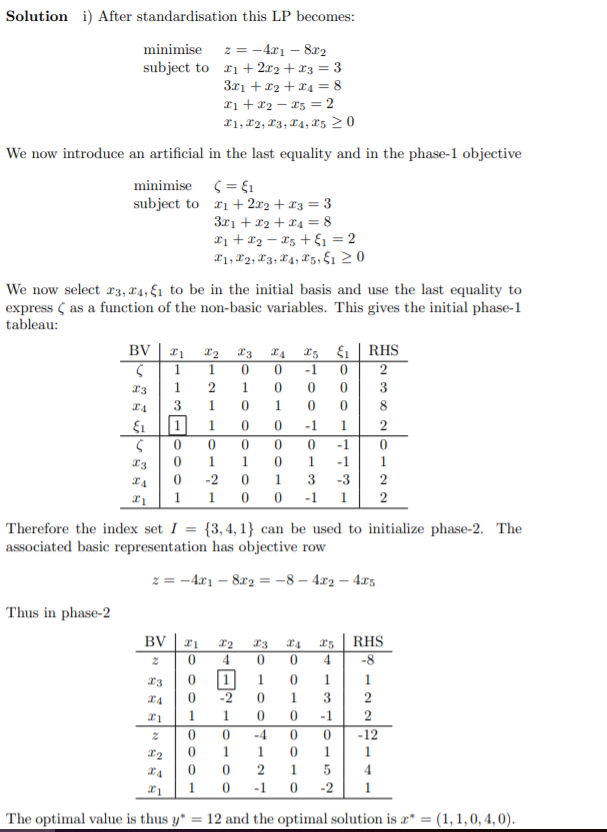
745=l==l 19 **hello darkness my old friend :(**

(a) From tutorial:§



(b)

We first formulate the constraints as equalities:

Since x1 is free, we substitute it using equality 3:

. Then we get

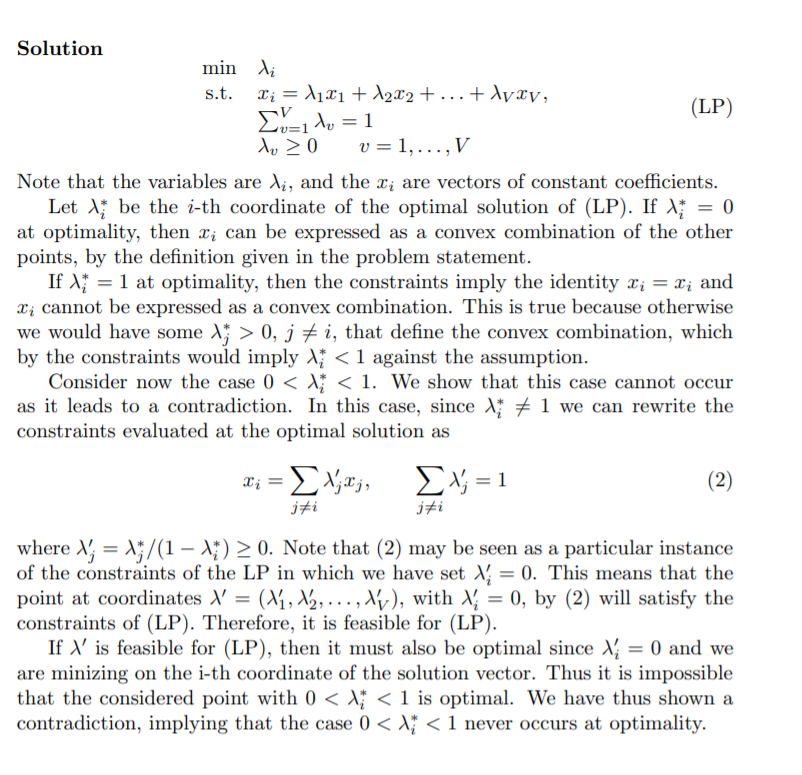
subject to

We now homogenise the equation by introducing new variables and . We set each , which results in

subject to

To normalise the denominator to unity, we force it to be equal to 1 by adding a constraint, giving us the final standardised linear program

subject to

(c) (Exam 2014-2015, Q1b)

2

(a) <Absolute value suspected out of syllabus>

i)

The MM problem is equivalent to which is equivalent to finding the minimum of an LP problem. This is the case since for any arbitrary sets X and Y and

ii)

i=1:

subject to

By subtracting the second constraint from the first we get and an objective value of 2.

i=2:

subject to

By subtracting the second constraint from the first we get and an objective value of 4.

