



University
of Glasgow | School of
Computing Science

Algorithms for Sports Elimination

Gordon Reid: 1002536R

Ryan Wells: 1002253W

Kris Stewart: 1007175S

David Selkirk: 1003646S

James Gallagher: 0800899G

Level 3 Project — March 5, 2013

Abstract

The primary aim of the project is to implement the Ford-Fulkerson algorithm that, given a league table, remaining schedule of matches, and a team, will determine whether or not the given team has a chance of finishing at the top of the league table. The results will then be displayed on a user interface. The user interface also allows the user to navigate between weeks to see how the league progressed and when teams were first eliminated.

Acknowledgements

We would like to thank our project supervisor, Dr. David Manlove, for his extensive guidance, knowledge, and support throughout the project. We would also like to thank all of the participants of our user evaluation during the evaluation stage of our project.

Education Use Consent

We hereby give our permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____ Signature: _____

Name: _____ Signature: _____

Name: _____ Signature: _____

Name: _____ Signature: _____

Name: _____ Signature: _____

Contents

1	Introduction	5
1.1	Motivation	5
1.2	Background	5
1.3	Aims	5
1.4	Outline	6
2	Preliminaries	7
2.1	Terminology	7
2.2	Graph Theory and Network Flow	8
3	Design and Implementation	9
3.1	Desktop User Interface	9
3.1.1	Design	9
3.1.2	Implementation	9
3.2	Ford-Fulkerson Algorithm	10
3.2.1	Design	10
3.2.2	Implementation	10
3.3	Parser	11
3.3.1	Design	11
3.3.2	Implementation	11
3.4	Web Application	12
3.4.1	Design	12

3.4.2	Implementation	13
4	Evaluation	14
4.1	Correctness Testing	14
4.2	Evaluation	14
4.2.1	Introduction	14
5	Conclusion	16
5.1	Summary	16
5.2	Future Work	16
5.3	Lessons Learned	16
6	Contributions	17
6.1	Gordon Reid	17
6.2	Ryan Wells	17
6.3	Kris Stewart	17
6.4	David Selkirk	17
6.5	James Gallagher	17
A	Installation and Running of Application	18
A.1	Desktop Application	18
A.1.1	Installation	18
A.1.2	Running	18
A.2	Web Application	18
A.2.1	Installation	18
A.2.2	Running	19
B	User Manual	20
C	Web Application Wireframe	21

D Web Application N-Tier Architecture Diagram	22
Bibliography	22

Chapter 1

Introduction

1.1 Motivation

1.2 Background

The project builds on two papers: one by K. D. Wayne, ‘A new property and a faster algorithm for baseball elimination’, SIAM Journal on Discrete Mathematics <http://epubs.siam.org/doi/abs/10.1137/S0895480198348847> [3] and one by I. Adler, A. L. Erera, D.S. Hochbaum and E. V. Olinick, ‘Baseball, optimization and the world web’, manuscript, 1998 <http://riot.ieor.berkeley.edu/~dorit/pub/baseball.ps> [1].

The project came about from the American love for baseball, and the love for a multitude of statistics relating to baseball. A common news story for sports fans is the announcement that a team can no longer win the current season however the calculations used to work this out are typically naïve and do not take into account the remaining schedule of games.

The papers stated above, build on the naïve calculations and make use of a simple, yet effective, algorithm to reliably answer the question: ‘Is it mathematically impossible for a given team to win the baseball league?’ The algorithm is known as the Ford-Fulkerson algorithm, named after L. R. Ford, Jr. and D. R. Fulkerson. The Ford-Fulkerson algorithm removes the need for an exhaustive search over all of the possibilities of results the remaining matches can have.

1.3 Aims

The primary aim of the project is to implement the aforementioned Ford-Fulkerson algorithm that, given a league table, remaining schedule of matches, and a team, will determine whether or not the given team has a chance of finishing at the top of the league table. The results will then be displayed on a user interface with data being obtained from either a text file or parsing dynamic up-to-date information from a sports website. The user interface will be fairly simplistic with a large central table displaying the league and elimination information with buttons and/or radio buttons for navigating between leagues. The user interface may also allow the user to navigate between weeks

to see how the league progressed and when teams were first eliminated. A web-based version of the user interface is of interest and will be explored.

