techniques, i.e. use t-SNE for finding clusters and use LAMP, Isomap or LLE if you are interested in outliers.

To compare the LAMP, Isomap and LLE technique in terms of quality, we distinguish the LAMP technique with the Isomap and the LLE by stating that, as Joia et al. [2] suggests, that LAMP is more visual orientated. This was an advantage for this technique in terms of usability, but the accuracy is decreased by this. Hence, in terms of quality, Isomap and LLE score better.

Finally, to compare Isomap and LLE, Roweis and Lawrence [5] suggest that Isomap is an improvement to the LLE method, where this improvement is aiming at the global structure of nonlinear manifolds. Hence, a choice in terms of quality between these two methods depends on the input data.

5 Conclusion

Multidimensional Projections can be done in several ways. We did look at techniques which map from $\mathbb{R}^n \to \mathbb{R}^2$. Four state-of-the-art techniques are t-SNE, Isomap, LAMP and LLE. These techniques differ in the way they are constructed. Even while Isomap and LLE do have same elements, their outcomes and computation time are not similar.
We have found that the LAMP method has the lowest computational complexity, which makes us conclude that this method is the best scalable method of the above mentioned four methods. The LAMP and t-SNE techniques are performing better in terms of usability, where the Isomap and the LLE technique need a good understanding of the technique for the user to be able to read the plots. The quality is depending on what the user aims to obtain in the resulting plot using a MP technique. For obtaining cluster information, t-SNE has the better quality, compared to the other three. If the inter point distances are more of use for the user, Isomap or LLE should be used, depending on the input data.

Using the above results, we can conclude that the 'best' technique to use for the reader depends on the context. However, we would suggest in general cases to use the LAMP technique, due to the decent visual characteristics and the scalability when we are more interested in inter point distances and outliers, while we would suggest to use the t-SNE technique to gather more information about clusters.

For further research, we suggest to construct, or obtain, data sets to run these four techniques on, and compare the outcomes to see if the above stated results based on the posed theory are being confirmed.

References

[1] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.
[2] P. Joia, F. V. Paulovich, D. Coimbra, J. A. Cuminato, and L. G. Nonato. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics 17.12*, pages 2563–2571, 2011.
[3] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[4] R. R. O. D. Silva, P. E. Rauber, and A. C. Telea. Beyond the third dimension: Visualizing high-dimensional data with projections. *Computing in Science & Engineering*, 18:98–107, 2016.
[5] J. B. Tenenbaum, V. D. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290.5500:2319–2323, 2000.
[6] W. S. Torgerson. Multidimensional scaling i: Theory and method. *Psychometrika*, 17:401–409, 1952.
[7] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Software Support for Cloud Migration

Timon Back and Peter Ullrich

Abstract— Migrating an enterprise system toward a cloud computing environment requires ample preparation and poses certain risks and difficulties, which are best evaluated before starting the process. Moreover, the migration process is iteratively performed in multiple steps, for each of which scientific tools offer automatizing and support.
In this paper, we describe the steps necessary for migrating an enterprise system to a cloud based infrastructure. We give an overview of a chosen collection of scientific tools, which help with each of these steps. The migration process can be divided into a preparation and execution part, each of which use a different set of tools. Tools used in the preparation phase offer decision support for finding the best suited cloud service provider and cloud machine instance type to avoid under- and over-provisioning. Some of the considered tools for the execution phase contain frameworks for semi-automatic migration, which also take a look at dependencies in the software or processes.
We present the approaches of these tools and give an analysis of their usefulness depending on the situation in which they are used. Eventually, we identify areas of the migration steps, which lack coverage of scientific tool support and propose functionality for such tools.

Index Terms—Cloud, Migration, Framework, Software Support, Decision Support, Enterprise.

1 Introduction

In recent times, hosting an enterprise system in a private infrastructure has become less appealing since cloud computing providers offer very appealing solutions for the disadvantages of having an own infrastructure.

Managing an own server infrastructure to host an enterprise system entails having an abundance of computing and storage power in order to be prepare for future scaling of the system. However, such unused resources cost in their acquisition and add to the overall maintenance bill since they have to be in working conditions and updated regularly as well. Physical restrictions like spacial capacities and connection throughput limits complicate scaling the system quickly and without much overhead. However, having full control over ones infrastructure helps in ensuring data privacy and security since a dependency on a third party is absent.

Nevertheless, cloud computing services offer an appealing product to companies, which can hand over the responsibility of providing and maintaining their own infrastructure to the cloud provider and save costs by only paying for the resources they actually use. Continuously falling prices for cloud infrastructure paired with an increasing number of supported technologies as well as improved usability has let many enterprises to migrate their existing systems into the cloud. However, unlike enterprise systems which are still in planning and whose system architecture can be aligned with a cloud based infrastructure from the beginning on, already existing enterprise systems tend to be developed along the constraints of a specific infrastructure and might not be easily migrated to the cloud. Additionally, since cloud computing services are a relatively new and disruptive technology, migrating an enterprise system entails certain risks and probably unforeseeable costs and poses new security issues since the own system is not under the full control of oneself anymore.

A series of tools have been developed in recent times, which aim to help enterprises in their decision making on whether to perform a migration of their system to the cloud (from now on called "Cloud Migration") or not, and help with the actual migration. This article is aimed to offer an overview of a chosen set of such tools and to explain their usefulness and potential overlap.

We chose to describe the CloudGenius tool[10], the CloudMIG Model[3], the ACL Model[7], and two tools developed taken from [9]. In comparison with other papers about cloud migration, the chosen papers provide actual tools and decision support in opposite to lists of questions (e.g. [2]) and non-concrete requirements engineering (e.g. [1]). In [12] the migration of an existing application is documented, however it lacks a tool or standard approach, which can also be applied in other migration processes.

We extracted the mentioned tools and ordered them based on for which step of the migration process each tool is helpful. By doing this we created a roadmap of the migration process and specified which steps of it are covered by the mentioned tools. Steps that are not covered at all or only partially will be discussed in the Conclusion section.

- *Timon Back, E-mail: T.Back@student.rug.nl.*
- *Peter Ullrich, E-mail: P.J.Ullrich@student.rug.nl.*

2 Background

A general manual about the steps toward a cloud based infrastructure is presented by [10]. The manual consists out of five steps (see Figure 1), which will be explained shortly. Each of them lower the flexibility in the next step. Additional to the five steps presented by [10], we extended the manual by the Cloud Compatibility Evaluation step proposed by [3].

This added, initial step we find important to add in order to evaluate whether a migration to the cloud is feasible at all or whether such a migration entails an impractical amount of work, creates significant security and/or privacy issues, or constitutes other important reasons contra to a migration. This fundamental feasibility analysis was in our opinion only insufficiently covered by the original five step model by [10], which is why we added it. All these steps are the structure of this paper and will be explained in more detail in their corresponding sections, but we will give a brief abstract of their proceedings here.

First, the *Cloud compatibility evaluation* based on [3] is conducted. The architecture of the system is analysed and its compatibility for a migration is categorized based on the five level Cloud Suitability and Alignment (CSA) scale proposed by [3]. Eventually, based on the score of the system on the CSA scale, a go/no-go decision is made on whether to migrate the system or not.

Second, the *Cloud infrastructure service selection* is done and the service provider of the Infrastructure-as-a-Service (IaaS) is selected. This requires already a "well-thought decision", since multiple factors like price and the Service Level Agreement (SLA) is involved.

Third, the *Cloud VM image selection* is done. Since the infrastructure in the cloud is different and a VM image compatible with the cloud service provider is required, a selection for the appropriate VM image has to be made.

Following is the *Cloud VM image customization*. Within this step, the VM image is adapted to reflect the local infrastructure. In the end, the functionality of the VM and the local system should be the same.

In step 5, the *Migration strategy definition* is created. This involves a plan on how to move from the local infrastructure to the cloud in-

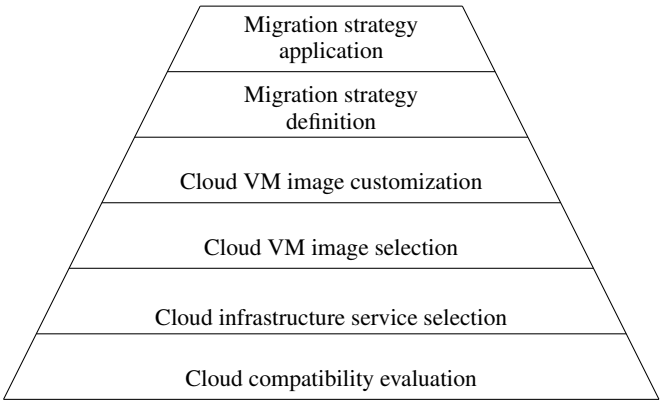


Fig. 1. Six steps towards a cloud infrastructure (bottom to top) based on the five step steps model of [10] as described in the background section

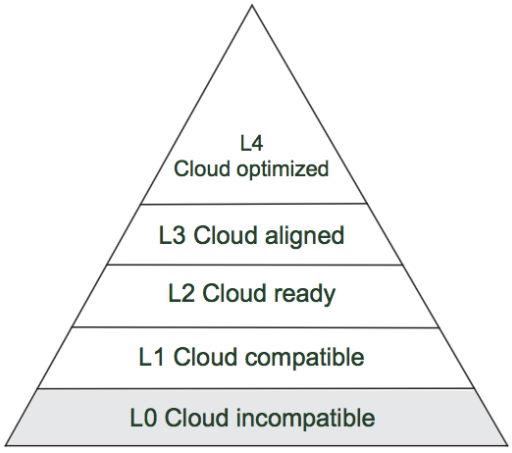


Fig. 2. The CSA Hierarchy according to [3]

frastructure. This also includes a plan for transferring the data into the cloud.

Finally, the *Migration strategy application* finishes the migration process. All configurations and settings are applied and/or updated. In the end, a complete, production-ready system is available in the cloud.

In the following, we separate the migration process into the preparation phase and execution phase. The preparation phase includes considerations about the feasibility and implied tasks needed to be fulfilled for the migration. The goal is an approximation whether the migration can succeed, is feasible, and if the migration provides a benefit to the enterprise considering the costs of a migration. If a migration seems feasible, then the execution phase starts, which entails setting up and customizing the cloud infrastructure and eventually to deploy the system into the cloud.

3 Preparation of a Cloud Migration

The preparation phase of the migration to the cloud contains the most critical steps on the way to a cloud based system infrastructure. In this phase, the system which is to be migrated has to be evaluated on its fitness for migration. Furthermore, the most suiting cloud service and Virtual Machine (VM) configuration have to be decided upon. A set of tools support the decision-making process and will be described in more detail.

3.1 Cloud compatibility evaluation

The company conducting a migration should first evaluate to what extent and with what ease the system can be migrated. A modular system without critical dependencies on other local systems can be seen to be fitter for migration than a highly complex and intertwined system, whose functionality is strongly dependent on other local systems [3].

The company also needs to take into account security and privacy regulations and constraints that might make it impossible to migrate certain components of the system [3]. Therefore, the feasibility of a complete migration needs to be evaluated and if this seems to be not feasible, the migration of a only partial migration can be evaluated. This evaluation needs to be followed by a general decision on whether the whole system, only parts of it, or nothing at all will be migrated to the cloud.

[3] developed the Cloud Suitability and Alignment (CSA) Hierarchy, a scale along which a system's migration fitness can be evaluated and classified. The CSA Hierarchy is an analyzing tool, which checks a system's architecture for violations of Cloud Environment Constraints (CECs). CECs are constraints that were devised by [3] to check whether a system uses technologies that are not or only partially available in a cloud environment. If the system uses a technology that is not offered by any cloud service, it is said to have broken that particular CEC, which in turn lowers the system's overall fitness for cloud migration. The CSA Hierarchy classifies a system using the 5 categories as depicted in Figure 2.

Figure 2 shows the five levels of cloud compatibility on which an enterprise system can reside. According to the CSA, a system is analysed on whether it breaks any so called CECs, which are constraints that were devised by [3]. CECs are described in more detail in [4] and [5], but in general CECs are requirements set up by the cloud infrastructure. For example, if the enterprise system should be deployed to the Google App Engine, then a CEC would be that the system runs in a from Google App Engine supported sandbox environment like Java, Go, Python, or PHP [6].

The system can either comply with the CECs or otherwise will be said to violate the CEC. A violation can be *Warning, Critical, or Breaking*. Once the system is checked along the CECs, it can be categorized into one of the five CSA levels, which are described as follows as taken from [3]:

L0: **Cloud incompatible:** At least one CEC violation with severity *Breaking* exists.

L1: **Cloud compatible:** No CEC violations with severity *Breaking* exist.

L2: **Cloud ready:** No CEC violations exist.

L3: **Cloud aligned:** The execution context, utilized cloud services, or the migrated software system itself were configured to achieve an improved resource consumption or scalability without pervasively modifying the software system.

L4: **Cloud optimized:** The migrated software system was pervasively modified to enable automated exploitation of the clouds elasticity. An evaluation was conducted to identify system parts which would experience an overall benefit from substitution or supplement with offered cloud services. These substitutions and supplements were performed.

Based on how the system scores in this classification, a decision can be made about whether to migrate the system or not.

3.2 Cloud infrastructure service selection

In order to maximize the benefits from the migration to the cloud in terms of finance and provisioning, it is important to evaluate which cloud service provider offers the most suitable service for the enterprise system. The CloudGenius tool developed by [10] offers a decision support for doing exactly this.

In CloudGenius, in order to choose the most suitable cloud provider, the company first fills in a range of numerical options like preferences for e.g. hourly price, and service popularity followed by non-numerical options like preferences for e.g. operating system, location

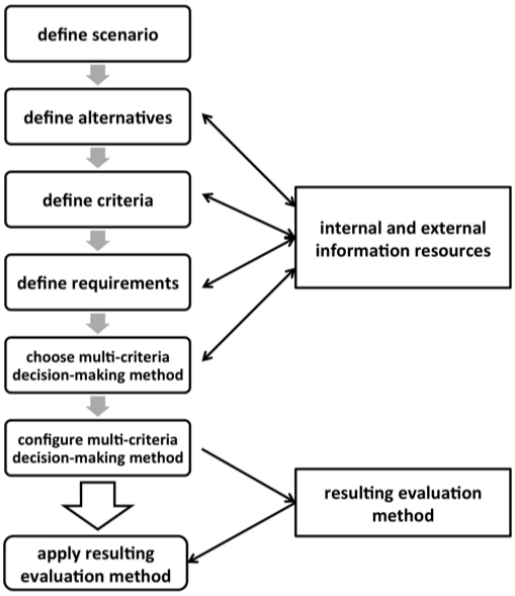


Fig. 3. Overview of the $(MC^2)^2$ Process taken from [10]

country, supported and implementation languages. The user can further specify a customized priority hierarchy of the entered options or choose to use the hierarchy proposed by CloudGenius itself[10].

The CloudGenius tool uses the $(MC^2)^2$ framework to weight and combine the entered preferences and calculates which service provider has the best fitting offer for the user. The $(MC^2)^2$ framework offers a method as displayed in Figure 3 which allows to evaluate alternatives for a solution based on a weighted set of requirements.

After the company has decided on a cloud service provider, the migration process moves to the next step, which is deciding on which virtual machines with which hardware and software specifications to use.

3.3 Cloud VM image selection

Once the company has entered preferences regarding the service of the cloud provider (e.g. hourly price, popularity), the next step is to specify requirements for the VMs on which the migrated system should run on eventually. CloudGenius offers a range of numerical options like e.g. CPU performance, maximal latency, RAM performance and offers a range of non-numerical options like e.g. Operating system and Virtualization Format as well [10]. The company can again order the preferences according to their importance.

Once all preferences regarding the service provider and virtual machine are entered, CloudGenius will use the $(MC^2)^2$ framework to compute combinations of service providers and their offered virtual machines to which fit best to the needs of the enterprise system and the company which wants to migrate the system. Figure 4 shows the integrated hierarchy of options which will be used to create a list of virtual machines and cloud providers will offer the best suiting product.

After CloudGenius computed and presented the list, the company is able to select a combination of virtual machine and cloud service and then, CloudGenius will automatically deploy the chosen combination for the company. With this, the decision phase of a cloud migration concluded and the actual migration of the enterprise system needs to be conducted.

4 Execution of a Cloud Migration

After deciding on the cloud service provider, virtual machine images and the foundation of the cloud system, the migration of the system is started. In the following three steps, the actual modifications and

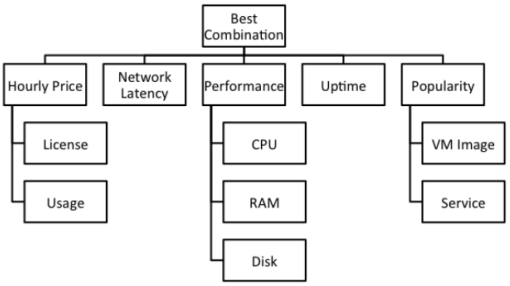


Fig. 4. CloudGenius Default Hierarchy taken from [10]

configurations are done to migrate the system or parts of the system into the cloud.

This is the latest point, a decision about how much of infrastructure should be moved into the cloud has to be made. This can be the whole system or also only parts of the system. This depends on the own requirements of the system and use cases. Instead of transferring the whole system, "hybrid cloud-based deployments" are also possible. This aspect is covered in later sections.

4.1 Cloud VM image customization

The basic customization includes the same configuration that has been done in the local environment. All required software and dependencies need to be installed and set up to provide their service. To a certain extend the cloud service providers also offer tools to import existing VM images into their service environment, but these have limitations. In most cases the VM images needs own customization in the cloud.

As [3] writes, scalability in the cloud is increased however there is no free lunch: "Running an existing application in the cloud does not imply relief of under- or over-provisioning concerns". A consequence might be a re-engineering of the software or even a redistribution of concerns.

The case study conducted by [3] shows that different combination of VM instances are able to fulfill their requirements. Furthermore, the more powerful the used VM instances the quicker the applications' response is. However, also the CPU utilization goes down and a continuous low CPU utilization is an indicator for over-provisioning (see Figure 5). Out of the six tested machines instances with the same software, the best matched instances almost reaches a 60% utilization, whereas overprovisioned machines only achieve around 20 to 40% average CPU utilization.

Technically every percent of unused CPU is too much, but since the load does vary over time, a certain buffer margin needs to be considered. As Figure 6 visualizes, the average CPU utilization during their experiment spiked up to 95% for a 2 minute time frame of the 24 minute experiment. Thus, a further increase in CPU utilization may lead to a delay in processing.

Over-provisioning is especially from a economical perspective undesirable. Every rented CPU - if used or not - costs money. However, to satisfy the customers, rather systems with a higher speed that are able to fulfill the request in time are used. So, to handle the load, systems that are also able to handle peaks in load are used.

Under-provisioning can easily be avoided by upgrading to a more powerful machine. The VM image stays the same and requires no further customization. To find the most optimal spot, some research and testing needs to be done. [9] suggests in general to use smaller instances to allow better scaling in small chunks to "avoid having underutilized servers". Additionally, some cloud service providers may offer dynamic scaling options. Then the cloud service provider takes over the responsibility for scaling the service up and down according to the current load on the system.