Thus, the optimal value for (MM) is 2.

(b)

let x\_{ABi} be the number of barrels transported between A & B in month i

Let s\_i0, s\_j0, s\_i1, s\_i2, s\_j1, s\_j2 be the stored barrels

Objective:

min 7 Σ x\_{AB1} for all A, B

Meeting demand:

x\_{JF1} + x\_{IF1} = 10 // cannot store oil at F

x\_{JF2} + x\_{IF2} = 20

Production constraints:

x\_{AI1} <= 12

x\_{AI2} <= 9

x\_{BI1} <= 5

x\_{BI2} <= 4

Flow:

Given store\_I = (x\_{BI} - x\_{IF}) + (x\_{AI} - x\_{IJ})

Given store\_J = x\_{IJ} - x\_{JF}

store\_I + store\_J >= 7 // because we need to have 7 stored barrels to meet 20 at F in month 2

// NB although it is trivial that we don’t store at J ( we can max costs in second month ), the LP is supposed to figure this out for us.

// S\_I1 = x\_AI1 + x\_BI1 – x\_IJ1 – x\_IF1 + S\_I0

// S\_I2 = S\_i1 + x\_AI2 + x\_BI2 – x\_IJ2 – x\_IF2

// S\_J1 = x\_IJ1 – x\_JF1 + S\_J0

// S\_J2 = S\_J1 + X\_IJ2 – X\_JF2

//isn’t it like this? You have stored quantity in month 1, but also stored quantity in month 2. Ofc in this specific formulation it makes no sense to have barrels stored after month 2, but this allows for extension if you have to figure out month 3 etc.

x\_{AI} >= x\_{IJ} not true, x\_{IJ} can include barrels previously stored at I

x\_{IJ} >= x\_{JF}

x\_{BI} >= x{IF}

All x >= 0, s >=0, s\_i0 = s\_j0 = 0, and x int

//A simpler version

Min z = 21\*A\_out + 7\*B\_out – 7\*A\_store-7\*B\_store

s.t. A\_out + B\_out >= 17

A\_out B\_out – A\_stor + e – B\_store >= 10

A\_out <= 12 B\_out <= 5

All vars >= 0

//Also simpler and includes month 2

-We can simplify the 7 out of the objective

-We can ignore the storage in month 2, essentially making it part of the slack

-We don’t need to model how much is stored in total, just how much of the production from month 1 is being sent to I/J to be used in month 2.

-Instead of modelling every individual transfer as a fraction of production or storage, variables can represent entire journeys like AF if they have the appropriate weight in the objective

Min z = 3AF + 2BF + BI + AI + 2AJ

AI + AJ + AF <= 12 as mentioned above, we model AJ even though it should be 0

BF + BI <= 5

AF + BF = 10

AF2 <= 9

BF2 <= 4

AF2 + BF2 + BI + AI + AJ = 20

All variables >= 0

(c) See Case Study 3.

The Klee-Minty cube illustrates the exponential time nature of the simplex algorithm. It shows that, at worst case, an LP with n = 3 will require 2^3 = 8 iterations (i.e. visit all vertices of the cube to find the optimal).