There may be various ways the given team can be eliminated from the league and a secondary objective is to provide a ‘certificate’ of elimination. This will provide the team, or teams, that are responsible for ensuring the given team cannot finish top of the league table. A simple example can be taken from the naïve calculations traditionally used by sports pundits. The teams responsible for the elimination is just the list of teams that have a points differential with the given team that is greater than the number of games the given team has to play.

1.4 Outline

Chapter 2

Preliminaries

2.1 Terminology

The web application design and implementation section discusses many web technologies along with their acronyms. The acronyms will be described with a brief overview of the technology.

- **HTML:** Hyper Text Markup Language. HTML is the basis of every web page structure. It is a structured document with tags denoted by angle brackets (such as `<html>`). These tags form a nested, tree-like structure that is read by the browser to create the page layout of the website.
- **CSS:** Cascading Style Sheets. CSS provides the style information for the HTML document. CSS enables the separation of the HTML document content from the document's presentation and defines a consistent approach for specifying the style of an HTML document.
- **JS:** JavaScript. JS is an interpreted computer programming language used by browsers to provide dynamic, interactive web page content without requiring communication with the web server. JS is bundled alongside an HTML document.
- **jQuery and jQuery UI:** These are libraries (bundles) of JavaScript code to simplify the development of JavaScript.
- **PHP:** PHP Hypertext Preprocessor. PHP is a server-side computer programming language used to generate dynamic web pages prior to sending the web page on to the browser in the form of HTML. PHP may interact with databases and other external data sources during the creation of the web page.
- **SQL:** Structured Query Language. A high-level language for expressing what a user wants from a database.
- **MySQL.** The relational database management system that manages some data set represented by tables. MySQL also accepts SQL queries and will return data that matches what the SQL query requests.

This report discusses graph theory and network flow relating to the Ford-Fulkerson algorithm in significant depth. This is discussed in a separate section, section [2.2](#).

2.2 Graph Theory and Network Flow

Graphs are made up of two components: vertices and edges. Vertices can be thought of as cities, with edges being roads between cities. A road between two cities can have a maximum number of cars on it at one time. This is known as the edges capacity. The road has a number of cars on it at present time, known as the edges flow. The flow of traffic can never exceed the capacity of the road.

The road from city A to city B is separate from city B to city A and the existence of one road does not imply the existence of the other.

Network Flow is the study of working out how to have as much traffic move from a start city (the source) to an end city (the sink) by passing through intermediate cities. By using this concept, it is possible to determine the elimination of teams in a sports league. This application of graph theory and network flow will be discussed in detail in section 3.2 on the Ford-Fulkerson algorithm.

Chapter 3

Design and Implementation

3.1 Desktop User Interface

3.1.1 Design

In designing the desktop user interface the first step was to identify the main goals of our users, in order to make sure that the system delivered the required information quickly and clearly. After a discussion it was agreed that the typical user would have two main priorities - checking whether a specific team had been eliminated and viewing the league tables for each division. With this in mind several different paper prototypes were created by different members of the team so as to increase the range of ideas we could incorporate into the final design. These prototypes were compared in terms of functional clarity and aesthetic appeal, with the effective elements of each taken and incorporated into a final prototype design. (IMAGE OF PROTOTYPE HERE)

3.1.2 Implementation

3.2 Ford-Fulkerson Algorithm

3.2.1 Design

Wayne paper [\[3\]](#)

Kern paper [\[2\]](#)

Adler paper [\[1\]](#)

3.2.2 Implementation

3.3 Parser

3.3.1 Design

3.3.2 Implementation

3.4 Web Application

3.4.1 Design

Introduction

This section discusses the design of the web-based version of the application. The design of the application was constructed with only the most important functional and non-functional requirements in mind. The web-based version was classed as the least important part of the project, with preliminary effort directed solely towards the desktop application.

System architecture

The web application is a standard multi-tier architecture with the presentation, logic, and data separated from each other.