Since the machine instances offered by the cloud infrastructure providers differ, each VM image needs to be tailored towards their provided context. So, the choice of the cloud service provider has a

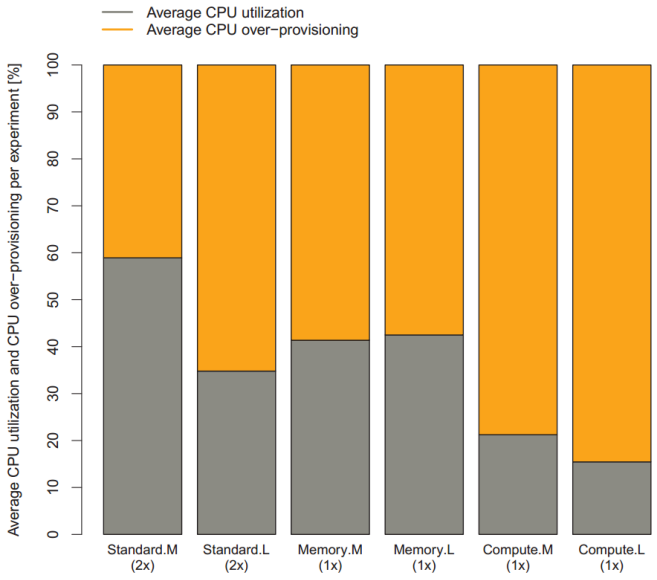


Fig. 5. Average CPU utilization of machines tested in the paper of [3]. The x-axes labels are various powerful machines with the amount of instances in parentheses.

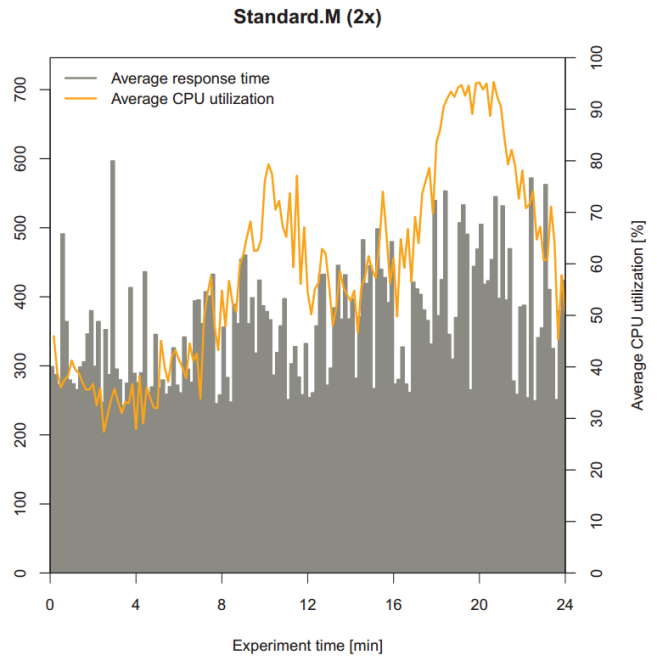


Fig. 6. CPU utilization of the Standard.M (2x) instance as described in [3] in their 24 minutes custom test.

huge impact in this step as the options and resources are already limited. This is represented as well in the pyramid style of Figure 1.

Therefore, also moving from one cloud service provider is not easy as the whole migration process needs to be done (again). However, since migration is a one time cost and the cost models of the providers do change, migration to another service is not impossible.

4.2 Migration strategy definition

Next, a strategy for the whole migration procedure has to be created. A strategy on which machines should be transferred into the cloud, how to handle data storage and transfer as well as the switch over period has to be created.

Whereas [3] only consider an all-or-nothing cloud migration approach, also hybrid cloud migrations are possible. Depending on for each project different individual factors, only certain systems are moved into the cloud. Others continue to operate at the local data center.

As [7] shows, hybrid systems are feasible within the requirements of the original infrastructure requirements. In their paper, the authors use two case studies to demonstrate that hybrid systems can offer the same functionality in a more flexible manner.

However, it should be noted that the approach of [7] recommends a complete replica of the services in the cloud. Services then exist in the local and the cloud environment. This keeps the latency of requests within a data center low - for example for database requests. If one requirement is to keep the database or other services in the local data center due to privacy reasons then the impact on latency and response time should be considered.

One part of the offered flexibility is the automatic scaling of the instances. So, instead of having 24/7 the same amount of instances running regardless of their usage, schedules can be created to automatically spin up and down instances depending on the time of the day - which is interesting for services that are only used at certain hours. For example, in the E-Commerce business, people usually go shopping before or after work and to a certain extend also during their lunch break. Therefore automatic adjustment can solve under- and over-provisioning.

Moreover, also complete automatic scaling based on the actual traffic is offered as well. As this simplifies resource allocation even more, the risk in case of misconfiguration or unexpected traffic rises as well, which can lead easily to bills of unexpected heights.

Auto-scaling based on schedules is only effective, when the average usage schedule of the user is known. Assuming that the users are distributed equally over the globe, than the load on the systems should be the same throughout one day. However, with focuses on certain geographical regions, a suitable and effective schedule can be created.

In any case, when the location of the user is known, then it can be used to its advantage. Since all the big cloud service providers offer multiple regions around the globe to deploy an application in or to deploy in all locations, the response times are kept low by using the service closest to them. For users that are accessing a service from a non-local location (Internet not intranet), the response times can even be faster - due to the possible optimization done in the background by the cloud service provider that are not done in simple application setups like content delivery networks (CDN).

For users who were using an intranet service before the migration, the general response times increase due to a longer network route. However, if the request processing takes a lot more time in computation (e.g. through database retrieval), then the added time is insignificant. On the other hand, also a cache proxy or a local mirror of the cloud service database can be introduced to reduce the impact on response time.

Especially for websites with an international audience, cloud service providers provide the opportunity to improve the user experience through faster content delivery because they offer multiple computing centers around the globe. Still the issue of data synchronization is not addressed and if that is the bottleneck, it will also exist in the cloud environment.

In general, hybrid solutions can also create longer network routes, because local systems may communicate with cloud systems in a distant data center. To avoid these long network routes, [7] introduces first a "flexible routing approach [...] based on the location of an application component or user". Although they see the difficulties in implementation, this can keep all requests in the local environment, which can be own data center or the cloud service. Through a wise organization of the system, that is having each service running at least once in each data center, the required communication between two data center can get limited.

Since additional communication for tasks like synchronization are necessary, these have to be taken into account, because it introduces extra fees on the bandwidth. [7] does include this in their model.

Another aspect that needs to be considered is, how to move all the existing data into the cloud. Mostly gigabytes and terabytes of data exist, which need to be moved as well. Since the upload of the data takes in most cases too long, the cloud service providers offer services especially for that which include picking up physical copies of the data from which the data can be copied directly into the cloud system. The idea for the data upload or download is so simple that [11] wrote: "Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway".

Among others, [7] points out that "the migration process may involve one-time cost", which are later compensated by the operational cost over multiple months as their findings and also the findings of [9] show.

After all, it is decided on a concrete strategy, which can get put into place without further decision making.

4.3 Migration strategy application

To complete the migration process, the migration strategy needs to get put in action and then the final adjustments can be done. This mainly includes configuration, but also transferring security policies into the cloud application and policies for service failure and data loss.

A very important point to consider during the cloud migration process is data security. Usually firewalls are used to exclude unwanted visitors or intruders. These can and are implemented the same way in cloud environments. Also access control lists (ACLs) are migrated similar to define access rights for all involved parties. A good practice is to generally permit traffic, except for "explicit" rules [7].

When the system is split into multiple data centers, like it is the case in a hybrid cloud migration model, then each data center needs protection by a firewall. Extra rules need to be added as well in the migration process to allow the communication between the data centers. [7] shows that these rules do not have to implemented in a "naive approach" and their "approach can scale well to large networks" according to the authors.

During the migration, the privacy should again be taken into consideration as well. Although in the previous preparation the privacy was already taken into account and especially in hybrid cloud solution also a decision has is on which system to migrate to the cloud and which to keep in the local data center, a (re-)validation is necessary. Not only the data itself needs to be protected, also the trace that the data leaves. This can be cache and logs files that potentially disclose private information.

To address the issue of data security, the cloud service provider has to be trusted to protect its data from intruders. They also put data security statements in the SLAs, but still the data is then not (only) available in a local, secure context. If the data should rather be kept in a local data center - mainly because of a lack of trust - then hybrid cloud solutions are an option to still allow a migration towards the cloud. However, when keeping one service in just one location, the advantage of increased response speed is lost. Also the flexibility in scaling is gone and so an over-provisioning necessary to avoid resource limitations.

Also, data duplication and backups are different at each cloud service provider. The SLAs define policies, but they might not fit the systems specific needs. Then the local data centers backup policies have to be adapted to the cloud service providers.

5 Conclusion

As we have seen in our analysis of the migration process, many steps that need to be taken are facilitated by scientific tools. Especially the preparation phase is supported well by CloudGenius ([10]), which takes over much of the decision making and presents ready-to-deploy cloud based virtual machines. Also this tool makes it easy to calculate the financial benefits of a migration to a cloud service and helps assessing the costs and benefits of the migration.

The addition of the cloud fitness evaluation step in the preparation ensures a better understanding of the software that is being migrated and enforces a feasibility check whether the software can function in a cloud environment.

In the migration phase, we have seen that many steps and decisions are only partially supported by tools, which is caused by the very unique nature of each system that needs to be migrated. However, [3] and [7] support this process to a decent extent by offering advice on whether to migrate the whole system or only parts of it and how to ensure the same level of security and privacy in the cloud as in the local infrastructure.

Under the precondition that a system is compatible for cloud migration according to the CSA hierarchy, a migration can be successfully performed. Although the cloud service providers might limit the enterprises to a certain extend in their offered flexibility, the same or similar steps have been already performed for the setup of the system as in the local data center.

The in our work proposed roadmap covers the steps for a cloud migration and also takes into account security and privacy concerns. As the complete migration process is a complex topic and many factors have to be considered, also manual configuration is still necessary as the tools at the moment are not capable to providing a full automatic migration.

6 Future Work

Many tools we have described are not fully or at least semi-automatic, which means that much work needs to be done by the entity that wants to implement a cloud migration. We suggest that the research field focuses on how tools like CloudMig ([3]) and Cloudward ([7]) can be automatized to a bigger extent to simplify the process of a cloud migration. Also, an automatization of the customization of the virtual machines suggested by e.g. CloudGenius would decrease the work needed for a successful migration significantly.

The option of migrating only a part of a system to the cloud might seem more appealing for privacy and security concerned enterprises than uploading sensitive data or system components to the system of a third-party. The decision of which system parts should be migrated and which not is supported by the CloudMig model, and still needs a lot of manual consideration from the side of the migrating company. The research field needs to devise a more comprehensive and compact decision guide for supporting such considerations.

As mentioned in [8] as well, we suggest that the research field for cloud migration needs to investigate more how decisions on architectural changes to make a system cloud compatible can be supported and eventually automatically executed.

Acknowledgements

The authors thank R. Bwana, R. Brandt and the anonymous reviewers for their work in providing feedback to improve this paper.

References

[1] V. Andrikopoulos, S. Strauch, and F. Leymann. Decision support for application migration to the cloud: Challenges and vision. In *Proceedings of the 3rd International Conference on Cloud Computing and Service Science, CLOSER 2013, 8 -10 May 2013, Aachen, Germany*, pages 149–155. SciTePress, 2013.

[2] P. V. Beserra, A. Camara, R. Ximenes, A. B. Albuquerque, and N. C. Mendonça. Cloudstep: A step-by-step decision process to support legacy application migration to the cloud. In *Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), 2012 IEEE 6th International Workshop on the*, pages 7–16. IEEE, 2012.

[3] S. Frey and W. Hasselbring. The cloudmig approach: Model-based migration of software systems to cloud-optimized applications. *International Journal on Advances in Software*, 4(3 and 4):342–353, 2011.

[4] S. Frey and W. Hasselbring. An extensible architecture for detecting violations of a cloud environment's constraints during legacy software system migration. In *Software Maintenance and Reengineering (CSMR), 2011 15th European Conference on*, pages 269–278. IEEE, 2011.

[5] S. Frey, W. Hasselbring, and B. Schnoor. Automatic conformance checking for migrating software systems to cloud infrastructures and platforms. *Journal of Software: Evolution and Process*, 25(10):1089–1115, 2013.

[6] Google Inc. The app engine standard environment. `https://cloud.google.com/appengine/docs/standard/`, 2017.

[7] M. Hajjat, X. Sun, Y.-W. E. Sung, D. Maltz, S. Rao, K. Sripanidkulchai, and M. Tawarmalani. Cloudward bound: planning for beneficial migration of enterprise applications to the cloud. In *ACM SIGCOMM Computer Communication Review*, volume 40, pages 243–254. ACM, 2010.

[8] P. Jamshidi, A. Ahmad, and C. Pahl. Cloud migration research: a systematic review. *IEEE Transactions on Cloud Computing*, 1(2):142–157, 2013.

[9] A. Khajeh-Hosseini, I. Sommerville, J. Bogaerts, and P. Teregowda. Decision support tools for cloud migration in the enterprise. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 541–548. IEEE, 2011.

[10] M. Menzel and R. Ranjan. Cloudgenius: decision support for web server cloud migration. In *Proceedings of the 21st international conference on World Wide Web*, pages 979–988. ACM, 2012.

[11] A. Tanenbaum and D. Wetherall. *Computer Networks*. Prentice-Hall. Englewood Cliffs [etc.], 1989.

[12] V. Tran, J. Keung, A. Liu, and A. Fekete. Application migration to cloud: a taxonomy of critical factors. In *Proceedings of the 2nd international workshop on software engineering for cloud computing*, pages 22–28. ACM, 2011.

Techniques for the comparison of public cloud providers

Frans Simanjuntak, Marco Gunnink

Abstract—Cloud computing is the delivery of computing services over the Internet. It enables users to deploy applications on hardware provided by external providers, rather than building and maintaining their own servers. Cloud computing rose to popularity from 2006 with the introduction of Amazon Elastic Compute Cloud, Microsoft's Azure and the Google App Engine.
Cloud computing offers many benefits, such as cost, scalability, rapid deployment and accessibility. While most public cloud providers offer similar functionality, there are still many differences between the features they provide, their performance and cost. To help users select their ideal cloud provider, several techniques for the comparison of public cloud providers have been researched and developed. In this paper we compare three of these techniques: Analytic Hierarchy Process (AHP), CloudCmp and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS).

Index Terms—Cloud computing, techniques, comparison, performance, cost, AHP, TOPSIS, CloudCmp.

1 Introduction

Over the past decade, cloud computing has become very popular [5]. And as the demand for cloud computing rises, so does the supply of public cloud providers. The main advantages of cloud computing are cost efficiency and usability [4]. Cloud providers operate on a pay-as-you-go model: users only pay for the resources they used. Users of public clouds can avoid large investments in hardware, upgrades and maintenance. This is especially beneficial for systems that only need to be used for a short time, such as those for research projects.

There are other benefits as well: cloud computing systems can be scaled up or down according to immediate demand. They provide virtually unlimited resources which can be used and paid for as needed. This also offers rapid deployment. New applications can be deployed quickly and upgraded as needed. The user can easily adapt the virtual hardware to the application's requirements when they change.

Cloud computing also offers benefits to the cloud providers. They can use their existing computing infrastructure more efficiently while gaining extra sources of revenue. Finally, use of cloud computing in large data centers helps reduce energy costs and pollution [1].

There are, however, also downsides to cloud computing. Access to the cloud computing systems depends on Internet connectivity: if there is an Internet outage at either the user or the cloud provider, the service cannot function. The user also loses some control and flexibility by using externally managed hardware. The available functionality depends almost entirely on what the cloud provider can offer and if it becomes inadequate the user may have to migrate their system to another provider. Finally, it may be difficult to ensure sufficient security for the user's system because it is always connected to the Internet and they often share the hardware with other users [10].

As we mentioned, there are many companies offering public cloud computing services. Choosing the most suitable one for a business' need is not easy. Different cloud providers offer different trade-offs in performance, cost and functionality. Comparing them is difficult, since a lot depends on what the business needs while all cloud providers attempt to make themselves appear attractive to all potential customers. Despite this, several techniques for comparing public cloud providers have been researched and developed, such as: Analytic Hierarchy Process (AHP), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), ELECTRE, CSP (Cloud Service Provider), CloudCmp framework, Fuzzy Inference System (FIS), Multi Attribute Group Decision Making (MAGDM). In this paper we compare AHP, CloudCmp and TOPSIS to determine the most suitable comparison technique.

- *Frans Simanjuntak, s3038971, E-mail: f.simanjuntak@student.rug.nl.*
- *Marco Gunnink, s2170248, E-mail: m.gunnink@student.rug.nl.*

We chose AHP and TOPSIS since they offer powerful techniques for decision making, which have already been proven and applied in other fields. CloudCmp, on the other hand is very new, but specific to the comparison of cloud providers. We are interested to see how these techniques can be combined for the selection of public cloud providers.

2 Description of Methods

In this section we will give the details of the three techniques used for the comparison of public cloud providers.

2.1 Analytic Hierarchy Process (AHP)

This section explains Analytic Hierarchy Process (AHP) method for decision making.

2.1.1 Basic Theory of AHP

Analytic Hierarchy Process (AHP) is an effective method for dealing with complex decision making. It is based on the experience gained by its developer, T.L. Saaty, while directing research projects in the US Arms Control and Disarmament Agency [2]. The theory behind this method is to derive the ratio scales from paired comparisons. The actual measurement such as price, weight, etc., or the subjective opinion such as satisfaction feeling and preference will be used as an input.

AHP generates a weight for each evaluation criterion according to the decision makers pairwise comparisons of the criteria. The higher the weight, the more important the corresponding criterion. Next, for a fixed criterion, AHP assigns a score to each option according to the decision makers pairwise comparisons of the options based on that criterion. The higher the score, the better the performance of the option with respect to the considered criterion. Finally, AHP combines the criteria weights and the option scores, thus determining a global score for each option, and a consequent ranking. The global score for a given option is a weighted sum of the scores it obtained with respect to all the criteria [11].

Even though AHP is known as a very flexible and powerful tool, it may require a large number of evaluations by the user, especially for problems with many criteria and options. In order to reduce the decision maker's workload, AHP can be completely or partially automated by specifying suitable thresholds for automatically deciding some pairwise comparisons. In order to obtain the final results, AHP performs sequential steps as follows [2]:

- *Step 1*: The problem is decomposed into a hierarchy of goal, criteria, sub-criteria and alternatives.
- *Step 2*: Data is collected from experts or decision-makers corresponding to the hierarchic structure, in the pairwise comparison of alternatives on a qualitative scale. Experts can rate the comparison as equal, marginally strong, strong, very strong, and

extremely strong. The opinion can be collected in a specially designed format as shown in Figure 1.

