Job searching engine based on Gale-Shapley stable matching algorithm

Piotr Staniów

Department of Computer Science
Faculty of Fundamental Problems of Technology
Politechnika Wrocławska

A thesis submitted for the degree of

inżynier

Supervisor: D.Phil. Mirosław Korzeniowski

Abstract

To

Contents

List of Figures

Chapter 1

Introduction

Information Technology labour market is quite distinct from others in terms of how people tend to seek for a job. Position requirements for a programmer are often very concise and precise, therefore it is feasible to quantify the skills and requirements an applicant has.

In recent years, a handful of job searching websites dedicated solely to programmers emerged on the internet[1] and presented a rather unique approach to filtering jobs. It is now preferable to put an emphasis on programming languages, technologies, libraries and frameworks which are involved in the post, rather than let users search by keywords and skim through a block of text within the offer.

1.1 Overview

In this particular case, especially at the beginning of recruitment process, a crucial part of matching an applicant and an employer is to ensure that some basic requirements are fulfilled. This gives way to providing a job search automation.

The both sides of the market can establish some matching based on applicant's and employer's requirements and offerings, however it cannot be guaranteed that participants do not change their mind. Informally, Gale-Shapley *stable matching algorithm* is the algorithm which outputs the most preferable pairs from two distinct sets, so that no other pair would rather be formed.

[1] Examples of this approach are: www.filltr.pl and www.nofluffjobs.com

1.2 Motivation

The need for automation of job searching process is motivated by a variety of problems it may solve. The first advantage of using the stable matching algorithm is that it forms pairs of employers and applicants such that there is neither employer nor applicant which would rather work with someone else and the one also prefers that matching.

One can easily imagine the situation in which an applicant is offered a job by *company A*, which at first is accepted, however in the near future *company B* may also offer a job to the applicant. As a result the candidate may decide to turn down the first proposal and incline towards the second one, which may increase the company A costs of recruitment process. Also the opposite situation is possible when it is the company which finds more suitable candidate for the post.

Furthermore, finding an IT job may turn out to be a daunting task, due to the fact that one has to skim through a vast number of announcements, yet rarely is the list represented and filtered only by a quantifiable set of skills and requirements. The approach presented in the following project will generate propositions for both companies and applicants without intense involvement. In turn, this will reduce the process of finding a suitable job to a mere procedure of filling in the profile.

Chapter 2

My First Chapter But Note The Numbering ...

2.1 First Paragraph

And now I begin my first chapter here ...

Here is an equation[1]:

$$CIF: \quad F_0^j(a) = \frac{1}{2\pi\iota} \oint_\gamma \frac{F_0^j(z)}{z-a} dz \tag{2.1}$$

2.2 Second Paragraph

and here I write more ...Knuth [1984]

2.2.1 sub first paragraph

... and some more ...

Now I would like to cite the following: Lamport [1986] and Knuth [1984] and Rudin [1973].

I would also like to include a picture ...

[1] the notation is explained in the nomenclature section :-)

