Technical University of Crete
School of Electrical and Computer Engineering
Course: Convex Optimization
Exercise 4
Instructor: Athanasios P. Liavas
Georgios Klioumis 2017030116

December 16, 2021

1

Στην αρχή έγινε η κατασκευή των πινάκων A και b με τις προδιαγραφές της εκφώνησης, ώστε να καταλήξουμε με ένα feasible πρόβλημα της μορφής:

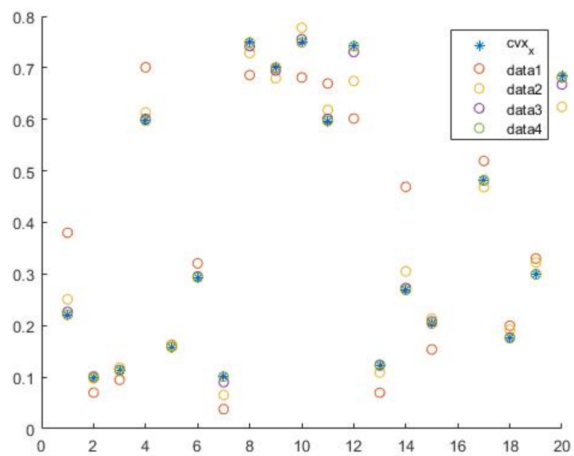
$$min\ f(x) = -\sum_{i=1}^{n} log(x_i)$$
$$st\ Ax = b$$

2

Το πρόβλημα λύθηκε με τον cvx χωρίς κάποια δυσκολία

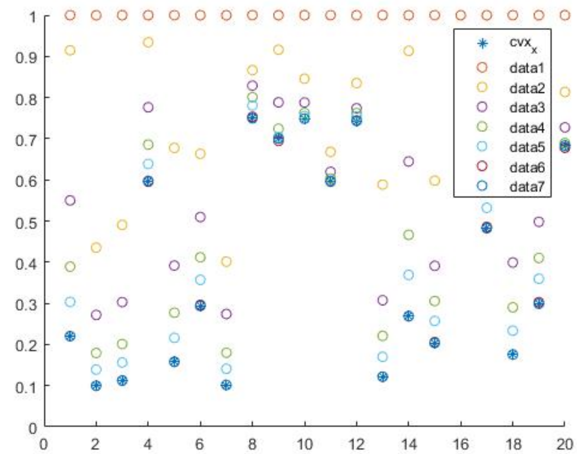
3

Αρχικά χρειάστηκε να βρούμε ένα feasible σημείο. Αυτό επετεύχθη με την χρήση του cvx χωρίς κάποιο objective function. Έτσι το διάνυσμα x που επέστρεφε το cvx δόθηκε σαν όρισμα στην συνάρτηση newton_affine_constraint που υλοποιεί τον γενικευμένο αλγόριθμο Newton. Για την υλοποίηση του αλγορίθμου Newton ακολουθήθηκε ο ψευδοκώδικας των σημειώσεων του μαθήματος. Για τον υπολογισμό του βήματος Newton, έγινε χρήση της αναλυτικής έκφρασης που βρίσκεται στις σημειώσεις. Αφού έτρεξε ο αλγόριθμος, συνέκλινε σε 4 επαναλήψεις για πρόβλημα με $p = 10$ και $n = 20$ στην τιμή που υπολόγισε ο cvx στο ερώτημα 1. Το progression του αλγορίθμου φαίνεται στο παρακάτω διάγραμμα που δείχνει το optimal point (που υπολόγισε ο cvx) και τα x που παράγει ο αλγόριθμος μέχρι να φτάσει σε αυτό.