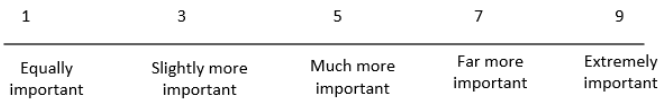


Fig. 1: Pairwise comparison [11]

- *Step 3*: The pairwise comparisons of various criteria generated at step 2 are organized into a square matrix. The diagonal elements of the matrix are 1. The criterion in the *i*th row is better than criterion in the *j*th column if the value of element (i, j) is greater than 1; otherwise the criterion in the *j*th column is better than that in the *i*th row.The (j,i) element of the matrix is the reciprocal of the (i, j) element.

$$A_n = (a_{ij})_{nxn} \tag{1}$$

- *Step 4*: The principal eigenvalue and the corresponding normalized right eigenvector of the comparison matrix give the relative importance of the various criteria being compared. The elements of the normalized eigenvector are termed weights with respect to the criteria or sub-criteria and ratings with respect to the alternatives.

- *Step 5*: The consistency of the matrix of order n is evaluated. Comparisons made by this method are subjective and AHP tolerates inconsistency through the amount of redundancy in the approach. If this consistency index fails to reach a required level then answers to comparisons may be re-examined. The consistency index, CI, is calculated as

$$CI = \frac{(\lambda_{max} - n)}{(n-1)} \tag{2}$$

where λ_{max} is the maximum eigenvalue of the judgement matrix. This CI can be compared with that of a random matrix, RI.

$$RI = \frac{(\lambda_{max}{}' - n)}{(n-1)} \tag{3}$$

The ratio derived is termed the consistency ratio, CR. Saaty suggests the value of CR should be less than 0.1 (see Table **??**).

$$CR = \frac{CI}{RI} \tag{4}$$

- *Step 6*: The rating of each alternative is multiplied by the weights of the sub-criteria and aggregated to get local ratings with respect to each criterion. The local ratings are then multiplied by the weights of the criteria and aggregated to get global ratings.

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Table 1: Random Consistency Index [11]

Finally, AHP produces weight values for each alternative based on the judged importance of one alternative over another with respect to a common criterion.

2.1.2 AHP and Cloud Computing

Cloud computing is a paradigm for providing all sorts of cloud services, that can be rapidly deployed and delivered with minimal management cost and effort or reciprocity among service providers [8]. This paradigm has boosted many providers to release new cloud services in order to attract potential costumers. Unwittingly, the increasing growth of services among providers may be identical or even similar with different characteristic. This is, of course, difficult for the potential customers to judge and weight the one that balances their requirements best. As multiple criteria are involved during the selection of cloud services, selecting the best one can be difficult due to multiple consumer requirements and this process could be very time-consuming and error-prone.

Being the most widely used technique for complex decision making, AHP is one of the alternatives to overcome this problem. As mentioned earlier that AHP is intended for multi criteria decision making, therefore it is suitable to be applied. Since AHP computes the results based on criteria, first of all, we have to define the criteria for cloud computing.

There are two ways of defining criteria, one can be obtained from the user perspective and the other from the element of services offered by providers. If we decide to set the criteria based on the the user perspective, we must consider the things that really matter to them, e.g response time, throughput, availability, reliability, and cost. On the other hand, if we concern about the elements of cloud services, we can define the criteria based on the attributes they have in common, e.g price/hour, virtual core, memory, CPU performance, Disk I/O consistency, Disk Performance, and Memory Performance. After defining the criteria, the rest of AHP steps can be performed.

2.2 CloudCmp Framework

CloudCmp is a comparison framework focused primarily on providing a broad comparison tool that can be applied to many cloud providers while still providing useful data to cloud users [6]. To this end the developers of CloudCmp have identified a set of common services offered by cloud providers and used by cloud users. These services are: elastic compute cluster, persistent storage, intra-cloud network and wide-area network. For each service they implemented a set of performance metrics and tested them on four public cloud providers: Amazon AWS, Microsoft Azure, Google App Engine and Rackspace CloudServers. The next sections describe the performance metrics per cloud computing service.

2.2.1 Elastic compute cluster

An elastic compute cluster hosts and runs application code. The number of virtual instances can be scaled up or down, either by a request of the application or automatically by the cloud provider in response to the processor usage. For the elastic compute cluster, CloudCmp measures the following performance metrics:

- **Benchmark finishing time** To measure the CPU, memory and local disk I/O performance, CloudCmp uses a modified SPECjvm2008 [13] benchmark. Java was chosen because it is available on all measured providers. The benchmark is run on each cloud platform and the time to finish is used to compare the providers. For those that offer multi-threading capabilities a parallel version of the benchmark is also run.

- **Cost** CloudCmp uses the published prices or billing APIs to obtain the cost for running the benchmark.

- **Scaling latency** The scaling latency is the time it takes to request and launch a new virtual instance. CloudCmp divides this into two segments. The first segment is the time from when the new instance is requested and it is created and powered on. The second is the time between powering on and the instance being ready to use.

2.2.2 Persistent storage

Three types of persistent storage are offered by the cloud providers that CloudCmp looks at: table, blob and queue, though not all providers provide all three types. Tables provide storage for structural data, similar to the traditional relational database systems, but often with only limited support for complex queries, such as joins and grouping. Blob storage is designed for large unstructured chunks of data, like binary objects. Finally queues provide a way to send small amounts of data between different instances. CloudCmp uses the following metrics to compare storage performance:

- **Operation response time** The operation response time is measured from the time an operation is requested until the last byte reaches the client.
- **Time to consistency** The cloud providers measured by CloudCmp offer automatic replication for fault tolerance and high availability. However, this comes at the cost of consistency: a read operation that immediately follows a write may still return old or stale data. CloudCmp measures the time to consistency by continuously reading from the persistent storage until the latest data is returned.
- **Cost per operation** The cost per operation is also measured for persistent storage, using the published prices or the available billing APIs.

2.2.3 Intra-cloud & wide-area network

Since cloud computing services are always provided over the Internet, network connectivity and performance are of major importance. CloudCmp distinguishes between the intra-cloud and wide-area networks. The former connects instances within the same cloud to each other while the latter connects the cloud instances to the rest of the Internet. The following performance metrics are measured by CloudCmp for the intra-cloud and wide-area networks:

- **Throughput** CloudCmp uses TCP throughput to measure the path capacity: the maximum bandwidth between two endpoints. The throughput is measured both with the default TCP window size and a maximum TCP window size of 16MB to minimize the overhead.
- **Latency** The network latency is measured using the round-trip time between the instances within the cloud and to various locations on the wide-area network. CloudCmp uses the PlanetLab network [15] to measure latency from different physical locations.

2.3 Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

2.3.1 Basic Theory of TOPSIS

TOPSIS is a multi attribute decision making technique that sorts the alternatives for a decision problem which was introduced by Hwang and Yoon in 1981. The basic principle is that the selected alternative should have the shortest distance from the ideal solution and the farthest distance from the negative ideal solution in geometrical sense [12]. The TOPSIS method consists of the following steps [14]:

- *Step 1*: Determine the decision problem and identify the relevant evaluation criteria.
- *Step 2*: Develop the preference for the criteria by assigning them weight.
- *Step 3*: Construct the Decision Matrix for the alternatives based on the criteria.

$$DM = \begin{array}{c} \\ L_1 \\ L_2 \\ . \\ . \\ . \\ L_n \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & ... & C_n \end{array} \\ \begin{bmatrix} X_{11} & X_{12} & ... & X_{1n} \\ X_{21} & X_{22} & ... & X_{2n} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ X_{m1} & X_{m2} & ... & X_{mn} \end{bmatrix} \end{array} \tag{5}$$

  Where i is the criterion index ($i = 1..m$); m is the number of potential sites and j is the alternative index ($j = 1..n$). The elements $C_1, C_2..C_n$ refer to the criteria: while $L_1, L_2..L_n$ refer to the alternative locations. The elements of the matrix are related to the values of criteria i with respect to alternative j.

- *Step 4*: Calculate a normalized decision matrix for the above Decision Matrix.

$$NDM = R_{ij} = \frac{X_{ij}}{\sqrt{\sum_{i=1}^{m} X_{ij}^2}} \tag{6}$$

- *Step 5*: Compute the weighted normalized decision matrix (NDM) by multiplying weights of criteria with the corresponding alternatives value.

$$V = V_{ij} = W_j \times R_{ij} \tag{7}$$

- *Step 6*: Identify the Positive Ideal Solution

$$PIS = A^+ = \{V_1^+, ..., V_n^+\} = \{(max_j(V_{ij})|i \in I'), (min_j(V_{ij})|i \in I'')\} \tag{8}$$

  where I′is associated with benefit criteria, and I″is associated with cost criteria.

- *Step 7*: Identify the Negative Ideal Solution

$$NIS = A^- = \{V_1^-, ..., V_n^-\} = \{(min_j(V_{ij})|i \in I'), (max_j(V_{ij})|i \in I'')\} \tag{9}$$

  where I′is associated with benefit criteria, and I″is associated with cost criteria.

- *Step 8*: Compute separation of each criteria value for each alternative from both ideal and negative ideal solution.

$$S^+ = \sqrt{\sum_{j=1}^{n} (V_j^+ - V_{ij})^2} \quad i = 1, ..., m \tag{10}$$

$$S^- = \sqrt{\sum_{j=1}^{n} (V_j^- - V_{ij})^2} \quad i = 1, ..., m \tag{11}$$

- *Step 9*: Measure the relative closeness of each location to the ideal solution.

$$C_i = S_i^- / (S_i^+ + S_i^-), \quad 0 \leq C_i \leq 1 \tag{12}$$

- *Step 10*: Rank the preference order. According to the value of C_i the higher the value of the relative closeness, the higher the ranking order and hence the better the performance of the alternative. Ranking of the preference in descending order thus, allows relatively better performances to be compared.

2.3.2 TOPSIS and Cloud Computing

Since cloud service selection is a multiple criteria group decision-making (MCDM) problem, TOPSIS can be an alternative method that helps user to choose an ideal cloud provider. It can assist service consumers by analyzing available services using fuzzy opinions. Fuzzy TOPSIS is a method that can help in objective and systematic evaluation of alternatives on multiple criteria.

Similar to AHP, the first step to do is to define the criteria as input parameters. There are two ways of defining criteria either from the user perspective or from the attributes of public cloud providers as mentioned in section 2.1.2. After the criteria is defined, the rest of the TOPSIS steps can be performed.

3 Results

This sections lists the result of the comparison of public cloud providers using different techniques, namely AHP, CloudCmp Framework, and TOPSIS.

3.1 AHP

For the purpose of the analysis with AHP model, we studied and followed the experiment conducted by Mingrui.Sun et.al from the Harbin Institute of Technology, China [7]. The goal of the experiment was to select a cloud service for health medical rehabilitation system. In this experiment, the rehabilitation therapy system was deployed on three different cloud platforms. The users which were stroke patients can access these systems according to their personal preference and the demand of treatment. The experiment was expected to capture the paired comparison of criterion layer with respect to goal, the paired comparison of alternative layer with respect to criterion layer, and finally the comparison results of the alternatives.

In order to perform the AHP steps, first we have to compose the cloud computing selection problem into a hierarchy of goal, criteria and alternatives which subsequently used as the basis of performing the remaining steps of AHP. In this case, the hierarchy goal was to select a cloud service, the main criteria is the quality attributes which are assessed based on the user perspective such as the response time, throughput, availability, reliability, and cost and the alternative layers are the available cloud service providers[7].

After defining the goal, the main criteria, and the alternatives, then the pairwise comparison can be constructed followed by the organization of results into a square matrix. By using this square matrix, the rest of the steps can be performed. The results of the paired comparison matrix of criterion layer with respect to the goal is listed in Table **??** and the same pattern applies to criterion layer. The result of the paired comparison matrix of alternative layers with respect to each criterion layer is listed in Table 3. Finally, the weight of each cloud provider can be obtained by summing all weights of each alternative divided by the total number of criteria as listed in Table 4.

CR=0.0944<0.1;		**Global Weight=1.0000;**			**λ_{max}=5.4227**	
Select Service Cloud	Response Time	Throughput	Availability	Reliability	Cost	w_i
Response Time	1	1	0.1667	0.125	0.3333	0.045
Throughput	1	1	0.1429	0.11111	0.1667	0.0371
Availability	6	7	1	1	7	0.03791
Reliability	8	9	1	1	9	0.444
Cost	3	6	0.1429	0.1111	1	0.0947

Table 2: Paired comparison matrix criterion layer with respect to the goal [7]

From the results listed in Table 4, it can be seen that the value of CS3 is greater than CS1 and the value of CS1 is greater than CS2, therefore the cloud service 3 would be selected as an ideal cloud provider since the weight is higher than the others. The higher the weight, the more qualified the cloud service.

3.2 CloudCmp

Li *et al.* ran CloudCmp on AWS, Azure, App Engine and CloudServers during March, April and May of 2010. Rather than replicating all the results here, we give a short overview of the tests they could run

CR=0.0944<0.1;		**Global Weight=0.045;**		**λ_{max}=5.4227**
Response Time	CS1	CS2	CS3	w_i
CS1	1	5	3	0.6483
CS2	0.2	1	0.5	0.122
CS3	0.3333	2	1	0.2297
CR=0.0904<0.1;		**Global Weight=0.0371;**		**λ_{max}=3.0940**
Throughput	CS1	CS2	CS3	w_i
CS1	1	0.2	4	0.1991
CS2	5	1	8	0.7334
CS3	0.25	0.125	1	0.0675
CR=0.0053<0.1;		**Global Weight=0.3791;**		**λ_{max}=3.0055**
Availability	CS1	CS2	CS3	w_i
CS1	1	5	4	0.6908
CS2	0.2	1	1	0.1488
CS3	0.25	1	1	0.1603
CR=0.0707<0.1;		**Global Weight=0.4440;**		**λ_{max}=3.0735**
Reliability	CS1	CS2	CS3	w_i
CS1	1	3	0.1667	0.1667
CS2	0.3333	1	0.125	0.0726
CS3	6	8	1	0.7612
CR=0.0000<0.1;		**Global Weight=0.0947;**		**λ_{max}=3.0000**
Cost	CS1	CS2	CS3	w_i
CS1	1	3	1	0.4286
CS2	0.3333	1	0.3333	0.1429
CS3	1	3	1	0.4286

Table 3: Paired comparison matrix alternative layer with respect to the criterion layer [7]

Alternatives	**Weight**
CS1	0.4129
CS2	0.1349
CS3	0.4522

Table 4: Cloud Service Evaluation using AHP [7]

for which providers. We then show the case studies they ran and the results obtained.

Due to legal concerns Li *et al.* anonymized the identities of the providers and indicate them as C_1 through C_4. Additionally, for the providers that offer them, they tested the framework on different tiers, indicated by $C_j.i$ for tier i of provider j, where lower numbers of i indicate slower and cheaper instances and higher values indicate faster and more expensive tiers. Provider C_3 only offers a single performance tier.

On the elastic compute cluster they tested both single- and multi-threaded programs, except on provider C_3, which does not support multi-threading. C_3 also does not support local disk I/O, so that was not tested on C_3 either. They measured the scaling latency for providers C_1, C_2 and C_4 as C_3 does not offer manual instance requests.

For the persistent storage the table services of providers C_1, C_3 and C_4 were compared by measuring the performance of `get`, `put` and `query` operations. Li *et al.* also measured the time to consistency, but found that only C_1 exhibited any inconsistency. They furthermore tested the blob storage on all providers, except C_3. And finally they tested the queue service on providers C_2 and C_4.

Finally, they measured the latencies of the intra- and inter-datacenter networks and the wide-area network. Since C_3 does not allow direct communication between instances, intra- and inter-datacenter latency was not measured for it. On the other hand, for the wide-area network, C_3 has the best performance, due to the fact that it offers automatic load-balancing. C_2 has the worst round-trip time, because they have fewer data-centers and all located relatively close to each other.

3.2.1 Case Studies

In addition to separate specific tests for the above-mentioned performance metrics, CloudCmp also measures the cloud performance with more integrated case studies. These case studies are intended to show the cloud performance in a more realistic scenario. The three case studies are an E-commerce website, a Parallel Scientific Computation

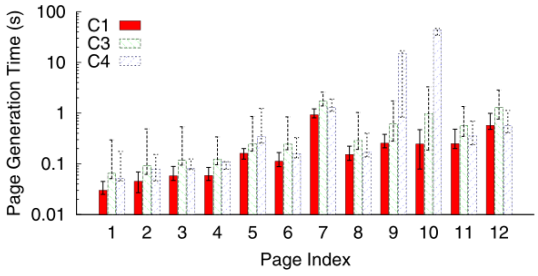


Figure 13: The page generation time of TPC-W when deployed on all three cloud providers that support it. The y-axis is in a logarithm scale.

Fig. 2: Page generation of TPC-W [6]

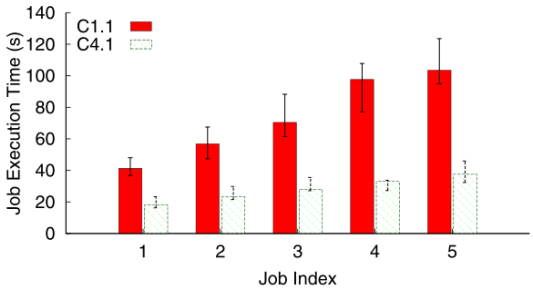
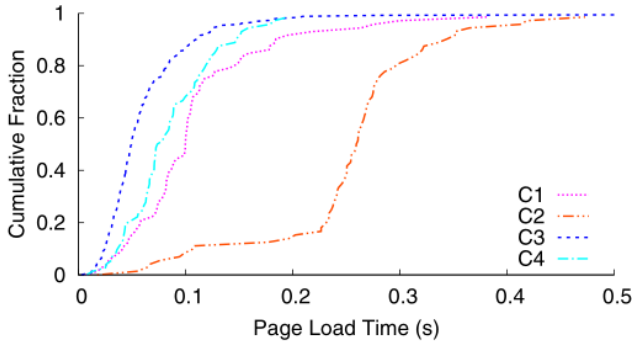
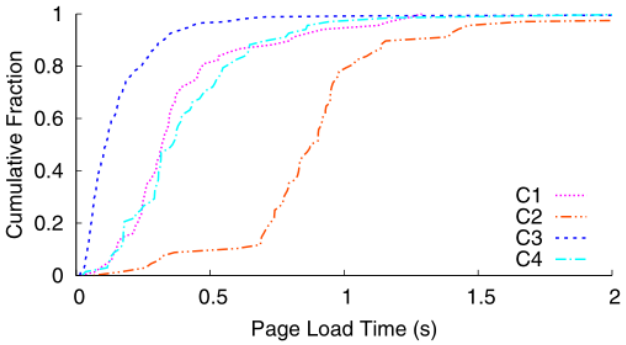


Figure 14: The job execution time of Blast when deployed on both C_1.1 and C_4.1. We show the results for five example jobs.

Fig. 3: Job execution time on Blast [6]



(a) 1KB Page



(b) 100KB Page

Fig. 4: Distribution of page download time [6]

application and a Latency Sensitive website.

The E-commerce website is tested with TPC-W [3], a standard benchmark for transactional web services. The E-commerce website mostly measures persistent storage speed as the goal is to minimize page generation time.

TPC-W was not deployed on C_3 because it does not offer a table service. Figure 2 shows the page generation time for the various pages it implements. Provider C_1 has the lowest generation time on all pages, consistent with the measurements from the table service. C_4 is much slower on pages 9 and 10 which contain many `query` operations. This too is consistent with earlier measurements, where C_4 is also shown to be slower on `query`s.