3

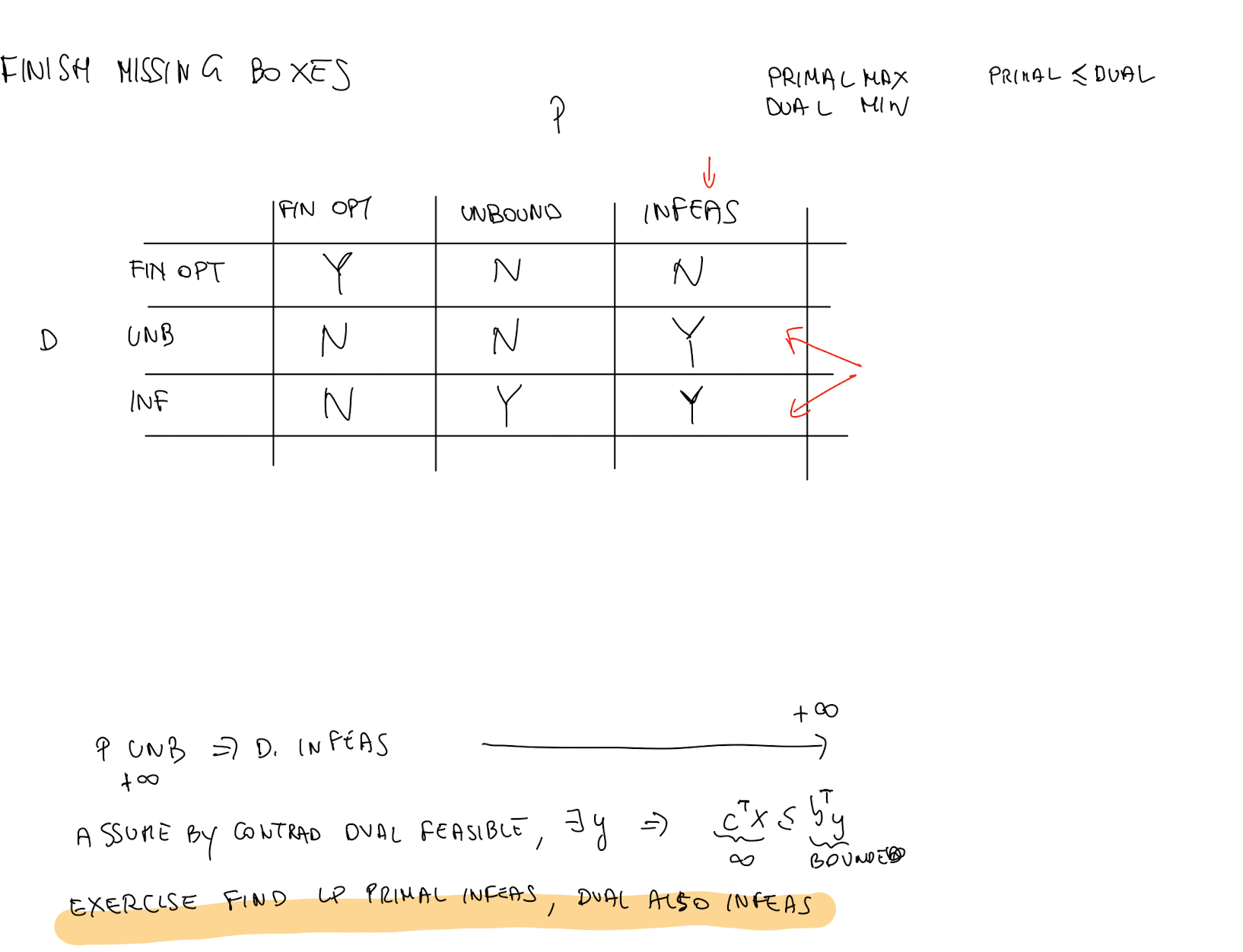
(a)

O X X

X O O

X O X

A)



FF

(P)

max x\_1 + 2x\_2

x\_1 + x\_2 >= 1

x\_1 + x\_2 <= 1

x\_1, x\_2 >= 0

(D)

min y\_1 + y\_2

y\_1 + y\_2 >= 1

y\_1 + y\_2 >= 2 (yes, inelegant, but should make the point (?))

y\_1 <= 0, y\_2 >= 0

IU

(P)

max 2x\_1 + x\_2

x\_1 + x\_2 >= 2

x\_1 + x\_2 <= 1

x\_1 >= 0, x\_2 >= 0

(D)

min 2y\_1 + y\_2

y\_1 + y\_2 >= 2

y\_1 + y\_2 >= 1

y\_1 <= 0, y\_2 >= 0

UI

opposite of IU

II

(P)

max x\_1 + 2x\_2

x\_1 + x\_2 >= 2

x\_1 + x\_2 <= 1

x\_1 <= 0, x\_2 >= 0

(D)