4

Η κατασκευή του αλγορίθμου έγινε πάλι με χρήση του ψευδοκώδικα των σημειώσεων. Για την επίλυση του συστήματος εξισώσεων που μας δίνουν το βήμα του αλγορίθμου primal-dual έγινε χρήση της συνάρτησης solve, αφού κατασκευάστηκαν κατάλληλα οι γραμμικές εξισώσεις. Το αποτέλεσμα της επίλυσης του συστήματος χωρίστηκε στο διάνυσμα Δx_{pd} και w (το w δίνει το διάνυσμα Δv_{pd}). Αξίζει να σημειωθεί ότι ο αλγόριθμος ξεκινάει από το $x = 1$ και $v = 1$. Αφού έτρεξε ο αλγόριθμος, συνέκλινε σε 7 επαναλήψεις για πρόβλημα με $p = 10$ και $n = 20$ στην τιμή που υπολόγισε ο cvx στο ερώτημα 1. Αξίζει να σημειωθεί ότι προστέθηκε στον αλγόριθμο μία γραμμή που επιστρέφει την λέξη 'feasible!!!' στην κονσόλα όταν εκτελέσουμε επανάληψη με $t = 1$, σε αυτό το σημείο είναι δυνατόν να γυρίσουμε στον γενικευμένο Newton για το πρόβλημα μιας και πια είμαστε feasible.
Το progression του αλγορίθμου φαίνεται στο παρακάτω διάγραμμα που δείχνει το optimal point (που υπολόγισε ο cvx) και τα x που παράγει ο αλγόριθμος μέχρι να φτάσει σε αυτό.



5

Το dual του γενικού προβλήματος

$$min\ f(x)$$
$$st\ Ax = b$$

είναι το

$$max\ g(v) = -b^t v - f^*(-A^T v)$$

όπου f^* είναι η conjugate σνάρτηση της f. Στην δική μας περίπτωση
$f^*(x) = -n - \sum_{i=1}^{n} log(v_i)$
Συνεπώς το πρόβλημα είναι το

$$max\ g(v) = -b^t v + \sum_{i=1}^{n} log(v_i) + n$$

Από τις συνθήκες βελτιστότητας έχουμε ότι:

$$\boldsymbol{\nabla} f(x_*) + A^T v_* = -1/x_{*i} + A^T v = 0 \Leftrightarrow x_{*i}(v_*) = 1/(A^T v)$$

Με χρήση του cvx και την εντολής maximize (αυτή την φορά) βρίσκουμε τα v_* που μεγιστοποιούν την $g(v)$. Έτσι μπορούμε να βρούμε τα x_* που ελαχιστοποιούν το primal πρόβλημα από την έκφραση

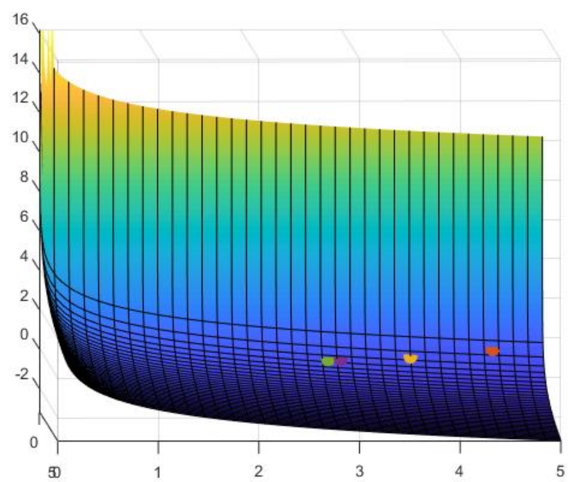
$$x_{*i} = 1/(A^T v_*)$$

και παρατηρούμε ότι βρίσκουμε την ίδια βέλτιστη τιμή για τα ίδια x με τους προηγούμενους αλγορίθμους.

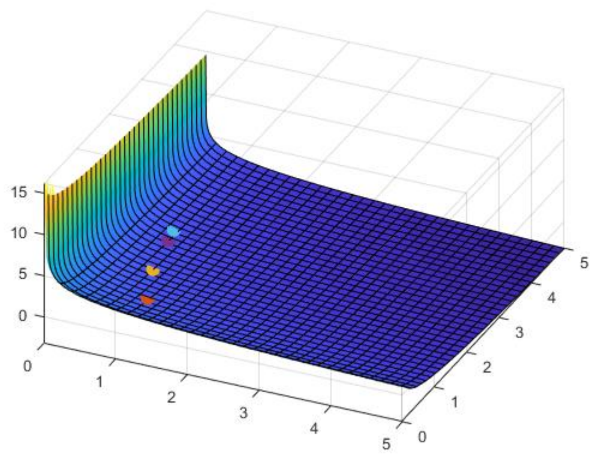
6

Για την αναπαράσταση του convergence των αλγορίθμων έγινε χρήση του trick με το pause ώστε στα διαγράμματα να αναπαρίστονται τα x που παράγουν οι αλγόριθμοι σιγά σιγά, ώστε να φανεί ο τρόπος με τον οποίο συγκλίνουν στην λύση.