For the Parallel Scientific Computation, CloudCmp uses Blast [9], an application for DNA alignment. Blast is very CPU-intensive and therefore mainly assesses the performance of the Elastic Compute Cluster. It has one instance to accept jobs and uses a message queue to send the jobs to worker instances.

Since Blast requires a queue service, it was only tested on providers C_1.1 and C_4.1. The execution times are much lower on C_4.1, indicating it has much better computing performance than C_1.1. This was also the conclusion from the CloudCmp elastic compute cluster measurements.

Finally, the Latency Sensitive website is implemented with a simple web server that serves static pages. For CloudCmp, Li *et al.* downloaded two pages: a small one of 1KB and a large one of 100KB. The pages are downloaded from various locations, using PlanetLab. This case study evaluates the wide-area network latency.

Figure 4 shows the distribution of the page downloading times for all four providers. C_3 stands out as the fastest one, for both the small and large page. This comes as no surprise, since C_3 was also shown to have the lowest wide-area latency.

3.3 TOPSIS

In order to prove the capability of TOPSIS model as one of the alternative methods for selecting cloud providers, we reused the case study of AHP. Firstly we studied the experiment conducted by Mingrui.Sun et.al [7]. We notice that we can reuse some elements from the case study such as the criterion, the alternatives and also the weights which already listed in Table 3.

Reusing the AHP weights is indeed allowed because both AHP and TOPSIS have similar preliminary steps. However, the other input parameter such as the criterion parametric dataset cannot be obtained directly from the paired comparison matrix of alternative layer because TOPSIS works in different ways after the first step done. So, we analyzed the matrix of alternative layer and we made some adjustments towards its value. After that, we created a dataset 5 and constructed a decision matrix. This decision matrix was used as a base for executing the rest of the steps.

Criteria	**Alternatives**		
	CS1	CS2	CS3
Response Time	8	5	7
Throughput	7	9	5
Availability	8	5	6
Reliability	6	4	9
Cost	6	7	6

Table 5: Criterion Parametric Dataset

After applying the TOPSIS method using the above dataset combined with the criterion weight taken from AHP, it turns out the relative closeness value of CS3 is greater than the others, 0,957844269 to be exact (See Table6). This made the CS3 as the most preferable cloud provider followed by CS1 and CS2 respectively. Furthermore, the output of this experiment indicates that both TOPSIS and AHP produced similar results.

Alternatives	Relative Closeness
CS1	0.525435122
CS2	0.070962879
CS3	0.957844269

Table 6: Cloud Service Evaluation using TOPSIS

4 Discussion and Conclusion

In this paper, we have explained three techniques for the comparison of services offered by cloud providers: AHP, CloudCmp, and TOPSIS. We evaluated those techniques using case studies obtained from the literatures. Since AHP and TOPSIS are intended for multi-criteria decision analysis, we used similar cases when assessing them. However, it's not possible for us to use similar cases when it comes to evaluating CloudCmp because the evaluation properties are different compared to AHP and TOPSIS.

The results obtained from the studies show that both AHP and TOPSIS generated similar results. This phenomenon, of course, is easily explained since these techniques are meant to solve complex decision making. AHP uses a hierarchical structure and systematization in order to solve the cloud services selection problem and TOPSIS performs analysis of available cloud services using fuzzy opinions.

Although AHP is an effective tool for decision making, it has disadvantages, such as the decision maker's subjectivity, which can yield uncertainties when determining pairwise comparisons.

Then, we looked at CloudCmp. As mentioned, CloudCmp works significantly differently from the other two techniques. It sacrifices depth for broadness by only comparing common functionality, offered by all cloud providers. Despite their effort to be as broad as possible, it turned out that many features measured by CloudCmp were not available on all four providers that they tested. This means the results are not as comprehensive as they aimed to be and it remains difficult to give a complete comparison of all providers.

Fortunately, even though the tests executed by CloudCmp may seem very synthetic, their case studies show that they are still fairly representative of the performance of more realistic applications.

While CloudCmp cannot be used to select for specific functions, it is useful to get a detailed insight into the performance and cost trade-offs. CloudCmp does this by computing the performance metrics for the compute cluster, persistent storage and network connections. Users may find CloudCmp useful to quickly determine which cloud services fit their superficial needs before delving into deeper comparisons with AHP or TOPSIS. On the other side, cloud providers can use CloudCmp to discover performance bottlenecks in their systems or find ways to improve their service and remain competitive.

TOPSIS was also introduced as another technique for cloud provider selection problems. As already mentioned before, TOPSIS performs analysis of available cloud services using fuzzy opinions. This technique makes use of the criterion's weights produced by AHP as an input parameter. TOPSIS is a popular technique because of its theoretical rigor, ability to represent the human rationale during selection, and prominence in solving traversal ranks [16].

As all three methods have their relative merits, there is no clear winner here. AHP and TOPSIS are both very comprehensive, but require a great deal of set-up and run-time on the cloud platforms. CloudCmp on the other hand is simple, but also only offers a shallow insight into the performance and does not take special functionality into account. It is therefore our advice to use a combination of the three techniques.

If the application has a special need for specific functionality, AHP and TOPSIS may be used to select the providers that offer this and then CloudCmp could be used to find the cloud provider with the optimal performance/cost ratio. On the other hand, if the application does not rely on special functionality, but instead needs high performance in processing, disk I/O or network, CloudCmp can be used to select one or more providers that offer the required performance. If the CloudCmp comparison delivers in many close results, AHP and TOPSIS can then be used to make a finer selection.

5 Future Work

Since AHP has some disadvantages such as the decision maker subjectivity can yield uncertainties when determining pairwise comparisons, we will incorporate AHP with another technique called fuzzy AHP in order to overcome this problem. We defer this for future work.

Acknowledgements

We would like to thank our expert reviewer Vasilios Andrikopoulos and our colleagues Timon Back and Ankita Dewan who gave us feedbacks in order to improve our paper.

References

[1] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker. Green cloud computing: Balancing energy in processing, storage, and transport. *Proceedings of the IEEE*, 99(1):149–167, Jan 2011.

[2] Bushan.R and Rhai.K. Strategic decision making applying the analytic hierarchy process, 2004.

[3] D. F. García and J. García. Tpc-w e-commerce benchmark evaluation. *Computer*, 36(2):42–48, Feb. 2003.

[4] R. L. Grossman. The case for cloud computing. *IT Professional*, 11(2):23–27, March 2009.

[5] P. Gupta, A. Seetharaman, and J. R. Raj. The usage and adoption of cloud computing by small and medium businesses. *International Journal of Information Management*, 33(5):861 – 874, 2013.

[6] A. Li, X. Yang, S. Kandula, and M. Zhang. Cloudcmp: Comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 1–14, New York, NY, USA, 2010. ACM.

[7] X. R. Mingrui.S, Tianyi.Z. Consumer-centered cloud services selection using ahp, 2013.

[8] G. Motta, N. Sfondrini, and D. Sacco. Cloud computing: An architectural and technological overview. In *Service Sciences (IJCSS), 2012 International Joint Conference on*, pages 23–27. IEEE, 2012.

[9] National Center for Biotechnology Information. BLAST: Basic local alignment search tool. `https://blast.ncbi.nlm.nih.gov/Blast.cgi`. Accessed: 2017-03-03.

[10] S. Ramgovind, M. M. Eloff, and E. Smith. The management of security in cloud computing. In *2010 Information Security for South Africa*, pages 1–7, Aug 2010.

[11] T. Saaty. The analytic hierarchy process, 1980.

[12] J. R. San Cristóbal. *Multi criteria analysis in the renewable energy industry*. Springer Science & Business Media, 2012.

[13] K. Shiv, K. Chow, Y. Wang, and D. Petrochenko. Specjvm2008 performance characterization. In *Proceedings of the 2009 SPEC Benchmark Workshop on Computer Performance Evaluation and Benchmarking*, pages 17–35, Berlin, Heidelberg, 2009. Springer-Verlag.

[14] V. Srikrishna.S, Sreenivasulu Reddy. A. A new car selection in the market using topsis technique, 2014.

[15] The Trustees of Princeton University. Planetlab. `http://planet-lab.org/`. Accessed: 2017-03-03.

[16] M. Whaiduzzaman, A. Gani, N. B. Anuar, M. Shiraz, M. N. Haque, and I. T. Haque. Cloud service selection using multicriteria decision analysis. *The Scientific World Journal*, 2014, 2014.

2D keypoint detection and description

Willem Dijkstra and Tonnie Boersma

Abstract— The image processing industry requires a way to detect and compare keypoints in images. SIFT is over a decade old and still one of the most precise and robust options. SURF is SIFT his potential successor and claims to have similar results with less computational effort. In recent years other 2D keypoint detection and description algorithms like BRISK, ORB and KAZE appeared which all claimed to be better then SIFT or SURF in some aspect. This paper compares SIFT, SURF, BRISK, ORB and KAZE, based on their respective algorithm and an experiment which compares the repeatability of the corresponding algorithms. This paper discusses common goals and hurdles each method has or faces and aims to provide an unbiased advice of when to use which method.

Index Terms—2D keypoint detection, 2D keypoint description, SIFT, SURF, ORB, BRISK, KAZE

1 Introduction

Many computer vision applications use 2D keypoint detection and description algorithms as initial step. These applications can be used for object recognition or motion tracking. 2D keypoint detection can be described as locating points of interest in images, while description refers to the representation of each keypoint.

In this paper we discuss and compare the following 2D keypoint detection and description algorithms: SIFT, SURF, ORB, KAZE and BRISK. Each of these algorithms varies widely in the types of features that are detected (e.g. edges, corners or blobs), the computation time required for detecting and describing the keypoints and the repeatability of the keypoints. Repeatability in this case refers to detecting similar keypoints in similar scenes regardless of distortion. Each comparison is based on the original papers with respect to the algorithms in combination with our own experiments. The focus of these experiments is the repeatability of the keypoints and the robustness of the detection, without taking into account the design goals of the algorithm itself. For example the repeatability of algorithms designed for repeatability is compared to an algorithm designed for low computational cost.

The sRD-SIFT [3] dataset is used for the experiments, this dataset contains two sub sets: planar scenes and object scenes. Each scene consists of multiple images grouped by radial distortion, this combination is ideal for our experiments.

In this paper we start with a background section containing the general idea of computer vision, the aims of feature detecting and description and a general introduction to the five 2D keypoint detection and description algorithms. In section 3 we discuss the sRD-SIFT dataset in combination with the used methods. Followed by section 4 in which we discuss the results of the experiments. In section 5 the five algorithms are compared using their respective papers in combination with our experiments, followed by the final section 6 containing the conclusion and suggestions for future research.

2 Background

This section consists of a short introduction to computer vision, followed by different aims of feature detection and description algorithms and a general explanation of each algorithm.

2.1 Computer vision

The simplistic idea of computer vision is the process of acquiring, processing and analyzing digital images. Based upon the resulting analyses, data can be extracted which can be used to, for example, make decisions.

Regardless of the goal of a computer vision application it requires a specific initial step. This step dictates how to locate objects of interest in images in order to analyze them. In general for this step one of the following three methods can be selected: 1) Writing a dedicated algorithm. 2) Using a feature detector in combination with a feature descriptor, or 3) Deep learning.

Writing a dedicated algorithm is the quickest solution in terms of computational time. If you want to locate strawberries in a field, you can use predefined information of the strawberries, whereas strawberries have a red color. By using this information an algorithm can detect objects relatively fast, but is in practice often limited by constraints, because strawberries are initially green.

Using a feature detector and descriptor is more robust but requires in general more computational time. This method detects similarities between the reference images (e.g. strawberries) and the provided image (e.g. field with strawberry plants). In general providing more reference images results in a more robust solution, due to more reference points for comparison. But the extra comparisons do increase the computational cost.

Finally deep learning, this method learns based on a ground-truth. Providing a large quantity of images in combination with the desired output a generic system is trained to classify new input. While in theory this method is in production the best solution in terms of both computational cost and robustness it requires a large amount of reference data and a computational intense training phase.

Dedicated algorithms are in general fast but are limited in usage, because you have to tailor the algorithm for the specific goal. Deep learning on the other hand requires a large amount of training data and a computational heavy training phase before you are able to work with it. Feature detectors and descriptors are in the middle ground. Feature detectors and descriptors are relatively fast and require no training phase, but they can still produce proper results. The next section will give more information about feature detectors and descriptors in general.

2.2 Feature detectors

Feature detectors are used to locate point of interest in images and its quality is based on the following points:

- Repeatability
  Running the feature detector multiple times on images of a same scene but from a different viewpoint should result in similar, if not identical keypoints.

- Distinctiveness
  Each keypoint should be based on distinctive features.

- *Willem Dijkstra is a MSc. Computing Science student at the University of Groningen, E-mail: w.dijkstra.16@student.rug.nl.*
- *Tonnie Boersma is a MSc. Computing Science student at the University of Groningen, E-mail: t.boersma.3@student.rug.nl.*

- Robustness
  The influence of noise and transformation (translation, rotation, scale, etc.) on the keypoints should be minimal.

- Computational time
  The detection process should take little computational time.

In figure 1 three examples of feature detectors are shown. In the left image of figure 1 features detected by SIFT are shown. The middle image shows features detected by SURF and the right image shows features detected by ORB. This shows the different features that can be detected by different feature detectors.

(a) Features detected by SIFT (b) Features detected by SURF (c) Features detected by ORB

Fig. 1: Feature detector examples using different algorithms

2.3 Feature description

Feature descriptors encode relevant information into a feature vector and are also referred to as some sort of numerical "*fingerprint*" which can be used to distinguish one feature from another. Feature descriptors are used to compute a unique description for each keypoint. This computation is performed by using a sampling grid. Figure 2 shows the sampling grid of SIFT, figure 3 shows the sampling grid of BRISK. The quality of a feature descriptor is based on the following points:

- Consistent description
  All different keypoints should have a unique description, while similar keypoints should have a similar description.

- Robustness
  The influence of noise and transformation (translation, rotation, scale, etc.) on the descriptions should be minimal.

- Vector size
  The size of the description should be as small as possible while minimizing the information loss. This is important since the description size is related to the time it takes to compare features.

- Type
  Multiple options are available, but binary vectors and feature vectors are the most common. Feature vectors can contain more information compared to a binary vector of the same size. However, the distance between binary vectors can be computed faster in comparison with the distance between feature vectors.

- Computational time
  The description process should take little computational time.

2.4 Feature detector and descriptor

In practice optimizing all previous mentioned points of a feature detector and descriptor may be difficult. That is why there are specialized detectors and descriptors for various goals. If you want to, for example, build a stitching or a real-time object tracking application each application has its requirements.

A stitching application has to be precise and requires detailed keypoint descriptions. However, if you want to stitch lunar images, they do not have to be computed in real time which allows a more in depth comparison. Also the provided data of the lunar images has little to no translation and rotation.

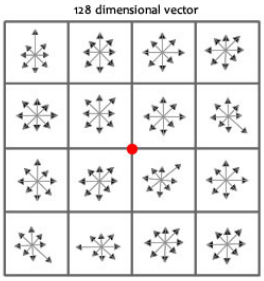


Fig. 2: Sampling grid of SIFT [2]

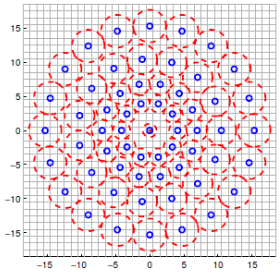


Fig. 3: Sampling grid of BRISK [7]

Real-time object tracking on the other hand requires a detector and descriptor with a small computational cost, it also has to be robust with respect to distortion and illumination. If not the object might be lost when rotated or if a different illumination is used.

It is possible to use a combination of a detector and descriptor of different methods, e.g. use the detector of SIFT and the descriptor of SURF. It is advised however to use a combination of the detector and descriptor of the same method since they complement each other. This is the reason that this paper focuses on the original detector and descriptor combinations.

In the following sections the considered feature detector and descriptor algorithms are discussed. In these sections an introduction to the considered algorithms is given. This includes for what purpose the algorithm is made and a short summary of what the algorithm does.

2.5 Distinctive Image Features from Scale-Invariant Keypoints (SIFT) [8]

The original state-of-the-art algorithm is SIFT, it is translation invariant and robust. Making it the best choice when robustness is key. The downside is its computation time, making it unfit for real-time applications. While being more than a decade old it is still today a worthy competitor.

2.5.1 Algorithm

First a scale space is created to ensure scale invariance. Then the Difference of Gaussian (*DoG*) is used to approximate Laplacian of Gaussian (*LoG*), in order to reduce the computation time. The minima and maxima in this approximations represent keypoints. Bad keypoints are filtered using an Harris Corner Detector variant making the following steps more robust. An orientation is calculated for each of the resulting keypoints. The description of the keypoints is relative to this orientation, making the description rotation invariant. The final step is generating 128 feature values representing each keypoint. These features are based on the gradient magnitudes and orientations found at the keypoint location weighted by a Gaussian function.

2.6 Speeded Up Robust Features (SURF) [5]

SURF, uses a similar approach as SIFT but uses either short-cuts or computational less expensive steps. This results in a similar results and reduced computation time, because of this it is more likely to be used in real-time applications.

2.6.1 Algorithm

Similar to SIFT the algorithm of SURF starts by creating a scale space and by approximating the Laplacian of Gaussian *LoG*. But instead of using Difference of Gaussian (*DoG*) box filters are used. These are less computational expensive and still able to give a proper approximation. The next step calculates the orientation of each keypoint using wavelet responses. These wavelet responses can be more easily calculated than gradients, because computing wavelet responses is computationally less expensive. The final step is generating 64 feature values

representing each keypoint. These features are based on the horizontal and vertical wavelet response.

2.7 Oriented FAST and Rotated BRIEF (ORB) [10]

ORB, as its title suggests is a combination of the FAST keypoint detector and the BRIEF keypoint descriptor. It is designed to be used in real-time applications and in an environment like cellphones which have less processing power. At the same time ORB has a similar performance compared to SIFT.

2.7.1 Algorithm

ORB uses FAST detector in combination with scale pyramids to generate scale invariant keypoints. Since FAST does not provide an orientation option it is extended with intensity centroids resulting in Oriented FAST (OFAST) For keypoint description ORB uses BRIEF, which has a small computational cost but does not take rotation into account. BRIEF is made orientation aware using the orientation calculated with OFAST, which is called Steered BRIEF. The downside of Steered BRIEF is a significant drop in distinctiveness with respect to each description. This is negated using a greedy similarity search through a predefined training set with 300k keypoints, which results in features with the most variance. This method is called rBRIEF. The combination of OFAST and rBRIEF is called ORB, which results in a 256 binary feature vector.

2.8 KAZE Features (KAZE) [4]

The aim of KAZE is to take a step forward in performance both detection and description against previous state-of-the-art methods, while having a similar computational time. The successor of KAZE is called AKAZE which produces similar results, but with a lower computational cost.

2.8.1 Algorithm

The base of KAZE is its usage of Non-linear Scale Space. Gaussian blurring used in SIFT and SURF smooths equal for all the structures in the image, whereas in the non-linear scale space strong image edges remain unaffected. The detection itself is similar to SIFT implementation adapted to the different scale space. The orientation is calculated using the same method as SURF. For the description itself M-SURF is used which results in a 64 feature vector.