The presentation tier is the client/browser who has Hyper Text Mark-up Language (HTML) and Cascading Style Sheets (CSS) for the static presentation of content. In addition there is JavaScript supported by JQuery and JQueryUI for the dynamic user interface elements.

The logic tier runs on a web server called Lighttpd (pronounced lighty) that is supported by PHP: Hypertext Preprocessor (PHP). The logic tier has two data sources that make up the data tier, a MySQL database containing the latest data and a Java jar for looking back at older data.

The N-Tier Architecture diagram is available from appendix [D](#).

User interface

The user interface of the web application was intended on being as close to the desktop interface as viable within the constraints of a web browser and within the realms of what is a typical layout of a web page.

A wireframe for the web application is shown in appendix [C](#).

The web application has a single page containing the six available divisions. Each division is a table and only one is available for viewing at a time. The reasoning behind this is to keep as much information 'above the fold' (above the lower page boundary on a browser's window).

There are links at the top of each page that will allow the user to traverse the entire date range for the season allowing them to view the scoreboard and elimination status at any point in time.

3.4.2 Implementation

Introduction

This section discusses the implementation of the web-based version of the application. The implementation discussion will be split up into the three main sections as shown in appendix **D**.

Presentation - Client/Browser

The website uses three of the most common web technologies in use: HTML, CSS, and JavaScript (supported by jQuery and jQuery UI).

Logic - Web Server/PHP Processor

The website is dynamically created by using PHP. PHP interfaces with data one of the data layer sections and, from the results obtained from the data layer, produces the division tables and pushes the generated HTML along with the attached CSS and JavaScript.

Data - MySQL Database/Java JAR

The web application has two primary sources of data: a MySQL database and a Java JAR file. The MySQL database contains the recent information available on the sports divisions whereas the Java JAR contains a text file of the entire result set allowing the user to request to see the state of a division at any point in the league.

Chapter 4

Evaluation

4.1 Correctness Testing

4.2 Evaluation

4.2.1 Introduction

After completing the implementation of the desktop and web based applications a thorough user evaluation was carried out of both systems. The user evaluation functioned as a reliable way to test the usability and likeability of both system designs. All usability testing performed by the team was done in accordance with the University of Glasgow ethics procedures. The Evaluation was performed in a three stage process.

Participant Brief

Starting with an introductory briefing, the team member conducting the evaluation introduced the test participant to the system, provided the user with a test number and a copy of the user test script and described the aims of the user evaluation that was about to take place.

During this stage the participant was asked to answer a few simple questions to gauge their competency using desktop and web based applications, and to gauge their personal interest in the systems domain -sports statistics- .

During the introductory brief, it was made clear to the participant that no personal or identifying information would be collected from them during the user evaluation. It was decided by the team that this would hopefully decrease the amount of participants who would not complete the evaluation fully and to ease the process of gaining ethical permissions from the university . Due to the fact that no test participants decided to stop the evaluation half way through , and gaining ethical approval for the user evaluation was a simple process it was felt like this was a beneficial design .

In accordance with the university's ethical procedures during the introductory briefing the test participant were reminded of their right to stop the evaluation at any time with no requirement to give reason. The user was then further reminded that it was not them, but the system that was under evaluation, and the user was provided with the contact details of the team member conducting the evaluation. to allow the user to contact the team to answer any question or after thoughts that they had about the system or the user evaluation after it had been completed and they had some time to think on the process .

Think-Aloud (usability)

The evaluation was performed using the Think-Aloud technique which consists of participants performing a series of tasks set out to provide a complete overview of the full functionality provided by both the web based and desktop based Applications.

The test participants were encouraged to talk out loud as they performed a series of tasks, designed to provide a full overview of the complete functionality provided by both systems

At this stage the role of the team member conducting the evaluation was to take note of any hesitation shown by the test candidate, possible confusion, or errors encountered when using the system. The reactions shown by the test participant when interacting with both the web based and desktop based Applications would clearly highlight usability problems which have went unseen in the teams initial systems design.