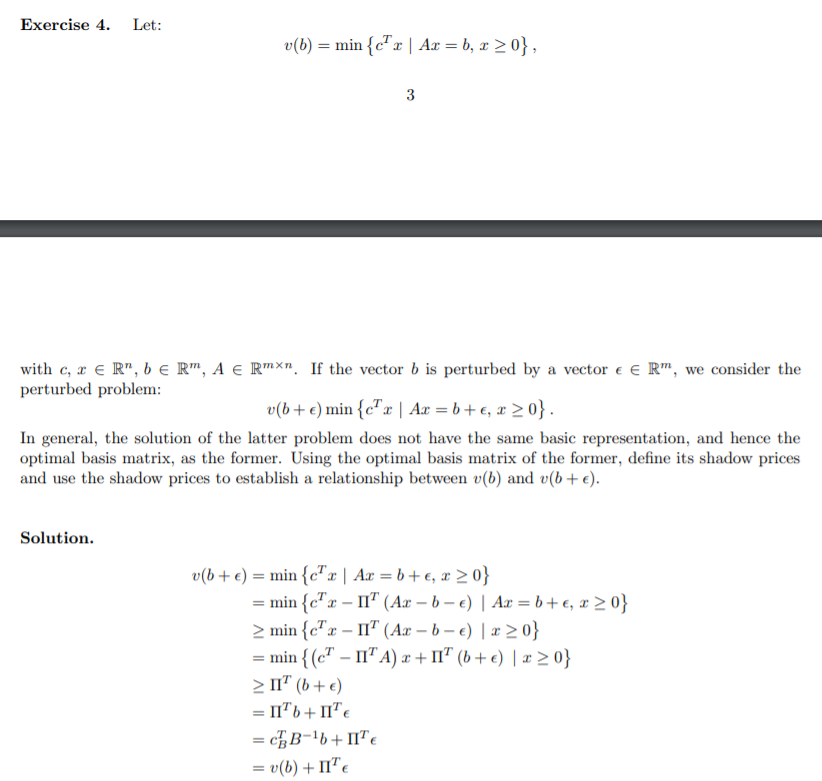
min 2y\_1 + y\_2

y\_1 + y\_2 <= 1

y\_1 + y\_2 >= 2

y\_1 <= 0, y\_2 >= 0

3 (b) (i) see Tutorial, v(2b) >= 2v(b)



(ii) B^{-1} (2b) >= 0 (See slides)

4

(a)

See tutorial for a brief intro

d\_1 + d\_2 + d\_3 = 1 (for each of the three regions, if d\_i == 1 then it’s on)

// region 1

x\_1 >= 1

x\_1 <= 2 + M(d\_2 + d\_3)

-M(d\_2 + d\_3) <= x\_2

x\_2 <= 2x\_1 - 1.5 + M(d\_2 + d\_3)

// region 2

x\_1 >= 2 - M(d\_1 + d\_3)

x\_1 <= 3 + M(d\_1 + d\_3)

-M(d\_1 + d\_3) + 2.5 <= x\_2

x\_2 <= 3 + M(d\_1 + d\_3)

// region 3

x\_1 >= 3 - M(d\_1 + d\_2)

x\_1 <= 4

-M(d\_1 + d\_2) <= x\_2

x\_2 <= 0.5x\_1 + 1 + M(d\_1 + d\_2)

M has to be sufficiently large to allow disablement of the inequalities, e.g. 100.

(b)

2/3 x2 + 1/3 x4 - x5 + \xi\_1 = 1/3

does not include - basically cuts off optimal of -20/3, non int (Proof is in notes)

Alternative (unchecked):

2/3 x\_2 + 1/3 x\_4 >= 1/3 (for both cutting on x\_1 and x\_3)

=> 2x\_2 + x\_4 >= 1

Substitute so in terms of x\_1, x\_2: [x\_4 = 10 – 3x\_1 – 2x\_2]

(GC) : x\_1 <= 3

New LP same as last with additional constraint of (GC)

(c)

1，5

4，5

3，5

2，5

4， 3， 2

1， 4， 3

2， 4， 1

1， 3， 2