2.9 BRISK [7]

BRISK aims at reducing the computational cost while having similar results compared to SIFT. The major difference between BRISK and other mentioned algorithm is its descriptor. Instead of using a regular distributed grid (see figure 2) in combination with gradients, a circular grid(see figure 3) is used in combination with intensities. It is computational less expensive to calculate intensities in comparison with gradients, and the circular grid results in a shorter feature vector making it easier to compare. This again makes it more suitable for real-time applications.

2.9.1 Algorithm

First of all points of interest are identified across both the image and scale dimensions using a saliency criterion. The location and the scale of these keypoints are obtained in the continuous domain via quadratic function fitting. Finally the oriented BRISK sampling pattern is used to obtain pairwise brightness comparison results which are assembled into the binary BRISK descriptor.

3 Materials and Methods

In this section the content of the sRD-SIFT dataset is explained, as well as the experimental methods used to compare the 2D keypoint detection and description algorithms.

3.1 Dataset

To compare the repeatability of the 2D keypoint detection and description algorithms, the sRD-SIFT [3] dataset is used. This dataset consists of two subsets, one with images of planar scenes and one with images of various objects. Each of these subsets consists of three different levels of radial distortion (10%, 25% and 45%). This radial distortion is a real lens distortion due to the fact that different types of lenses are used to acquire the images.

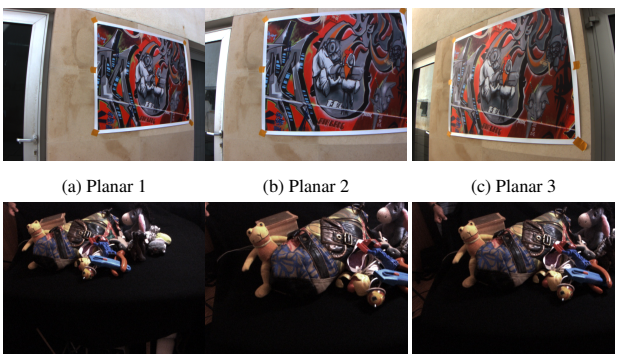
An overview of the subset with images of planar scenes is shown in table 1. This subset also includes projective transformation between planes which is also known as the homography. These are used as ground-truth during experiments since they represent the exact translation between images An overview of the subset with images of objects is shown in Table 2. These tables shows the amount of radial distortion as a percentage and the number of images taken. In figure 4 six images are shown providing a general impression of the sRD-SIFT dataset.

Planar scene	Radial distortion (%)	Number of images
1	10	13
2	25	13
3	45	13

Table 1: sRD-SIFT planar scenes subset

Object scene	Radial Distortion (%)	Number of images
1	10	7
2	25	7
3	45	7

Table 2: sRD-SIFT objects scenes subset



(a) Planar 1 (b) Planar 2 (c) Planar 3

(d) Object 1 (e) Object 2 (f) Object 3

Fig. 4: 6 images from sRD-SIFT dataset

3.2 Experiment framework

For performing the experiments a framework is developed in c++. This framework uses the OpenCV library [6] in combination with the *contrib* module. OpenCV with the *contrib* module contains the implementations for 2D keypoint detection and description algorithms such as SIFT, SURF, ORB, KAZE and BRISK, for that reason OpenCV is implemented in the framework to perform the experiments.

The plots shown in this paper are generated using Matlab with the data provided by the c++ framework.

3.3 Experiments

The first experiment is aimed at evaluating the repeatability of the considered keypoint descriptors. In this experiment we use the complete dataset of the planar scenes and the complete dataset of the object scenes. The second experiment is aimed at evaluating the robustness

of the considered keypoint descriptors, and uses the complete subset containing planar images.

3.3.1 Repeatability

This experiment estimates the repeatability with respect to each algorithm and scene type.

The following process is repeated for each individual algorithm. Each image in the dataset is processed using the keypoint detector and descriptor, Followed by matching each combination using the OpenCV *Brute Force Matcher*. These matches represent the distance between keypoints. Whereas similar keypoints will have a minimal distance and uncorrelated keypoints have a maximum distance.

Correct matches are determined using the method proposed in the paper about SIFT [8]. In short the distance between the best and second best keypoint is used as reference in combination with a user defined ratio. Keypoints with a distance larger than this reference are marked as bad match and are excluded.

The repeatability itself is calculated using the correct matches in combination with the total amount of matches as described in [9]. The total amount of correct matches is divided by the minimum amount of the original matches with respect to the two images. Using these ratios the average repeatability with respect to algorithm and dataset is calculated.

3.3.2 Robustness

This experiment estimates the robustness with respect to each algorithm and scene type.

Similar as the previous experiment each image is processed using the individual algorithms. Using the resulting keypoints an homography is calculated, which contains the exact translation between two images. The planar dataset contains the original homography, comparing these the average deviation is calculated.

4 Results

This section describes the results of the performed experiments. The results of the repeatability experiments are shown in figures 5, 6 and 7. For each type of radial distortion (10%, 25% and 40%) a boxplot is generated. Each boxplot includes the planar scene and the object scene with respect to the radial distortion and algorithm.

Table **??** and table **??** show for each algorithm and each type of radial distortion the average of matching keypoints in terms of percentage.

In general it can be seen that regardless of the algorithm the planar scene has a higher score compared to the object scene. However when the radial distortion is increased this difference is still present but less prominent. In terms of general repeatability the SIFT and KAZE have the highest score while ORB is general has the lowest score. And radial distortion as expected has a negative influence on all algorithms, decreasing on average the repeatability by 9.2%

Due to time constraints the second experiment could not be completed and is excluded during the discussion. This experiment could provide valuable information, because of that it is included in the future work section.

Method	10% RD	25% RD	40% RD
SIFT	50.5	38.8	27.5
SURF	36.8	25.0	19.1
ORB	17.4	12.7	07.5
KAZE	46.2	32.0	21.7
BRISK	22.5	15.2	11.0

Table 3: Average matching keypoints in terms of percentage for the planar scenes, for each algorithm and each type of radial distortion (RD)

Method	10% RD	25% RD	40% RD
SIFT	26.1	31.9	22.2
SURF	24.7	28.1	20.1
ORB	05.3	11.6	08.5
KAZE	24.0	31.1	24.8
BRISK	08.1	15.6	07.7

Table 4: Average matching keypoints in terms of percentage for the object scenes, for each algorithm and each type of radial distortion (RD)

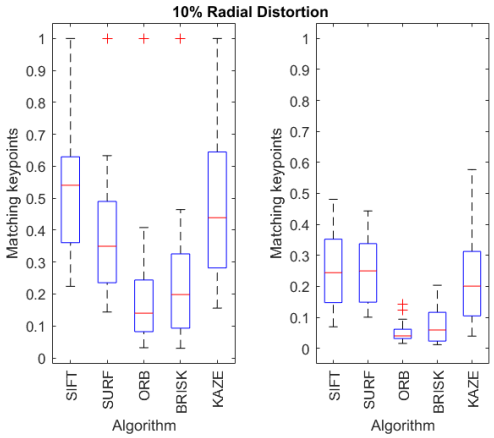


Fig. 5: Compare matching keypoints between 2D keypoint with 10% radial distortion. Left figure with planar scenes. Right figure with object scenes

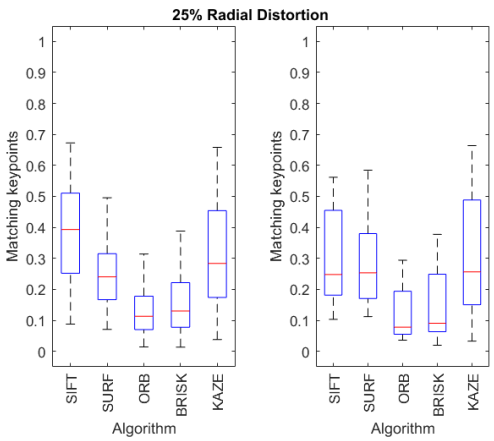


Fig. 6: Compare matching keypoints between 2D keypoint with 25% radial distortion. Left figure with planar scenes. Right figure with object scenes

5 Discussion

There are no experiments performed with regard to the computational cost of each algorithm. The comparison of the computational cost is therefore solely based upon the papers of each considered algorithm.

ORB is designed to minimize the computational cost and in general it has. Using a simple rotation invariant algorithm and making it rotation aware is the technique used in ORB. Due to its simplicity and optimization it is by far the quickest solution with regard to the considered algorithms.

The design goal of BRISK is similar to ORB. BRISK claims to provide similar results compared to SIFT and SURF, but BRISK comes with less computational cost. In BRISK the largest gain factor is the

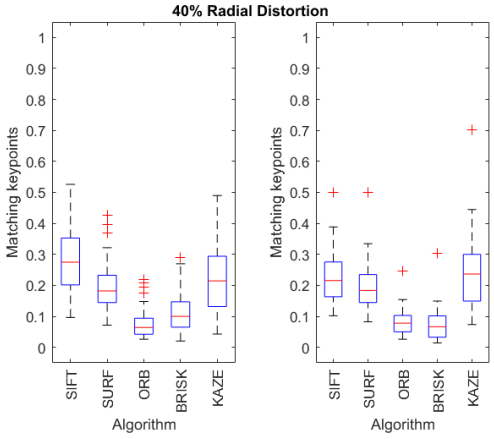


Fig. 7: Compare matching keypoints between 2D keypoint with 40% radial distortion. Left figure with planar scenes. Right figure with object scenes

binary descriptor which can be calculated and compared very efficiently resulting in the second spot.

SURF, as proven before, has a reduced computational cost compared to SIFT. Unfortunately SURF does not outperform ORB and KAZE.

KAZE and SIFT are more or less the same in terms of computational cost. Both algorithms are not designed for a low computational cost which is definitely noticeable when comparing these two algorithms with SURF, ORB and BRISK.

In terms of repeatability with little distortion the order solely based on the papers can be roughly estimated. The experiment done in this paper is to reinforce this estimation.

SIFT is the original state-of-the-art solution and has earned that spot for a reason. While almost all alternatives focus on a lower computational time with similar results, only KAZE has the focus to be an improvement.

Comparing the techniques of these two algorithms one thing becomes clear, both are based on a similar design. The major difference is the used scale space. SIFT uses a linear scale space while KAZE uses a non-linear scale space. This allows KAZE to apply Gaussian filtering locally resulting in a the removal of small detail while respecting the contours of large objects, which global Gaussian filtering is not able to do. This results in more exact keypoint locations allowing KAZE to make a more precise comparison compared to SIFT.

Intuitively this leads to the conclusion that KAZE is better than SIFT. Using the results of our experiments (see figure 5 and table **??** & **??**) this is unconfirmed. Although the object scenes differ only a little the planar scenes show an improvement. Due to this, SIFT wins with regard to the repeatability of the keypoints, whereas KAZE is a close second.

For the other algorithms it is difficult to determine the repeatability based on their respective papers. Since SURF, ORB and BRISK claim to have similar results compared to SIFT in specific situations or certain scenes. Using the differences from our own experiment as base line we conclude that SURF better than BRISK while BRISK is better than ORB.

In terms of repeatability with increasing distortion (see figure 6 & 7, table **??** & **??**) the order is difficult to estimate. Since most papers discuss the quality up to a certain point, but rarely include these types of outliers and none of the algorithms are build in a way to specifically deal with this. Still SIFT and KAZE are expected to remain as a solid choice, while ORB cannot decrease much further.

Again using the results of our own experiments it can be seen that all the repeatability ratios drop noticeably. Except for ORB which stay on average (both planar and object combined) and does not change much. Because of this the percentage difference between for example SIFT and KAZE becomes negligible. Resulting in SIFT and KAZE as still the best algorithms, followed by SURF, ORB and BRISK.

ORB with respect to the repeatability drop could be the winner, due to the fact that its repeatability, even though not great, remains almost constant on average.

In general a trend can be determined, whereas algorithms with a low computational cost pay for this in terms of repeatability. Hinting that cutting to many corners in the process is not a feasible strategy in the long run. Also different approaches are required to further improve both aspects without giving up on either one of them.

It is also clear that objects which can obscure each other differently due to translation are more difficult to track. This is simply because some keypoints can not be detected from certain angles. This is of course not something that can be fixed with an algorithm, but explains the difference in repeatability based on scene type.

6 Conclusion

SIFT which is the oldest algorithm is still the better algorithm in terms of design. SIFT is designed to be precise and robust and still today it tops most other feature detectors and descriptors. KAZE is a more recent algorithm which rivals SIFT in terms of robustness. KAZE even surpasses SIFT in certain situations, making it a better choice depending on the exact application. If robustness is the key target of the computer vision algorithm using one of these 2D keypoint detection and description algorithms is advised.

However, when a small computational cost is required (e.g. for smart phones, real-time application) ORB is the best contestant. Although ORB lacks some robustness, it is still able to yield decent results.

SURF and BRISK are in the middle ground, yielding both decent results for a reasonable computational cost.

Important to note is that both SIFT and SURF are patent protected. This means that for non educational purposes both SIFT and SURF require permission to be used. While this does not influence the performance of the algorithms it should still be taken into account when deciding what algorithm to choose.

To get a better comparison of the algorithms more experiments could be performed and more algorithms could be included in the research. The next section gives a few suggestions for future research.

6.1 Future work

Future work can include the following options:

The presented research only includes a comparison of five algorithms. More algorithms like for example AKAZE or FREAK can be taken into consideration, since a wider range of algorithms can result in an improved contrast between algorithms and highlight exceptional results.

In this research experiments which evaluate the computational cost of the algorithms have been omitted. Instead in this paper computational cost is based on the original papers and not on experiments. Experiments which evaluate and compare the computational costs of each algorithm could give a better comparison.

Due to lack of time the experiment which would evaluate the robustness of the algorithms is not performed. Performing these experiments could give a valuable insight in the actual robustness of each algorithm.

Also larger and/or different datasets to generate more exact results could be used.

Acknowledgements

We would like to thank the expert reviewer Nicola Strisciuglio and the anonymous reviewers who reviewed drafts of this paper

REFERENCES

[1] Key point example. https://nl.mathworks.com/help/vision/feature-detection-and-extraction.html.

[2] Sift descriptor. http://aishack.in/tutorials/sift-scale-invariant-feature-transform-features.

[3] srd-sift data sets. http://arthronav.isr.uc.pt/ mlourenco/srdsift/dataset.html.

[4] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. Kaze features. In *European Conference on Computer Vision*, pages 214–227. Springer, 2012.

[5] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[6] G. Bradski. Opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.

[7] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[9] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72, 2005.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

Segmentation of blood vessels in retinal fundus images

Michiel Straat and Jorrit Oosterhof

Abstract—In recent years, several automatic segmentation methods have been proposed for blood vessels in retinal fundus images, ranging from using cheap and fast trainable filters [1] to complicated neural networks and even deep learning [2] [3] [4].

One example of a filted-based segmentation method is B-COSFIRE [1]. In this approach the image filter is trained with example prototype patterns, to which the filter becomes selective by finding points in a Difference of Gaussian response on circles around the center with large intensity variation.

In this paper we discuss and evaluate several of these vessel segmentation methods. We take a closer look at B-COSFIRE and study the performance of B-COSFIRE on the recently published IOSTAR dataset [5] by experiments and we examine how the parameter values affect the performance. In the experiment we manage to reach a segmentation accuracy of 0.9419.

Based on our findings we discuss when B-COSFIRE is the preferred method to use and in which circumstances it could be beneficial to use a more (computationally) complex segmentation method. We also shortly discuss areas beyond blood vessel segmentation where these methods can be used to segment elongated structures, such as rivers in satellite images or nerves of a leaf.

Index Terms—Blood vessel segmentation, Image processing, B-COSFIRE, Retinal image analysis, Fundus imaging, Medical image analysis, Retinal blood vessels, Segmentation, Fundus, Retina, Vessel segmentation.

1 Introduction

The inspection of the blood vessel tree in the fundus, which is the interior surface of the eye opposite to the lens, is important in the determination of various cardiovascular diseases. This can be done manually by ophthalmoscopy, which is an effective method of analysing the retina. However, it has been suggested that using fundus photographs is more reliable than ophthalmoscopy [1]. Additionally, these images can be used for automatic identification of the blood vessels, which can be a difficult task due to obstacles such as low contrast with the background, narrow blood vessels and various blood vessel anomalies. A segmentation method with high accuracy can serve as a significant aid in diagnosing cardiovascular diseases, as it highlights the blood vessel tree in the fundus.

In recent years, several segmentation methods have been proposed for the automatic segmentation of blood vessels, ranging from using cheap and fast trainable filters [1] to complicated neural networks and even deep learning [2] [3] [4].

One example of a filted-based segmentation method is Bar-Combination Of Shifted Filter Responses, in short B-COSFIRE [1]. In this approach we train the image filter with example prototype patterns, to which the filter becomes selective by finding points in a Difference of Gaussian response on circles around the center with large intensity variation. Azzopardi *et al.* used two prototype patterns for the segmentation: One for bars and one for bar-endings. The filter achieves rotation invariance by rotating the points of interest for the prototype patterns, yielding a filter that is selective for these vessel orientations as well.

In section 2 we discuss the theory behind the filter-based method B-COSFIRE in more detail. We then discuss supervised vessel segmentation methods that are based on machine learning in section 3. In section 4 we describe our set-up for experiments with B-COSFIRE on a recent dataset that contains retinal images acquired with a camera based on Scanning Laser Ophtalmoscopy (SLO) technology and we discuss the results in section 5. Based on the study and the results of the experiments we then discuss the advantages- and disadvantages of the discussed methods.

- *Michiel Straat is a first year Computing Science master student at the University of Groningen.*
- *Jorrit Oosterhof is a first year Computing Science master student at the University of Groningen.*

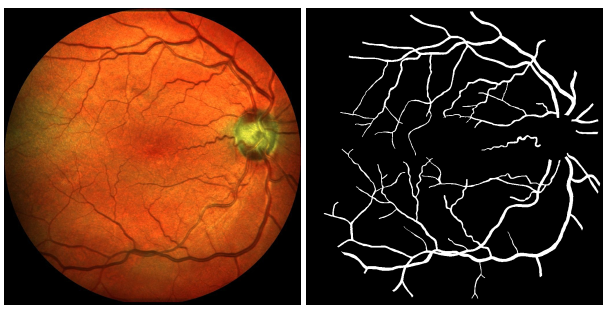
2 B-COSFIRE

In this section we discuss B-COSFIRE in more detail. The method is based on the **C**ombination **O**f **S**hifted **FI**lter **RE**sponses, which is a trainable filter used for interest point detection, as described in [6]. The filter is trained in a training stage in which it is configured to be selective to specific prototype patterns.

The B-COSFIRE method is a specific case of COSFIRE, in which the B stands for "bar". The filter is trained using vessel-like prototype patterns, which allows it to be selective for such structures. Azzopardi *et al.* proposed two B-COSFIRE filters: The *symmetric* filter, which is suitable for bars, and the *asymmetric* filter, suitable for bar endings. In the following, we briefly discuss the method as outlined in [1].