Questionnaire (likeability)

4.1 Desktop Application

Evaluation Results

4.2 Web Application

Evaluation Results

Chapter 5

Conclusion

5.1 Summary

5.2 Future Work

5.3 Lessons Learned

Chapter 6

Contributions

6.1 Gordon Reid

- Team leader
- Ford-Fulkerson algorithm
- Second user interface iteration
- Second parser iteration (in use)
- Post-second user interface iteration file opening.
- Web application user interface
- Web application back-end

6.2 Ryan Wells

6.3 Kris Stewart

6.4 David Selkirk

6.5 James Gallagher

Appendix A

Installation and Running of Application

A.1 Desktop Application

A.1.1 Installation

The desktop application requires the Java Runtime Environment (JRE) available from <http://www.java.com/en/download/index.jsp>. The application has been tested on JRE 6 and JRE 7 without issue.

The print functionality requires a LaTeX distribution that includes the executable ‘pdflatex’. The installation procedure varies for each operating system and instructions are available from <http://latex-project.org/ftp.html>.

A.1.2 Running

After the Java Runtime Environment (JRE) is installed, running the application only requires double clicking the supplied JAR file.

The print functionality is executed within the Java application and thus is transparent to the user. In the event that the command ‘pdflatex’ cannot be found, the application will fail to print however will not crash. Print functionality is known to work on standard installations of the distribution on Linux/GNU-based and Mac OS operating system.

A.2 Web Application

A.2.1 Installation

Installation of the web application is not required as a remote host is running the required software. This can be accessed via <http://www.gordonrenfrewshire.com/teamw>. For purposes

of completeness and satisfying the potential desires of the reader, an installation procedure is supplied.

In the event that the supplied URL fails to work, please contact Gordon Reid via any of the following methods:

Student email: 1002536r@student.gla.ac.uk

Personal email: gordon.reid1992@hotmail.co.uk

Mobile phone: 07706 477 672

The web server has numerous standard applications running to service the web application. Each one is required for full functionality:

1. A web server (such as Lighttpd or Apache)
2. PHP (known to work on PHP 5.x)
3. Java Runtime Environment (version 6 or 7)
4. MySQL (version 5.x)

Installation of packages

The installation procedure assumes you have super user access on a Debian-based distribution. The official procedure for installation of a 'LAMP' (Linux Apache, MySQL, PHP) server is available from the Debian Wiki at <http://wiki.debian.org/LaMp>

Set up of database

COMMAND FOR DB CREATION HERE

SQL FOR CREATION OF TABLE HERE

In the folder 'website/content/php/includes/functions.php' there are a number of variables at the top of the page indicating the values for the server, user, password, and database. These can be modified to suit your requirements however the default are highly recommended.

A word of warning, the variable scope is very insecure and however was designed as such for simplicity of installation and testing. Please do not run the server code on a public or production server.