- Newton
  Παρατηρούμε ότι τα σημεία που παράγει ο αλγόριθμος βρίσκονται όλα πάνω στον αφφινικό περιορισμό, στην περίπτωσή μας πάνω σε μία γραμμή, και είναι όλα feasible.



- Primal-Dual
  Παρατηρούμε ότι τα σημεία δεν βρίσκονται πάνω σε κάποιον αφφινικό περιορισμό. Αυτό φαίνεται από το γεγονός ότι τα σημεία είναι διάσπαρτα στον χώρο. Γενικά τα σημεία δεν είναι όλα feasible, παρά μόνο κάποια στο τέλος (σε κάποιες περιπτώσεις εκτέλεσης μόνο το τελευταίο).



Για να γίνει κατανοητό το πόσο γρήγορα συγκλίνουν οι δύο αλγόριθμοι που κατασκευάσαμε, κατασκευάστηκε το παρακάτω διάγραμμα που δείχνει την απόλυτη διαφορά της βέλτιστης τιμής της συνάρτησης από τα διάφορα $f(x_k)$ που παράγουν οι δύο αλγόριθμοι. Αξίζει να σημειωθεί ότι για προβλήματα διαστάσεων $p = 1$ και $n = 2$, ήταν αμφιλεγόμενο το ποιος αλγόριθμος συγκλίνει πιο γρήγορα, ενώ για προβλήματα μεγάλων διαστάσεων, η σύγκλιση του αλγορίθμου Newton ήταν γρηγορότερη. Παρακάτω φαίνονται οι εκδοχές του διαγράμματος για προβλήματα 2 set διαστάσεων, ενός μικρών διαστάσεων και ενός μεγάλων.

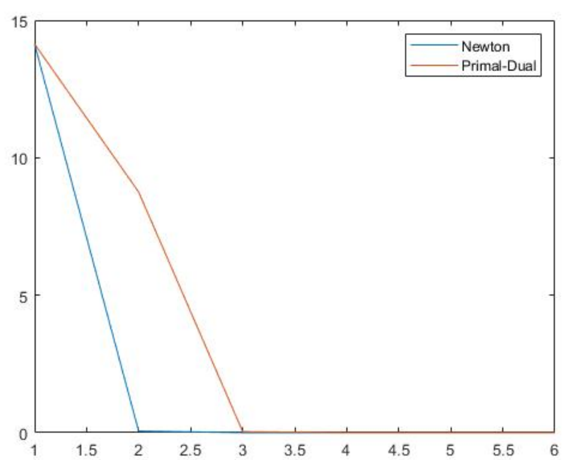


Figure 1: p=10 n=20

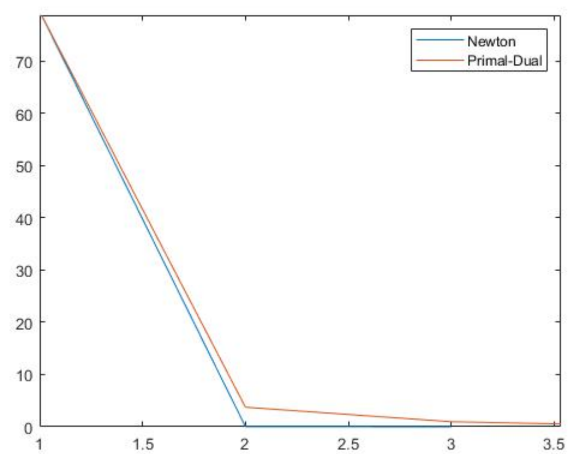


Figure 2: p=50 n=100