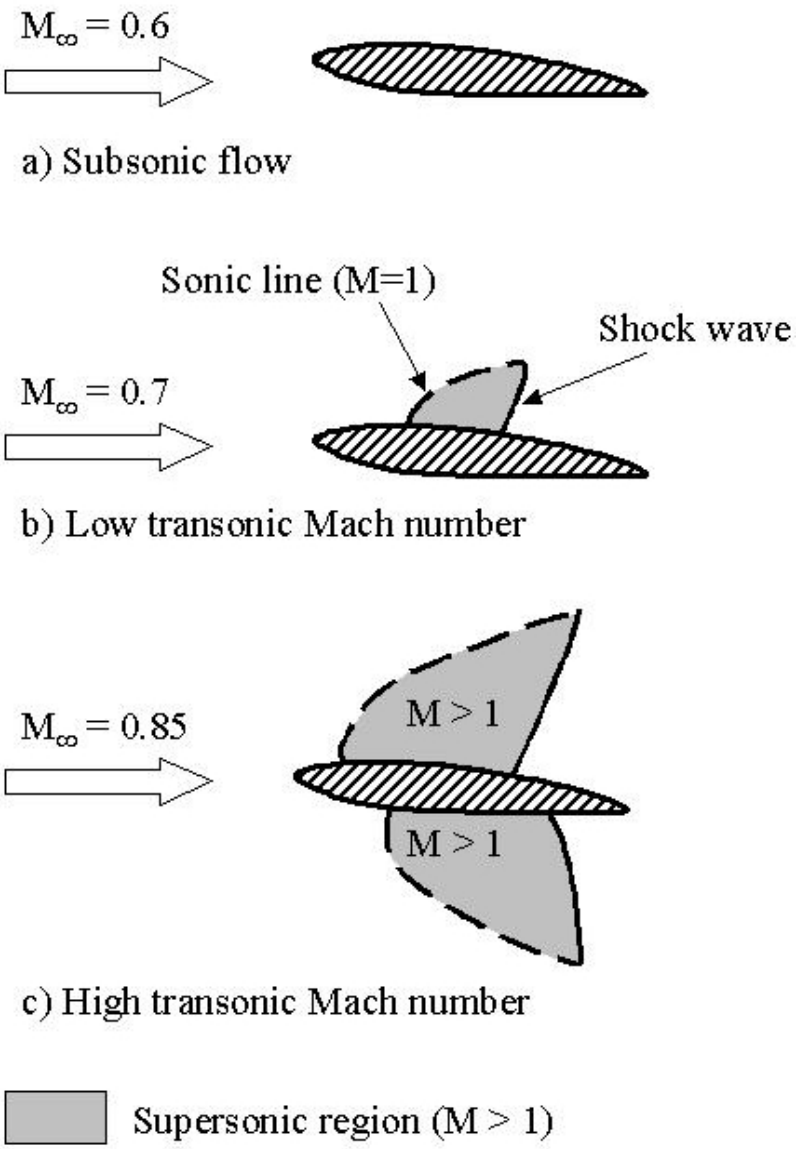

Figure 2.1: Airfoil Picture

So as we have now labelled it we can reference it, like so (2.1) and it is on Page 4. And as we can see, it is a very nice picture and we can talk about it all we want and when we are tired we can move on to the next chapter ...

I would also like to add an extra bookmark in acroread like so ...

Chapter 3

My Second Chapter

3.1 First Section

nd now I begin my second chapter here ...

3.2 Second Section

nd here I write more ...

3.2.1 first subsection in the Second Section

... and some more ...

3.2.2 second subsection in the Second Section

... and some more ...

3.2.3 third subsection in the Second Section

... and some more ...

Chapter 4

My Third Chapter

4.1 First Section of the Third Chapter

And now I begin my third chapter here ...

4.1.1 first subsection in the First Section

... and some more

4.1.2 second subsection in the First Section

... and some more ...

4.1.2.1 first subsub section in the second subsection

... and some more in the first subsub section otherwise it all looks the same doesn't it? well we can add some text to it ...

4.1.3 third subsection in the First Section

... and some more ...

4.1.3.1 first subsub section in the third subsection

... and some more in the first subsub section otherwise it all looks the same doesn't it? well we can add some text to it and some more and some more and some more and some more and some more and some more and some more ...

4.1.3.2 second subsub section in the third subsection

... and some more in the first subsub section otherwise it all looks the same doesn't it? well we can add some text to it ...

4.2 Second Section of the Third Chapter

and here I write more ...

Chapter 5

My Conclusions ...

Here I put my conclusions ...

Appdx A

and here I put a bit of postamble ...

Appdx B

and here I put some more postamble ...

References

Donald E. Knuth. *The TEXbook*. Addison-Wesley, 1984. 3

Leslie Lamport. *LATEX: A Document Preparation System*. Addison-Wesley, 1986. 3

W. Rudin. *Functional Analysis*. McGraw-Hill, New York, 1973. 3