A.2.2 Running

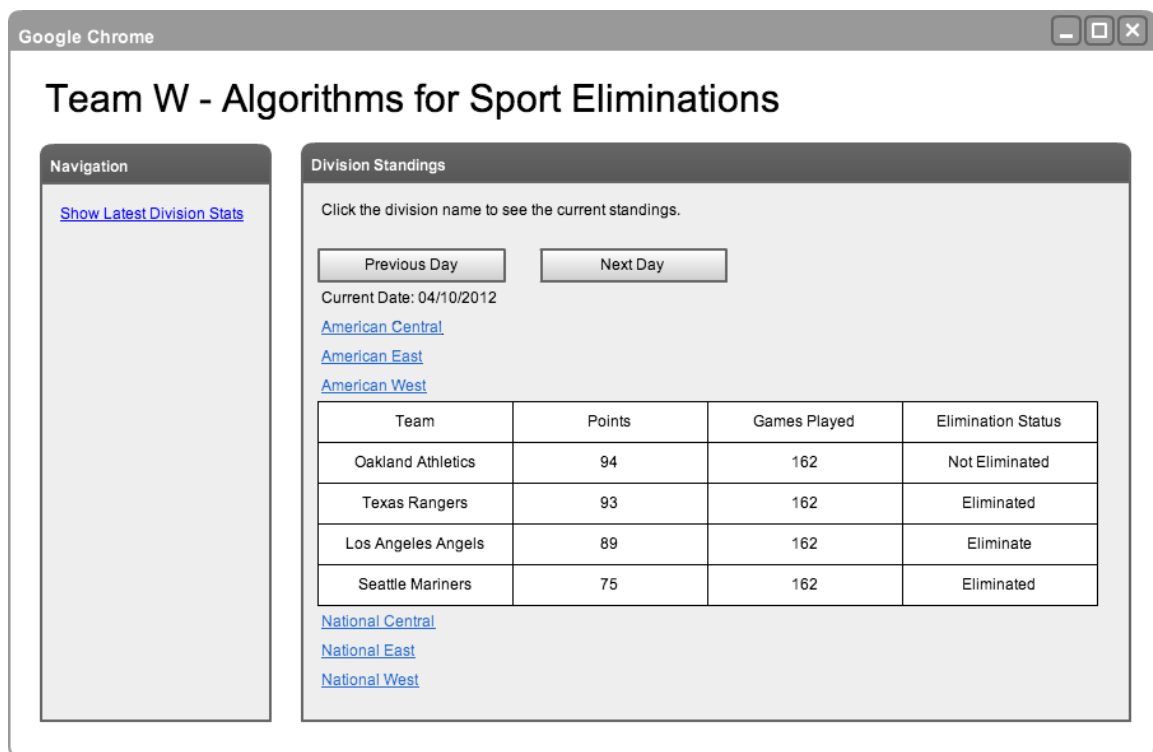
As stated in the installation section, the web application is available for viewing at <http://www.gordonrenfrewshire.com/teamw>. If a personal installation has been executed then running the application will be dependent on your own set up.

Appendix B

User Manual

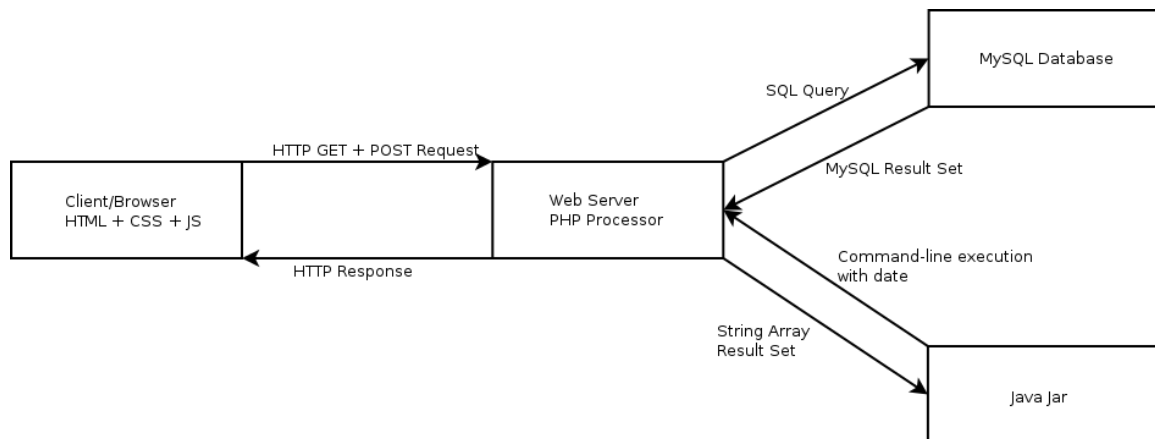
Appendix C

Web Application Wireframe



Appendix D

Web Application N-Tier Architecture Diagram



Bibliography

- [1] Ilan Adler, Alan L. Erera, Dorit S. Hochbaum, and Eli V. Olinick. Baseball, optimization, and the world wide web. 32(2):12–22, 2002.
- [2] Walter Kern and Daniel Paulusma. The new fifa rules are hard: complexity aspects of sports competitions. 108(2001):317–323, 2000.
- [3] Kevin D. Wayne. A new property and a faster algorithm for baseball elimination. 14(2):223–229, 2001.