Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Computer and Information Science

Master Thesis

Linear programming on Cell/BE

by

Åsmund Eldhuset

Supervisor: Dr.Ing. Lasse Natvig
Co-supervisor: Dr. Anne C. Elster

Trondheim, June 1, 2009

Abstract

(TODO)

Acknowledgements

(TODO)

Contents

List of Figures

List of Tables

Listings

List of Symbols and Abbreviations

Abbreviation	Description	Definition
LP	Linear programming	page 3

Chapter 1

Introduction

(TODO)

Chapter 2

Background

(TODO)

Chapter introduction

2.1 Linear programming

(Natvig)

Do we need section introductions too?

2.1.1 Problem formulation. Standard and slack forms

The term *linear programming* (LP) refers to a type of optimisation problems in which one seeks to maximise or minimise the value of a linear function of a set of variables that are constrained by a set of linear equations and/or inequalities[1].

Linear programming is a fairly general problem type, and many important problems (TODO) can be cast as LP problems — for instance, network flow problems and shortest path problems (see [**?**]).

(other than those problems that are initially formulated as an LP problem)

Throughout this report, we will consistently use n to refer to the number of variables and m to refer to the number of inequalities. The variables will typically be (TODO: spell "label(l)ed") x_1 through x_n.

The function to be optimised is called the *objective function*. In the real world situation that gives rise to an optimisation problem, the function may contain a constant term. However, since this term (TODO), we drop it from the objective function, which can then be written as $f = c_1x_1 + c_2x_2 + \ldots + c_nx_n = \sum_{j=1}^{n} c_jx_j$, where c_j are the coefficient values.

(TODO)

Nonnegativity of variables, which is often the case in real world prolems.

The equations and inequalities that (together with the objective function) constitute an LP problem may be represented in different forms. We shall first consider the *standard form*, in which only less-than-or-equal-to inequalities with all variables on the left hand side are allowed. (TODO) A problem containing an

Why are not less-than allowed?

[1]Hence, LP is not (as the name would seem to suggest) a programming technique.

Should I label the coefficients $a_{i1}, \ldots, a_{in}$ instead to maintain consistency with the standard/slack forms?

How to indent?

equalities of the form $a_1x_1+\ldots+a_nx_n = b$ (Natvig) may be rewritten by splitting each equality into two inequalities: $a_1x_1+\ldots+a_nx_n \leq b$ and $-a_1x_1-\ldots-a_nx_n \leq -b$. Also, the goal must be to maximise the objective function (if the original problem is to minimize f, we let our objective function be $-f$). A linear program in standard form can be expressed as follows: (TODO)

Maximise

$$f = \sum_{j=1}^{n} c_j x_j$$

with respect to

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad \text{for } i = 1, \ldots, m.$$

equation or equality?

The other common representation, which is employed by the simplex algorithm (to be presented shortly), is *slack form*, which only allows a set of equations (and a nonnegativity constraint for each variable). An inequality of the form $a_1x_1 + \ldots + a_nx_n \leq b$ is converted to an equation (TODO) by adding a *slack variable* w. Together with the condition that $w \geq 0$, the equation $a_1x_1 + \ldots + a_nx_n + w = b$ is equivalent to the original inequality (whose difference, or "slack", between the left and right hand sides is represented by w).

2.1.2 Simplex algorithm

2.1.3 Interior point algorithms

2.2 Cell Broadband Engine

2.2.1 Architecture

2.2.2 Programming methods

Chapter 3

Design

(TODO)

Chapter 4

Implementation and testing

(TODO)

Chapter 5

Evaluation

(TODO)

Chapter 6

Conclusion

(TODO)

Future work

Bibliography