The B-COSFIRE filtering method proceeds in a number of stages, which are roughly divided in training each filter with a prototype pattern in order for the filter to become selective for its prototype pattern, pre-processing the retinal input image in order to enhance contrast of the vessels and the actual filtering, which yields the final output response. The response is thresholded to obtain a binary image, in which for each pixel in the response a gray value above the threshold is shown as a white pixel (1) indicating "vessel", and a gray value below the threshold is shown as black indicating "non-vessel". This yields the typical vessel tree image shown in Figure 1.

In the next sections we discuss the stages mentioned above in more detail.



(a) Retinal fundus image (b) Corresponding vessel segmentation.

Fig. 1: A typical vessel tree extracted from a retinal image, in which each pixel is labeled either "vessel" (white) or "non-vessel" (black).

2.1 Training a filter using a prototype pattern

Difference of Gaussian (DoG) filters are used to detect changes in intensity.

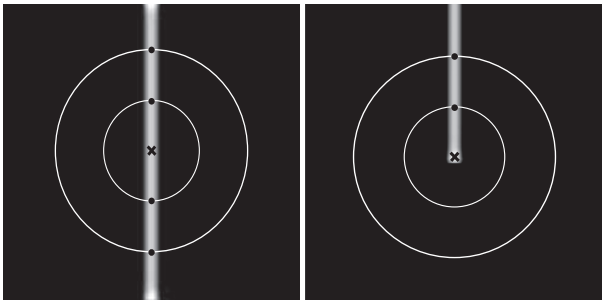
$$DoG_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) - \frac{1}{2\pi(0.5\sigma)^2}\exp\left(-\frac{x^2+y^2}{2(0.5\sigma)^2}\right). \quad (1)$$

The response by applying a DoG filter as in Equation 1 to an image f is defined as the convolution of f with the filter kernel in Equation 1:

$$c_\sigma(x,y) = |f \star DoG_\sigma|^+ . \quad (2)$$

The selectivity of a B-COSFIRE filter is automatically configured by presenting the filter with a prototype pattern. Then, a number of concentric circles are put around the filter's center of support and the points on these circles with a high intensity variation are determined, called the points of interests. One such point is described by a triple (σ,ρ,ϕ), where σ is the standard deviation of the DoG filter that contributed this point, ρ is the radius of the circle that this point lies on and ϕ is the angle with respect to the center of support. The training of one prototype pattern yields a set of triples $S = \{(\sigma_i,\rho_i,\phi_i)\, i = 1,...,n\}$ describing n points of interest.

In Figure 2a we see an example of how selectivity is obtained for a vertical bar pattern. The center point is surrounded by two circles. On each circle two local maxima in the DoG response are found, where two are above the center point and the other two below the center point. For this reason we call this the *symmetric filter*. In Figure 2b an example is shown of the configuration of a bar-ending. In this case two points above the center point are marked as local maxima in the DoG response. This filter we call *asymmetric*.



(a) Symmetric filter (b) Asymmetric filter

Fig. 2: (a): A DoG filter applied to a straight full vertical bar pattern. After applying the DoG filter four local maxima are detected on the two concentric circles surrounding the center point, i.e. $|S| = 5$ (including centerpoint). These correspond to the positions of the highest intensity variations around the center point. (b): A DoG filter applied to a straight half vertical bar pattern. Two local maxima are found on the circles, i.e. $|S| = 3$ (including centerpoint). Images taken from [1].

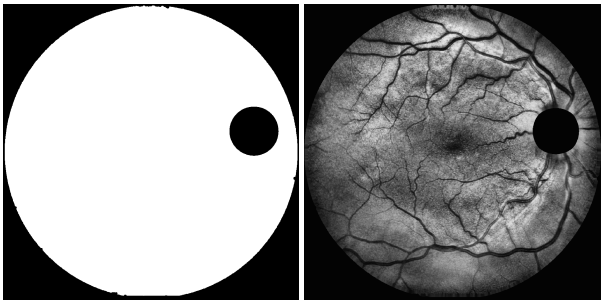
As the selectivity of the filter directly depends on the specified prototype patterns, by training with the two prototype patterns in Figure 2 the filter is only selective for vertical bars and bar-endings. To achieve selectivity also for other vessel orientations, we could train bar prototype patterns with more orientations. A more easy and efficient way to achieve rotation invariance is to transform the set of points S that were found for the vertical bar patterns to a new set:

$$R_\psi(S) = \{(\sigma_i,\rho_i,\phi_i+\psi) | \forall(\sigma_i,\rho_i,\phi_i) \in S\}, \quad (3)$$

where ψ is the angle of the orientation of the bar for which the set $R_\psi(S)$ is selective. In the symmetric case, by taking $\psi = \frac{\pi}{12}, \frac{\pi}{6}, ..., \frac{11\pi}{12}$, we obtain a filter that, including the original vertical bar orientation, becomes selective for 12 bar orientations. In the asymmetric case we must naturally consider 24 orientations, as in this case a rotation by π gives rise to a different orientation due to the asymmetry.

2.2 Pre-processing the retinal image

We follow the pre-processing steps as discussed by Azzopardi *et al*. Previous works have shown that the green channel of RGB images defines the contrast between vessels and the background better than the red channel, which has low contrast, or the blue channel, which shows a small dynamic range [7][8][9][10][11]. Therefore, we only use the green channel for the segmentation. The Field Of View (FOV) masks are provided with the dataset, and we smooth the borders around the FOV to ensure there will be no false positives in these areas because of the high contrast. The dataset IOSTAR also comes with masks that indicate the Optic Disc (OD) in the retina. We use these masks instead of the normal FOV masks to ensure that segmentation will not take place inside the OD, since the ground truth images do not have a specified segmentation inside the OD area as well, which gives rise to a better comparison. In the last step the contrast-limited adaptive histogram equalization is applied. The output of the pre-processing stage for the retinal image of Figure 1a is given in Figure 3b.



(a) Mask (b) Green channel

Fig. 3: (a): The mask that is used to segment the retina including an indication of the optic disc. (b): The resulting green channel of a retinal image after pre-processing is applied as described in subsection 2.2.

2.3 Filter application

We have shown how a B-COSFIRE filter can achieve selectivity for specific bar patterns, by selecting points around center of support of the filter with a high intensity variation. We have also shown how from there, rotation-invariance can be easily achieved by rotating the points of interest by angles ψ. We have shown how this works for two prototype patterns that give rise to a symmetric- and an asymmetric bar filter. In order to achieve a more tolerant filter and also a ranking of the points of interest according to their importance such that the points more closely to the center point get a higher weight, the DoG responses are blurred with a Gaussian $G_{\sigma'}(x',y')$. Because closer points need to get more weight, the standard deviation of the Gaussian is defined in terms of ρ_i, the radius of the circle on which the point of interest lies:

$$\sigma' = \sigma'_0 + \alpha\rho_i , \quad (4)$$

where σ_0 is a constant for the base standard deviation and α is the rate at which the standard deviation increases. Therefore, the higher α, the more tolerant the filter becomes. The DoG responses are moved to the center of the filter support by shifting with $(\Delta x_i = -\rho_i\cos\phi_i, \Delta y_i = -\rho_i\sin\phi_i)$. The ith blurred and shifted DoG response then becomes:

$$S_{\sigma_i,\rho_i,\phi_i}(x,y) = \max_{x',y'}\{c_{\sigma_i}(x-\Delta x_i-x', y-\Delta y_i-y')G_{\sigma'}(x',y')\} , \quad (5)$$

where we consider a neighborhood around the points of interest of $-3\sigma' \leq x', y' \leq 3\sigma'$. As we can see, the maximum weighted response is the value we assign to $s_{\sigma_i,\rho_i,\phi_i}(x,y)$. Now at each pixel of the input image, each trained B-COSFIRE filter is applied by multiplying the now obtained blurred and shifted DoG responses in the set S in the following way:

$$r(x,y) = \left(\prod_{i=1}^{|S|} (S_{\sigma_i,\rho_i,\phi_i}(x,y))^{\omega_i} \right)^{1/\sum_{i=1}^{|S|}\omega_i}$$

$$\omega_i = \exp^{-\frac{\rho^2}{2\hat{\sigma}^2}},\ \hat{\sigma} = \frac{1}{3}\max_{i\in\{1\ldots|S|\}}\{\rho_i\}, \quad (6)$$

where we obtain $r_s(x,y)$ for the symmetric filter response and $r_a(x,y)$ for the asymmetric filter response. As a small sidenote, Equation 6 shows exactly why the filter is called COSFIRE: The response is a *combination*, which refers to the product, of *shifted* filter responses, which refers to the shifts of the filter responses by $(\Delta x_i, \Delta y_i)$ to the center of the B-COSFIRE filter support.

The sum of the symmetric- and the asymmetric filter $r_{as}(x,y) = r_s(x,y) + r_a(x,y)$ is then the total response of the filtering process. To get a binary image in which each pixel is classified as either vessel or non-vessel, we threshold the filter response $r_{as}(x,y)$ with threshold T, where pixel values of $r_{as}(x,y)$ above T are considered as vessel pixels, i.e.:

$$g(x,y) = \begin{cases} 1, & \text{if } r_{as}(x,y) > T \\ 0, & \text{if } r_{as}(x,y) \leq T \end{cases} \quad (7)$$

3 Alternative methods

The algorithms for segmenting blood vessels in retinal fundus images can be divided in supervised and unsupervised methods. B-COSFIRE is an example of an unsupervised approach to vessel segmentation. Supervised methods first train on a set of example images with their ground truth segmentation provided. [12] discusses several blood vessel detection methods that were published by 2012. In this section, we discuss a few of these methods and compare them with each other. Table 1, Table 2 and Table 3 show some performance metrics of the methods we discuss.

3.1 Deep Neural Networks

One can use deep neural networks to extract blood vessels from a fundus image. Deep neural networks are neural networks with many (hidden) layers. Liskowski *et al.* [3] use deep learning to identify blood vessels in fundus images. The deep learning algorithm is the error back-propagation algorithm extended with dropout [13]. The error back-propagation algorithm is a common algorithm in the field of neural networks. With dropout, during training a percentage of units is temporarily disabled, introducing an extra challenge for the training process.

Liskowski *et al.* use the DRIVE [7][14], STARE [15][16] and CHASE [17] datasets to verify their deep learning method. To classify pixels as being either *vessel* or *non-vessel*, a patch of $m \times m$ pixels is used, centred at the pixel. The neural network is fed with a triple of such patches. Each of those patches is at the same location, but each patch is from one of the three RGB channels. As a result, this method uses all three RGB channels to determine whether a pixel belongs to a blood vessel or not. This is in contrast to the filter-based method B-COSFIRE, which only uses the green channel as if it were a grey scale image [1], for reasons discussed in subsection 2.2.

For their experiments, Liskowski *et al.* use multiple configurations. Table 1 and Table 2 show metrics for the two configurations they found best. Furthermore, Liskowski *et al.* used structured prediction, an approach which also uses the neighbourhood of a pixel for the classification process. Separate results for the two best configurations using structured prediction are also shown in Table 1 and Table 2. Please refer to [3] for a detailed description of the configurations with and without structured prediction.

3.2 Ensemble classification-based approach

Fraz *et al.* [2] use an ensemble classifier of boosted and bagged decision trees. In short, they use multiple classifiers to classify the pixels and take a majority vote to determine whether a pixel is a *vessel* or *non-vessel*. Similar to the B-COSFIRE method, this method uses the green channel of the RGB image for the blood vessel segmentation. For experimenting, Fraz *et al.* use the DRIVE, STARE and CHASE datasets. The classifier is trained using a randomly selected subset of pixels for each image.

Method	AUC	Acc	Se	Sp
B-COSFIRE	0.9614	0.9442	0.7655	0.9704
Liskowski *et al.* [3]				
Balanced	0.9738	0.9230	0.9160	0.9241
No-Pool	0.9720	0.9495	0.7763	0.9768
Balanced SP	0.9788	0.9530	0.8149	0.9749
No-Pool SP	0.9790	0.9535	0.7811	0.9807
Fraz *et al.* [2]	0.9747	0.9480	0.7406	0.9807
Orlando *et al.* [4]	-	-	0.7897	0.9684

Table 1: Results on the DRIVE dataset

Method	AUC	Acc	Se	Sp
B-COSFIRE	0.9563	0.9497	0.7716	0.9701
Liskowski *et al.* [3]				
Balanced	0.9820	0.9309	0.9307	0.9304
No-Pool	0.9785	0.9566	0.7867	0.9754
Balanced SP	0.9928	0.9700	0.9075	0.9771
No-Pool SP	0.9928	0.9729	0.8554	0.9862
Fraz *et al.* [2]	0.9768	0.9534	0.7548	0.9763
Orlando *et al.* [4]	-	-	0.7680	0.9738

Table 2: Results on the STARE dataset

3.3 Fully Connected Conditional Random Field Model

Orlando *et al.* [4] use conditional random fields (CRFs). In contrast to normal classifiers, CRFs do not classify 'objects' purely based on the object. Note that word 'object' here can mean anything, like individual pixels or what activity belongs to an image. For example, for classifying that an image, from a sequence of snapshots of someone's life, displays an activity, it can be useful to know which activity was classified before.

CRFs work with graphs. Orlando *et al.* use a fully connected CRF (FC-CRF) where each pixel, being a node in the graph of the CRF, is linked to every other pixel. This has the advantage that the method can take long-range interactions between pixels into account, instead of only neighbouring information, which results in an improvement of the segmentation accuracy of the method [4]. Similar to the B-COSFIRE method, this method only uses the green channel of the RGB image for the blood vessel segmentation.

4 Experiments

To study the performance of the filter-based B-COSFIRE method discussed in section 2, we experiment with the method on a recent retinal image dataset. In this section we report on the parameters that we use and how we evaluate the resulting segmentation.

Method	AUC	Acc	Se	Sp
B-COSFIRE	0.9487	0.9387	0.7585	0.9587
Liskowski *et al.* [3]				
Using DRIVE for training				
No-Pool	0.9646	0.9473	0.7158	0.9810
No-Pool SP	0.9710	0.9515	0.7520	0.9806
Using STARE for training				
No-Pool	0.9543	0.9525	0.7091	0.9791
No-Pool SP	0.9880	0.9696	0.8145	0.9866
Fraz *et al.* [2]	0.9712	0.9469	0.7224	0.9711
Orlando *et al.* [4]	-	-	0.7277	0.9712

Table 3: Results on the CHASE dataset

4.1 Training parameters

A major advantage of the B-COSFIRE method is that the parameters are quite intuitive, so one can estimate a good set of parameter values for $(\sigma, \rho, \sigma_0, \alpha)$. In any case, the parameters must be carefully chosen, as the images of different retinal image dataset have varying properties. The parameters that are to be determined are:

- σ: The standard deviation of the outer Gaussian function in the DoG filter. Intuitively, the higher the detail in the image, i.e., the higher the resolution, the greater σ needs to be to still detect blood vessels with high accuracy, since they are made up of more pixels in high resolution images.
- ρ: The set of radii of the concentric circles surrounding the filter's center of support.
- σ_0: The base standard deviation of the Gaussian weighting function used for blurring the DoG responses, to allow for some tolerance in the position of the found points (See Equation 4).
- α: Parameter α that determines the standard deviation of the Gaussian used for blurring the DoG responses. The higher α, the more the standard deviation grows for larger concentric circles (See Equation 4).

To determine optimal $(\sigma, \rho, \sigma_0, \alpha)$, we split the datasets into a training set and a validation set. We use the training set for finding a good set of $(\sigma, \rho, \sigma_0, \alpha)$ and validate the decision on the entire dataset. We limit the search space of $(\sigma, \rho, \sigma_0, \alpha)$ by results found on specific datasets in previous research, such as the results of Azzopardi *et al*., in which for instance optimal values of σ were found for certain resolutions, and this hints us to the region in which the optimal value of σ is likely to reside. The procedure for determining the parameter values can then be summarized as follows:

1. Split the dataset into an equal sized training set and a validation set, each consisting of n images.
2. For a search space of $(\sigma_s, \rho_s, \sigma_{0s}, \alpha_s)$ which depends on the properties of the dataset, take the filter corresponding to this combination of parameters and filter all n training images with this filter obtaining n response images.
3. Iterate over a range of threshold $t \in [0,1]$ with intervals of 0.01. For each threshold t, segment all n response images according to Eq. (7). This yields n binary images with the vessel segmentation.
4. Having obtained these n binary images, we can compare all n images with the ground truth vessel segmentation images. Each vessel segmentation for one image f obtained from the filter and threshold combination paired with the ground truth segmentation of image f yields a confusion matrix, as observable in Table 4. We then compute for each of these n confusion matrices the Matthews Correlation Coefficient (MCC), which is directly computable from the corresponding confusion matrix as only quantities from the confusion matrix are used:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (8)$$

The MCC is particularly suitable for the assessment of binary classifiers, and can even be used if there is a disbalance between the number of datapoints, in this case pixels, per class. In our use case, the MCC is therefore suitable as vessel segmentation is a binary classification problem with a disbalance between classes, as there are naturally more non-vessel pixels compared to vessel pixels. Having computed the *MCC* for the responses of all training images thresholded with the current threshold, we average this set of *MCC* values. After that we consider the next threshold by going to step 4 and compute the new average *MCC*. In the end, the combination of parameters $(\sigma, \rho, \sigma_0, \alpha)$ along with a threshold t that yields the best average *MCC* on the training set are the parameters that we will use for segmentation of blood vessels in the entire dataset.

	Classifier: Vessel	Classifier: Non-Vessel
GT: Vessel	TP	FN
GT: Non-Vessel	FP	TN

Table 4: Confusion matrix: The classifier output (the thresholded filter response) is on the horizontal access. On the vertical access is the Ground Truth, which is the assessment of an expert that manually annotates all the pixels in the image as either vessel or non-vessel.

	σ	ρ_{max}	σ_0	α
Symmetric	4.8	20	3	0.3
Asymmetric	4.4	36	1	0.1

Table 5: The parameters used for the symmetric- and the asymmetric filter used for segmentation of blood vessels in the images of the IOSTAR dataset.

For the best set of filter parameters obtained from the above procedure, we store for each threshold t the corresponding average TPR and FPR over the segmented training images. By plotting the obtained values of TRP against FPR, we get a so-called ROC curve. The threshold t that gave the best average *MCC* value over the segmented training images is the threshold we use for the segmentation of the images in the validation set, and possible future images that are obtained with the same parameters as for the dataset under consideration.

4.2 Datasets

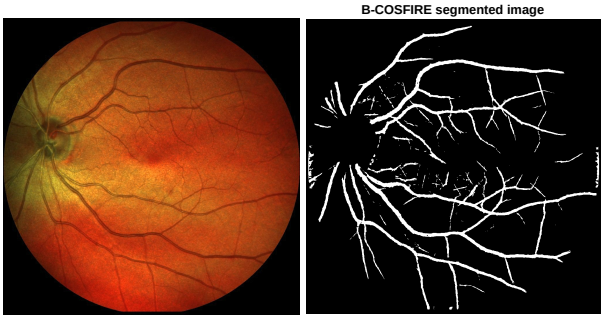
The retinal images in the IOSTAR dataset were obtained using Scanning Laser Ophtalmoscopy (SLO) [5], [18]. In 2015, the dataset was one of the first publicly available retinal datasets of which the data was obtained using this technique. The resolution of the images are 1024×1024. Based on results presented by Azzopardi *et al*., which show an expected positive correlation between resolution and σ, we limit the search space to $\sigma = [4.8, 4.9, ..., 5.1]$. We choose $\rho_{max} = 20$ as the maximum radius of the concentric circles, and following Azzopardi *et al*. we increment the circles by 2 pixels. Therefore the radii of the circles that are considered are $\rho = [0, 2, 4, ..., 20]$. σ_0 is limited to $\sigma_0 = \{1, 2, 3\}$ and α to $\alpha = \{0.1, 0.2, ..., 0.7\}$.

Once we choose the combination of parameters for the symmetric filter that yields the best average *MCC* over the images in the training set, we fix these parameters and search for a good combination of parameters for the asymmetric filter. Note that we know in advance that vessel endings are usually thinner, and therefore we search for σ_a in a value range that is strictly smaller than the obtained σ_s.

5 Results

In Figure 5 the ROC curve is traced out for the segmentation of the training images in the IOSTAR dataset with the set of parameters in Table 5, using the manually segmented ground truth images as the correct classification against which the filter segmentation is compared. The threshold with the highest average MCC value is also indicated, which turns out to be a threshold of 35. In Table 6 we see the resulting AUC value and also the segmentation accuracy corresponding with the best threshold. In Figure 4 we see an original IOSTAR training image alongside the segmented image using the filter parameters in Table 5 and threshold $t = 35$. In line with the good performance metrics, we can also see that the resulting segmentation is quite accurate in the sense that the main vessel tree has been segmented correctly and even the tinier vessels are for the most part in the segmentation.

From an experiment in [1] on the DRIVE retinal image dataset, it has been established that the parameter α is the most sensitive parameter, i.e., a correct configuration of this parameter is most determinant



(a) An IOSTAR retinal fundus image (b) Segmented with filter

Fig. 4: *Left*: An original training image from the IOSTAR dataset. *Right*: The corresponding segmentation using the trained B-COSFIRE filters with parameters specified in Table 5 and threshold $t = 35$.

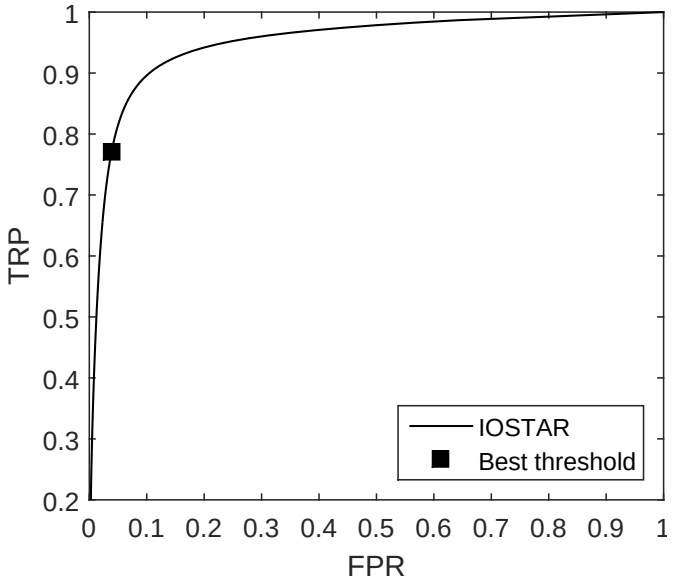


Fig. 5: The ROC curve obtained by varying the segmentation threshold from 0 to 255 which yields for each threshold a true positive rate (TPR) and a false positive rate (FPR). Indicated with a square is the TPR and FPR of the threshold $t = 35$, which yielded the best average MCC value.

for the performance of the segmentation. Here we perform a similar experiment for the IOSTAR dataset: From the parameter values as listed in Table 5 for the symmetric filter, we inspect the sensitivity of each parameter at a time by changing the parameter value by steps of 0.1 from the optimal value and keeping the rest of the parameters at their optimal values. We then compare the resulting 30 MCC values with the 30 MCC values obtained for the optimal parameter values in Table 5 by a two-tailed paired t-test with significance level $p = 0.05$. We regard the obtained MCC values to be significantly different from the optimal MCC values if $p < 0.05$. The t-values for which this is true are bold-faced in Table 7.

As can be see in Table 7, the parameters have a different level of sensitivity. From our experiments, the parameter α turns out to be the most sensitive parameter. By a small change the t-statistic changes considerably and we found that even a change of 0.1 yields a significantly different segmentation performance. We observe that the parameters σ_0 or σ are less sensitive, as the change in t-value is smaller compared to the case of α. However, we notice that the σ parameter is more sensitive when it is increased from $\sigma = 4.8$ than if it is decreased. The reverse is true for σ_0: It is less sensitive when increased

	AUC	MCC	Accuracy	Se	Sp
IOSTAR	0.9519	0.6979	0.9419	0.7705	0.9613

Table 6: IOSTAR segmentation accuracy with the parameters specified in Table 5, using the segmentation threshold $t = 35$.

	σ_0	σ	α
-0.5	**5.0363**	-0.7858	-
-0.4	**3.7276**	-0.5248	-
-0.3	**2.4602**	-0.3723	-0.9215
-0.2	1.3496	-0.1580	1.7180
-0.1	0.5614	-0.0862	**5.8397**
0	-	-	-
+0.1	-0.6672	-0.4170	**-2.4933**
+0.2	-1.1272	-0.9986	**-3.6321**
+0.3	-1.5860	-1.9217	**-4.3719**
+0.4	-1.9512	**-2.5296**	**-4.9806**
+0.5	**-2.2660**	**-3.1036**	**-5.4493**

Table 7: An experiment to study the sensitivity of the parameters σ_0, σ and α for a symmetric B-COSFIRE filter on the 30 images of the IOSTAR dataset. Each row compares the 30 MCC values for the optimal parameters with the 30 MCC values for the parameter values for the corresponding row. Each row indicates the t-statistic after changing the parameter value by the offset in the first column from the optimal parameter value and keeping the remaining two parameters at their optimal values. Bold-faced are the t-values for which the null-hypothesis is rejected with significance level $p < 0.05$.

from $\sigma_0 = 3$ and more sensitive when decreased. The result that α is the most sensitive parameter is in line with the result in [1].

6 Summary and Discussion

In this contribution we discussed various methods for the automatic segmentation of vessels in retinal fundus images. In this section we discuss how they perform compared to the B-COSFIRE method and our insights about the B-COSFIRE method.

6.1 Deep Neural Networks

Given the results of Liskowski *et al.*, their neural network aproach performs better than B-COSFIRE in general. In all cases, the neural network achieves a higher AUC value than B-COSFIRE. However, in terms of accuracy and specificity, B-COSFIRE does outperform the balanced configuration of the network on the DRIVE and STARE dataset. For the CHASE dataset, the neural network outperforms B-COSFIRE.

However, in terms of training, the B-COSFIRE method is cheaper, as Liskowski *et al.* report that training the network can take up to 8 hours on a single GPU. However, training times are not interesting for the end user, as one usually does not need to train the classifier for each new image. To classify an image, the neural network needs 92 seconds using a high end GPU. So, for end users with low end hardware and a limited budget, the B-COSFIRE method could be a better choice in terms of time, because the B-COSFIRE method only needs 10 seconds on a 2 GHz CPU. However, if the hardware or budget is available, the neural network is suitable, because we consider 92 seconds still an acceptable time, given the experiment results.

6.2 Ensemble classification-based approach

Compared to the B-COSFIRE method, the method of Fraz *et al.* is more accurate when tested on the three datasets (DRIVE, STARE and CHASE), however, the B-COSFIRE method has a higher sensitivity or true positive rate (TPR), i.e. B-COSFIRE is more able to detect *vessel* pixels. The method of Fraz *et al.* performs better than B-COSFIRE, but the B-COSFIRE is faster. B-COSFIRE only needs 10 seconds to process an image from the DRIVE or STARE datasets whereas the ensemble classification method needs 2 minutes.

The method of Fraz *et al*. performs slightly better than B-COSFIRE, but requires expensive training with retinal fundus images. The parameters of the B-COSFIRE method must either be tweaked manually by hand or be determined using a training set of the images. Based on experiments with the intuitive parameters of B-COSFIRE, manually tweaking the parameters quickly yields a good performance which is a major advantage of the B-COSFIRE method.

6.3 Fully Connected Conditional Random Field Model

Orlando *et al*. did not provide AUC and accuracy metrics, therefore, we only use the sensitivity and specificity to compare the Fully Connected Conditional Random Field method with B-COSFIRE. As it turns out, there is no winner. For the DRIVE dataset, Orlando *et al*. achieve higher sensitivity, but lower specificity. For the STARE dataset, it is the other way around and for the CHASE dataset, Orlando *et al*. achieve also less sensitivity and more specificity.

6.4 B-COSFIRE

Furthermore, we have studied the inner workings of the B-COSFIRE method and subsequently applied the method for a set of parameters to the recent IOSTAR dataset. We have seen that B-COSFIRE could achieve an accurate and fast segmentation on this dataset.

B-COSFIRE has turned out to be a fast trainable filter-based method that can definitely compete with much more complex methods that are based on machine learning methods like neural networks, that take a considerable amount of time to train. Given the complexity of those methods and B-COSFIRE's potential to compete well and its intuitive parameters, B-COSFIRE could even be the preferred method to use, especially when having less computational power at hand.

6.5 Other fields and future work

Segmentation methods like the ones we have discussed in this paper are not solely useful for blood vessel segmentation. In fact, one could use these methods to segment rivers from satellite images, segment the nerves of a leaf, to find cracks in concrete structures, in fact these methods could provide useful in all applications in which elongated structures need to be segmented.

For future work, any of these methods could be extended to not only extract the blood vessel tree but to also detect obstacles that could impede the blood flow. An additional benefit of B-COSFIRE is that its selectivity is not pre-programmed into the method itself but it is achieved by prototype patterns on which the filter is automatically configured. This gives the possibility to extend the filter's selectivity to all kinds of more complex structures which gives rise to applications other than vessel segmentation alone.

References

[1] G. Azzopardi, N. Strisciuglio, M. Vento, and N. Petkov, "Trainable cosfire filters for vessel delineation with application to retinal images," *Medical image analysis*, vol. 19, pp. 46–57, 1 2015.

[2] M. M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. R. Rudnicka, C. G. Owen, and S. A. Barman, "An ensemble classification-based approach applied to retinal blood vessel segmentation," *IEEE Transactions on Biomedical Engineering*, vol. 59, pp. 2538–2548, Sept 2012.

[3] P. Liskowski and K. Krawiec, "Segmenting retinal blood vessels with deep neural networks," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 2369–2380, Nov 2016.

[4] J. I. Orlando, E. Prokofyeva, and M. B. Blaschko, "A discriminatively trained fully connected conditional random field model for blood vessel segmentation in fundus images," *IEEE Transactions on Biomedical Engineering*, vol. 64, pp. 16–27, Jan 2017.

[5] J. Zhang, B. Dashtbozorg, E. Bekkers, J. P. W. Pluim, R. Duits, and B. M. ter Haar Romeny, "Robust retinal vessel segmentation via locally adaptive derivative frames in orientation scores," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 2631–2644, Dec 2016.

[6] G. Azzopardi and N. Petkov, "Trainable cosfire filters for keypoint detection and pattern recognition," *Ieee transactions on pattern analysis and machine intelligence*, vol. 35, pp. 490–503, 2 2013. Relation: http://www.rug.nl/research/jbi/ Rights: University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science.

[7] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken, "Ridge based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 501–509, 2004.

[8] A. M. Mendonca and A. Campilho, "Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction," *IEEE Transactions on Medical Imaging*, vol. 25, pp. 1200–1213, Sept 2006.

[9] J. V. B. Soares, J. J. G. Leandro, R. M. Cesar, H. F. Jelinek, and M. J. Cree, "Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification," *IEEE Transactions on Medical Imaging*, vol. 25, pp. 1214–1222, Sept 2006.

[10] E. Ricci and R. Perfetti, "Retinal blood vessel segmentation using line operators and support vector classification," *IEEE Transactions on Medical Imaging*, vol. 26, pp. 1357–1365, Oct 2007.

[11] M. Niemeijer, J. Staal, B. van Ginneken, M. Loog, and M. Abramoff, "Comparative study of retinal vessel segmentation methods on a new publicly available database," in *SPIE Medical Imaging* (J. M. Fitzpatrick and M. Sonka, eds.), vol. 5370, pp. 648–656, SPIE, SPIE, 2004.

[12] M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. Rudnicka, C. Owen, and S. Barman, "Blood vessel segmentation methodologies in retinal images a survey," *Computer Methods and Programs in Biomedicine*, vol. 108, no. 1, pp. 407 – 433, 2012.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jan. 2014.

[14] "DRIVE Dataset." `http://www.isi.uu.nl/Research/Databases/DRIVE/`, 2004. [Online; accessed March 5th, 2017].

[15] A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response," *IEEE Transactions on Medical Imaging*, vol. 19, pp. 203–210, 2000.

[16] "STARE Dataset." `http://cecas.clemson.edu/~ahoover/stare/`, 2004. [Online; accessed March 5th, 2017].

[17] "CHASE Dataset." `https://blogs.kingston.ac.uk/retinal/chasedb1/`. [Online; accessed March 5th, 2017].

[18] "IOSTAR Dataset." `http://www.retinacheck.org/`, 2015. [Online; accessed 27-Febr-2017].

Analysis of Optimisation Methods to Improve Data Centre Efficiency

D.I. Pavlov and R.M. Bwana

Abstract—Due to the increase in data centres' demand on energy grids, recent research suggests multiple approaches to reduce energy consumption and costs. The paper highlights literature that focuses on data centre optimisation, analyses papers that pursue optimisation through the manipulation of multiple appropriate parameters through certain criteria, and then discuss the results those papers achieved. The results obtained show that energy efficiency can be achieved through the optimisation of multiple varying parameters and that the optimal solution can only be determined based on the primary concern of those involved.

Index Terms—Data centres, Energy efficiency, Energy optimisation, Cost reduction.

1 Introduction

Data centres play an important role in the modern Internet infrastructure and are built in a geographically separated manner in order to reduce latency and increase availability. This has resulted in an increase of the associated costs such as energy consumption and environmental impact. The authors in [4] state that approximately 15% of the costs associated with data centres take the form of utility costs such as electrical supply. This has led to a sizeable amount of energy being consumed, nearly 70 billion kiloWatt-hours (kWh) in 2014 in the U.S. [1]. With the demand for productivity from data centres following an upward trend it is therefore in the interest of stakeholders to increase their efficiency and in turn lower the costs as well as the environmental impact that they may have.

The challenge regarding optimisation of data centre performance is the presence of competing goals in terms of energy usage, latency when responding to requests, throughput, and fairness in providing availability for requests [15]. Companies and stakeholders are likely to have different priorities regarding these metrics and so require different optimisation decisions.

Our work provides an analysis of research that focuses on optimising data centre performance using optimisation techniques. These involve the definition of a cost function in terms of an equation where the purpose of the model is then to minimise the solution of the equation. The term *cost* is not limited to financial costs but can represent any metric that would be considered negative to the scenario, including but not limited to wasted electricity, number of packets waiting in queue, and latency of communication. Once defined the model would actively seek to minimise these costs and in doing so achieve the desired efficiency behaviour.

The research papers are evaluated based on the following criteria: Reduced Costs, Power Efficiency, Performance Impact and Alternative energy supply.

Our research aims to determine which of the various solutions that have been proposed best achieve data centre efficiency through optimising a combination of these criteria.

The remainder of this paper is organised as follows: in section 2, we discuss literature relating to the topic of optimisation in data centres in terms of the varying parameters that could be adjusted. In section 3, we will outline the methodology and present the papers found to best satisfy the criteria stated and discuss these results further in context in section 4. We present our conclusions in section 5 and suggest future work in section 6.

- *D.I. Pavlov is a MSc Computer Science student at the University of Groningen, E-mail: d.i.pavlov@student.rug.nl.*
- *R.M. Bwana is a MSc Computer Science student at the University of Groningen, E-mail: r.m.bwana@student.rug.nl.*

2 Related Work

What follows is an overview of the available literature and their similarities and differences starting with more widespread scenarios that describe the alternatives and gradually focusing on the topic of this paper. For the sake of simplicity we have organised the studied examples in terms of those that pursue optimisation through load balancing and those that pursue optimisation through the use of alternate energy sources and energy storage solutions (ESS).

2.1 Load Balancing

Research has been conducted into lowering energy costs through the monitoring of market prices and using it to redirect traffic to data centres in lower priced areas by Qureshi *et al.* [13]. In their research they analyse the effects of a system's energy elasticity[1], the bandwidth - performance balance[2], and the benefits of redistributing the load to other locations. Doing so passes on the processing burden to a geographic area where it is more cost effective to do so. Such an approach would be best suited to batch operations which are less time-sensitive and are likely to achieve higher utilisation levels of the processing resources. This also does not take into consideration the source of the energy, merely its market price.

Pinheiro *et al.* [12] propose a solution which reduces the number of active nodes and redistributes their load to other nodes, sacrificing the overall performance. This approach considers that the power which a certain node consumes, even though it is idle, is relatively high and therefore a solution is sought by removing or adding nodes based only on the current demands. Such an approach, although achieving good results, is impractical in cases where low latency and response times are a primary concern.

Optimisation methods using cost functions are not the only machine learning techniques used to handle energy optimisation however. Motamedi *et al.* [10] prove that using data association mining can help in forecasting short-term power demands with some degree of accuracy. In their research they show that by examining historical data it is possible to make an accurate prediction about the expected power consumption in the near future. While the focus is on residential consumers, their work can be used in the context of data centres in order to predict upcoming load and estimate the resource needs.

Lin *et al.* [7] use online optimisation methods to determine the number of servers to run in order to meet the demand while reducing power costs. Servers not deemed necessary are then shut down in order to reduce the power consumption. This model of saving energy does not regard the source of energy nor the cost of supplier but simply minimises the on-time of the servers in the data centre.

Raghavendra *et al.* [14] propose a power management oriented architecture which is based on the coordination of several different ap-

[1] Energy elasticity is defined as the degree to which the energy consumed by a cluster depends on the load placed on it

[2] In their experiments the authors use the term 95/5 Constraint which is used to identify that each routing component should be utilised up to 95 percent of its optimal bandwidth

proaches. Their research examines the diversity of power optimisation methods and some of the conflicts which may arise if they are applied without coordination, such as excessive overload, while trying to optimise utilisation. They divide the problem in two: average power optimisation and peak power optimisation. As a result, they end up with an architectural solution which aims to optimise power consumption on different levels ranging from a single server to the entire data centre.

The energy supply is not the only resource in a data centre that can be optimised. Georgiadias *et al.* [3] suggests a method to monitor and optimise network traffic systems using Lyapunov optimisation in order to reduce the latency and wait times of requests. This is not only to the benefit of the source of the requests but also reduces the idle time of servers.

Liu *et al.* [8] presents an alternative model and while similarly using Lyapunov stability, it is intended to maximise the virtual machine (VM) scheduling and the packet admission and routing while observing a power budget designed to maximise profitability. This approach prioritises lower latency and response times over the amount of energy consumed as long as it maintains a predetermined maximum energy consumption.

2.2 Alternative Energy Sources and Energy Storage

The notion of using optimisation methods, specifically Lyapunov optimisation, to handle electricity supply and balancing is proposed as a Welfare Maximisation Algorithm (WMA) in [6]. This is shown to work best when the demand can be forecast to some reasonable extent which may not always be possible with data centres. While certain aggregate workloads can be forecast, a consideration that must be taken is that data centres have to easily respond to spikes in work rates which could be costly in scenarios where the primary source of emergency energy is the power grid. This problem is addressed in [5] where excess supply in electricity can be stored in finite-sized energy storage systems (ESS) when prices are lower and discharged or consumed at a later time should demand exceed planned electrical supply capacity.

The notion of handling and supplying energy from a renewable energy source/supplier is introduced and modelled in Neely *et al.* [11]. This research takes into consideration the variable and sometimes erratic supply of renewable energy which is a constant factor discussed in related papers.

Deng *et al.* [2] discusses a combined multi-source approach using stochastic optimisation and a two stage Lyapunov optimisation with the intention of minimising the cost of supplying power dynamically to a data centre. This combines reliable grid power with unreliable renewable energy with the aim to lower power costs and carbon emissions. This approach favours renewable energy sources where possible and only should the energy demand exceed the supply from renewable energy sources does the data centre use power supplied from the grid.

Liu *et al.* [9] uses geographic load balancing in order to use the least amount of energy and considers local renewable technologies and storage solutions. Yu *et al.* [17] also discuss geographical load distribution but considering power outages and exploiting renewable energy. They formulate it as stochastic optimisation problem which aims for long-term energy cost minimisation of distributed data centres. In their research they also consider selling back stored energy in to the main grid in order to minimise the overall energy costs.

3 Analysis

In this section we present the results of Qureshi *et al.* [13], Pinheiro *et al.* [12] and Yu *et al.* [17]. These were chosen as they were found to best incorporate the outlined metrics of reducing overall costs, power efficiency, the use of renewable sources, and their performance impact.

Based on the observed results each method's component is graded as follows :

- ✓- positive results
- X - negative results
- N/A - not applicable

3.1 Distance based load redistribution

Qureshi *et al.* [13], by applying their distance based approach in their experiments came up with the results shown in Figure 1.

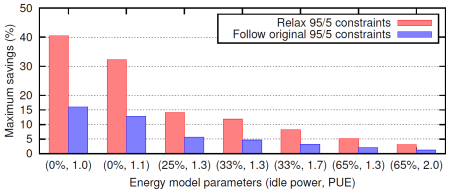


Fig. 1: Energy elasticity, taken from [13]

Based on the results in Figure 1, the authors discover that a system's efficiency highly depends on its energy elasticity and not so much on its location.

In their following experiments they outlined the relation between the energy cost and distance based redistribution. When a request arrives for a specified price and distance range the most suitable server is chosen, if this server is at the peak of its capacity iteratively another one is chosen. The results they achieved by doing this are shown in Figure 2 and Figure 3.

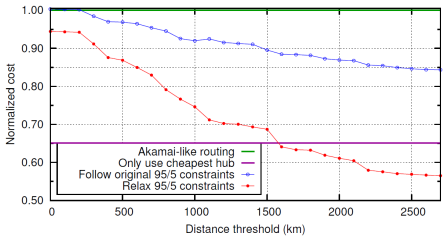


Fig. 2: 39-month electricity costs based on distance threshold. Taken from [13]

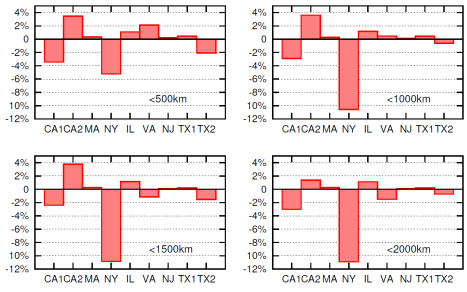


Fig. 3: 39-month per-server electricity costs by using distance threshold. Taken from [13]

We can see from Figure 2 and Figure 3 that this approach manages to provide long-term reduced energy costs. Main disadvantage however is that the available servers are not fully utilised. Their work is redistributed to another data centre, while they remain idle and as result there is no real power efficiency achieved. The performance however remains relatively high. No performance degradation is observed and the high number of available servers is able to process unexpected load peeks. The lack of alternative energy supply and relying only on the main grid can be considered as a major disadvantage to this method.

Table 1: Distance based load redistribution results

Reduced Costs	Power Efficiency	Performance	Alternative energy
✓	X	✓	N/A

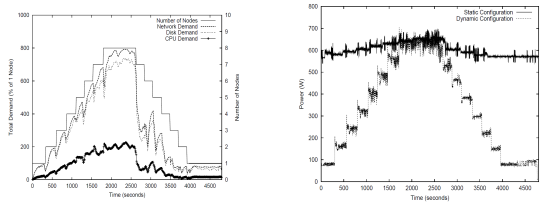
3.2 Intra-centre load relocation

In contrast to Qureshi *et al.* Pinhero's approach [12] offers intra-centre load relocation. In their experiments they used 8 personal computers (PCs) connected in cluster mode running web-servers. They implement the algorithm defined by the pseudo-code in Algorithm 1

Algorithm 1 Dynamic Configuration Load Balancing

```
Periodically :
  if Removal is acceptable then
      Choose nodes (victims) with low demand to be turned off.
      if There are any victims then
          Determine nodes to receive their load and ask victims to migrate their load out. Ask victims to turn themselves off
  else
      if Addition is necessary then
          Turn on new nodes
          if Necessary load to be sent to added nodes then
              Ask nodes to share their load with added nodes
```

A set of tests was then conducted. One without the implementation of the algorithm (static configuration) and another one with the implemented algorithm (dynamic configuration). The results are shown in Figure 4.



(a) Cluster power demands (b) Cluster power consumption

Fig. 4: Graph representing cluster demands/power consumption. Taken from [12]

As can be seen from Figure 4, a significant improvement is achieved when applying the dynamic configuration compared to the static one. The downside is that in this case the overall throughput suffers.

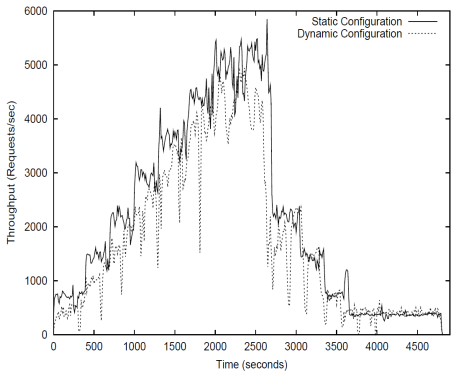


Fig. 5: Throughput of the web-server under static and dynamic configuration. [12]

From the results in Figure 5 we can see that this approach manages to overcome some of the problems of the distance based load redistribution. It manages to reduce the costs and at the same time achieve real power efficiency. There is a trade-off however. In order to achieve these results the performance is sacrificed. Another problem which should be considered as well is the lack of alternative energy supply.

Table 2: Intra-centre load balancing

Reduced Costs	Power Efficiency	Performance	Alternative energy
✓	✓	X	N/A

3.3 Alternative Energy Sources and Energy Storage

The research performed by Yu *et al.* [17] is considered further and the results obtained presented. The research identifies the source of energy as the main grid which could be connected to renewable sources but is not explicitly modelled. The research does include the use of an energy storage system (ESS) identified as a 'storage bank' and diesel generators as an alternate source of energy in emergency situations caused by power outages in the smart grid [16]. Figure 6 shows the energy management architecture studied in the research indicating the various sources and loads that would feature in such a scenario.

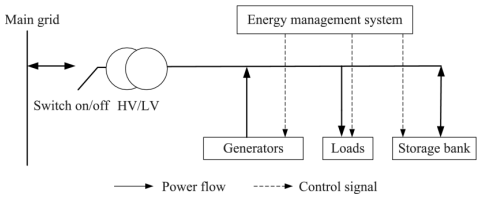


Fig. 6: Smart micro-grid architecture studied in [17]

The architecture shown in Figure 6 shows the connection to the main grid as well as the energy management system (EMS) which regulates the amount of energy produced by the generators, the amount of energy consumed by the work loads, and the amount of energy stored/retrieved from the energy storage bank.

Yu *et al.* go on to define a cost function outlined in equation 1 that seeks to minimise the expected average costs across

$$\min \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} E\{C(t)\}, \tag{1}$$

where $C(t)$ is calculated as:

$$C(t) = \sum_{i=1}^{N} \left\{ G_i^{2g}(t) S_i(t) - G_i^{2g}(t) W_i(t) \right\} \Delta T + \sum_{i=1}^{N} K_i(t) + \sum_{i=1}^{N} \left(Z_i^{bc}(t) + Z_i^{bd}(t) \right) \rho_i^b, \tag{2}$$

where T represents the time period being considered, $C(t)$ represents the total energy cost at time t, $G_i^{2g}(t)$ represents the power purchased from main grid i at time t with a purchasing price represented by $S_i(t)$, $G_i^{g2}(t)$ represent the energy sold back to grid i at time t with a selling price represented as $W_i(t)$, $K_i(t)$ represents the total energy cost of diesel generators, $Z_i^{bc}(t)$ and $Z_i^{bd}(t)$ represent whether the battery is a charging or discharging state respectively with ρ_i^b representing the cost of the charging/discharging action.

The cost function in Equation 1, expanded through Equation 2 outlines the costs of the data centres as a sum of: the costs incurred through purchasing energy subtracting the revenue made from selling energy back to the grid, the cost of generating energy through a diesel generator, and the cost of storing/discharging energy.

After formulating the optimisation problem the algorithm was tested based on real world traces and results are shown in Figure 7

and Figure 8. In these figures, V_{max} represents the maximum of a constant for the tradeoff between minimising expected energy cost and ensuring the stability of virtue queues. Figure 7 shows the total energy costs in terms of US Dollars when the main energy grid does not feature renewable sources. Alternately, Figure 8 shows the total energy costs should renewable sources, in this case wind energy, be included in the main energy grid.

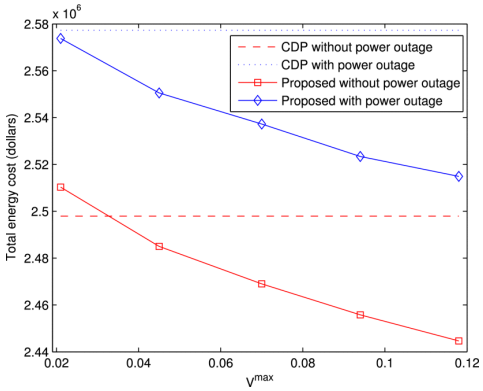


Fig. 7: Total costs when grid has no renewable sources taken from [17]

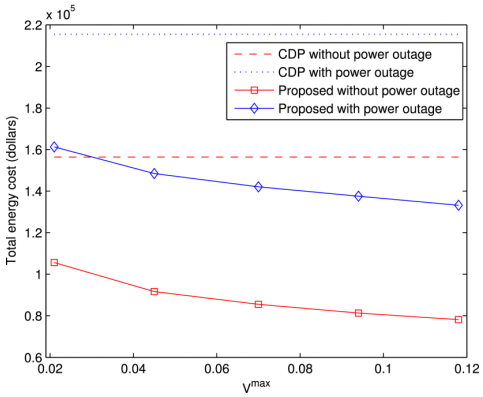


Fig. 8: Total costs when grid has renewable sources taken from [17]

Two primary observations can be made from Figure 7 and Figure 8. Firstly, both figures show a negative gradient as the value of V_{max} increases for the scenario when power outages occur and without power outages. Subsequently, the inclusion of renewable energy sources in the grid lowers the overall costs involved by a significant margin. As stated by Yu *et al.*, the costs involved in the deployment of the renewable energy sources are not factored in to the costs of this model but would have an impact on the results should they be.

Table 3: Alternative energy sources results

Reduced Costs	Power Efficiency	Performance	Alternative energy
✓	N/A	N/A	✓

4 Discussion

All three approaches analysed are seen to achieve the goal of optimising data centre efficiency although in different manners.

By applying the load distribution method up to 40% [13] can be saved just by redirecting the task to another geographic area where the price is lower. Although the numbers show a sizeable reduction in costs, this strategy has one very serious flaw - the same energy is consumed by another server but in a different location. Another issue is that in order for this to be achieved, a fast, reliable infrastructure is required so that the requests can still be handled in a reasonable time period. This may lead to investments in expensive hardware which causes increased expenditures.

One possible solution to some of the above mentioned problems would be to minimise the energy consumption of the data centre by forcing some nodes to run at higher load while others are turned off to preserve energy, as stated in Pinhero *et al.* [12]. During their research they measured up to 43% savings per cluster, and up to 86% when resource demands require only one node. All of this comes at the cost of performance; they observe a 23% drop in performance. Apart from the performance degradation another problem which may occur is quicker hardware amortisation. Some of the machines may receive excessive load for significantly longer periods than the others which may lead to either single component or full machine failure. This may result in unexpected downtime and high maintenance costs.

The downside of these methods is that the improvement is achieved at some cost. In the first case, even though the final results show that moneywise the algorithm is effective, in terms of energy consumption this is not true. Another place where this method fails is that in order to be implemented a relatively expensive investment in machines in different locations has to be made. On the other hand such an approach proves to be effective in systems where low latency and high availability are of the essence. The achieved performance through load balancing and unbalancing show that even a reduced number of active machines can perform at a sufficient level. This can successfully be applied in larger data centres with poor resource utilisation or in situations where there is no constraint on the response time. As stated above, this may lead to higher maintenance costs.

Using alternative energy sources presents an opportunity to overcome the high energy costs of running a data centre as well as lower carbon emissions. As shown by Yu *et al.*, alternative power sources prove to be efficient especially in an environment where power outages can be considered a possibility. Using alternative energy sources can increase the energy efficiency of a given data centre significantly. As the results show, a three-fold decrease is achieved during testing. As is the case with renewable energy sources, it is highly dependent on the geographical area where given data centre is located - solar energy generators are more suitable in sunny areas, such as Spain, Italy, deserts, etc., while wind turbine generators are more suitable in geographic areas with higher wind density throughout the year.

Other non-renewable alternative energy sources, such as diesel generators, are an alternative to survive power outages as they manage to provide the required energy in such situations but come at the price of environmental costs. A combination of alternative sources and main grid proves to be useful in cases where redistribution to another geographical area is not feasible, electricity prices vary hourly or the main grid may experience outages due to unforeseen accidental failures or more intentional and malicious causes [16].

Through our research we therefore determine that there is no ideal solution to optimise data centre efficiency, but rather the solution that would best be adopted depends on the primary concern. A significant reduction in costs can be achieved by all three models, however those involved would have to determine whether they would prioritise performance (Qureshi *et al.*), power efficiency (Pinhero *et al.*), or the use of alternate energy sources (Yu *et al.*). Exactly which combination would be best suited for whom was beyond the scope of this paper but could be considered as potential future work.

5 Conclusion

Throughout this paper we have discussed the various optimisation methods used to reduce the energy consumption in data centres or improve data centre efficiency. With energy consumption increasing its importance cannot be underestimated. Given the number of techniques to achieve optimisation such as load distribution, queue optimisation and virtual machine scheduling as well as the levels on which they are applied such as networking, single server, cooling, and entire data centres, efficiency in data centres can be achieved through various methods.

Even though the U.S. Data Center energy usage report [1] shows

that for the past ten years the development in this field has managed to adequately serve the increasing number of consumers, the projections show that in the near future demand is going to increase even further. With this in mind, the fact that data centres are still one of the main energy consumers only confirms that more research in this field is required.

Through our research we found that the models developed by Qureshi *et al.* [13], Pinhero *et al.* [12], and Yu *et al.* [17] all achieve the objective of increasing the energy efficiency in data centres, although through the optimisation of different parameters. This indicates that data centre efficiency can be obtained through multiple techniques depending on the trade-off of the primary criteria concerned.

6 Future Work

As a natural follow up to this paper, the models could be replicated and analysed using a single data set in order to calculate a comparable metric for all models. This would allow us to attempt to determine the most power efficient model based with respect to the metric chosen.

It would also be of interest to the authors of this paper to attempt to determine the feasibility of combining the various approaches of the papers discussed into one algorithm and measure the effectiveness and achieved efficiency of such a combination of approaches.

Acknowledgements

The authors wish to thank Brian Setz for his expert review as well as M.J.Ch Straat and F.N. Mol who have helped this paper achieve all it could.

References

[1] United states data center energy usage report. https://eta.lbl.gov/sites/all/files/publications/lbnl-1005775_v2.pdf.

[2] W. Deng, F. Liu, H. Jin, and X. Liao. Online control of datacenter power supply under uncertain demand and renewable energy. In *Communications (ICC), 2013 IEEE International Conference on*, pages 4228–4232. IEEE, 2013.

[3] L. Georgiadis, M. J. Neely, L. Tassiulas, et al. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends® in Networking*, 1(1):1–144, 2006.

[4] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The cost of a cloud: research problems in data center networks. *ACM SIGCOMM computer communication review*, 39(1):68–73, 2008.

[5] L. Huang, J. Walrand, and K. Ramchandran. Optimal demand response with energy storage management. In *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*, pages 61–66. IEEE, 2012.

[6] L. Huang, J. Walrand, and K. Ramchandran. Optimal power procurement and demand response with quality-of-usage guarantees. In *Power and Energy Society General Meeting, 2012 IEEE*, pages 1–8. IEEE, 2012.

[7] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM Transactions on Networking (TON)*, 21(5):1378–1391, 2013.

[8] F. Liu, Z. Zhou, H. Jin, B. Li, B. Li, and H. Jiang. On arbitrating the power-performance tradeoff in saas clouds. *IEEE Transactions on Parallel and Distributed Systems*, 25(10):2648–2658, 2014.

[9] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew. Greening geographical load balancing. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 233–244. ACM, 2011.

[10] A. Motamedi, H. Zareipour, and W. D. Rosehart. Electricity price and demand forecasting in smart grids. *IEEE Transactions on Smart Grid*, 3(2):664–674, 2012.

[11] M. J. Neely, A. S. Tehrani, and A. G. Dimakis. Efficient algorithms for renewable energy allocation to delay tolerant consumers. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 549–554. IEEE, 2010.

[12] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath. Load balancing and unbalancing for power and performance in cluster-based systems. In *Workshop on compilers and operating systems for low power*, volume 180, pages 182–195. Barcelona, Spain, 2001.

[13] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for internet-scale systems. In *ACM SIGCOMM computer communication review*, volume 39, pages 123–134. ACM, 2009.

[14] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu. No power struggles: Coordinated multi-level power management for the data center. In *ACM SIGARCH Computer Architecture News*, volume 36, pages 48–59. ACM, 2008.

[15] S. Ren, Y. He, and F. Xu. Provably-efficient job scheduling for energy and fairness in geographically distributed data centers. In *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, pages 22–31. IEEE, 2012.

[16] J. Stamp, A. McIntyre, and B. Ricardson. Reliability impacts from cyber attack on electric power systems. In *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*, pages 1–8. IEEE, 2009.

[17] L. Yu, T. Jiang, and Y. Cao. Energy cost minimization for distributed internet data centers in smart microgrids considering power outages. *IEEE Transactions on Parallel and Distributed Systems*, 26(1):120–130, 2015.

university of groningen / faculty of science and engineering / computing science