3-2Discrete Optimization

Hao Li

12/01/2019

#Note

This R repository is for demonstration of algorithms involved in the book Mathematical Modeling (4th Edition) written by Prof. Mark. M. Meerschaert

```
#Exp 3.2
#Discrete Optimization Problem

#R script to solve Exp 3.2 with NO SIMPLIFICATION ON THE ORIGIONAL MAP
#This uses some R conventions with origin on the top left corner of the matrix
#
#Define map matrix

gmap = cbind(c(3,2,5,8,10,0),
             c(0,1,3,5,6,2),
             c(1,1,3,2,3,3),
             c(4,2,0,1,1,1),
             c(2,3,1,0,3,1),
             c(1,2,2,0,1,1))
#gmap = edit(gmap)#Uncomment this to edit the 'map' in an e-spreadsheet

require(plot3D)
```

```
## Loading required package: plot3D
```

```
x = seq(from = .5,to=5.5,by=1);y = seq(from = .5,to = 5.5,by=1)
gbase = mesh(x,y)
#gbase$x
#gbase$y

#Define radius function r
r = function(pos,gbase) sqrt((pos[1]-gbase$x)^2 + (pos[2]-gbase$y)^2)
#This uses  vectorizations for many times, returns a radius matrix for
#position marked at pos on geographical base

ztime= function(pos,gmap,gbase) 3.2 + 1.7* sum(gmap* r(pos,gbase)^.91)/84

library(doParallel)
```

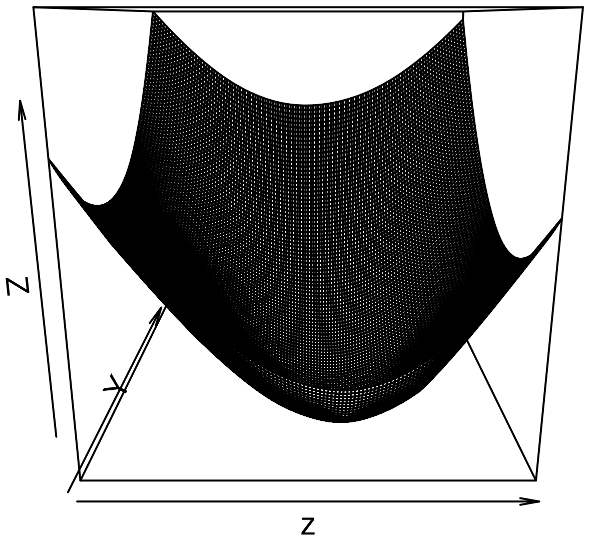
```
## Loading required package: foreach
## Loading required package: iterators
```

```
## Loading required package: parallel
```

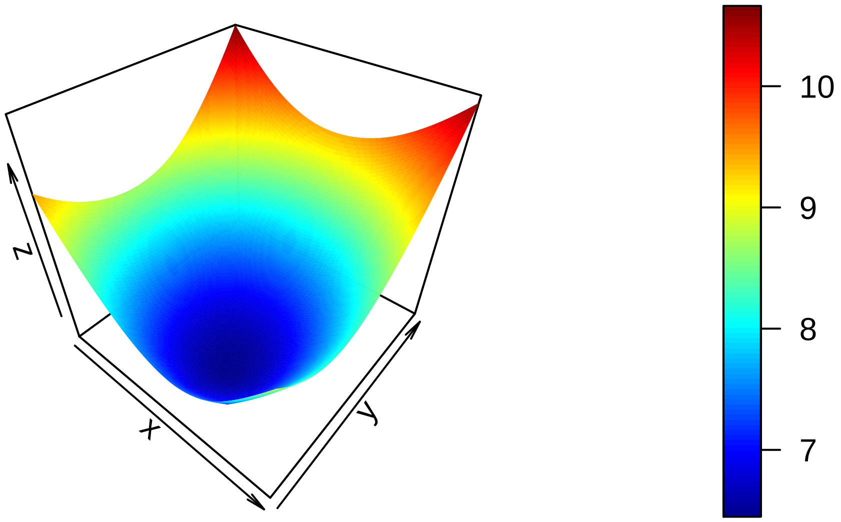
```r
#registerDoParallel(8)#Uncomment this to activate parallel computing

pix = .05
xbase = seq(from = 0, to =6, by = pix);ybase = xbase

z = foreach(i = seq_along(ybase),.combine = cbind) %dopar%{
  v= numeric(length(xbase))
  for(j in seq_along(xbase)) v[j]<-ztime(pos =c(xbase[j],ybase[i]),gmap,gbase)
  v
}
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

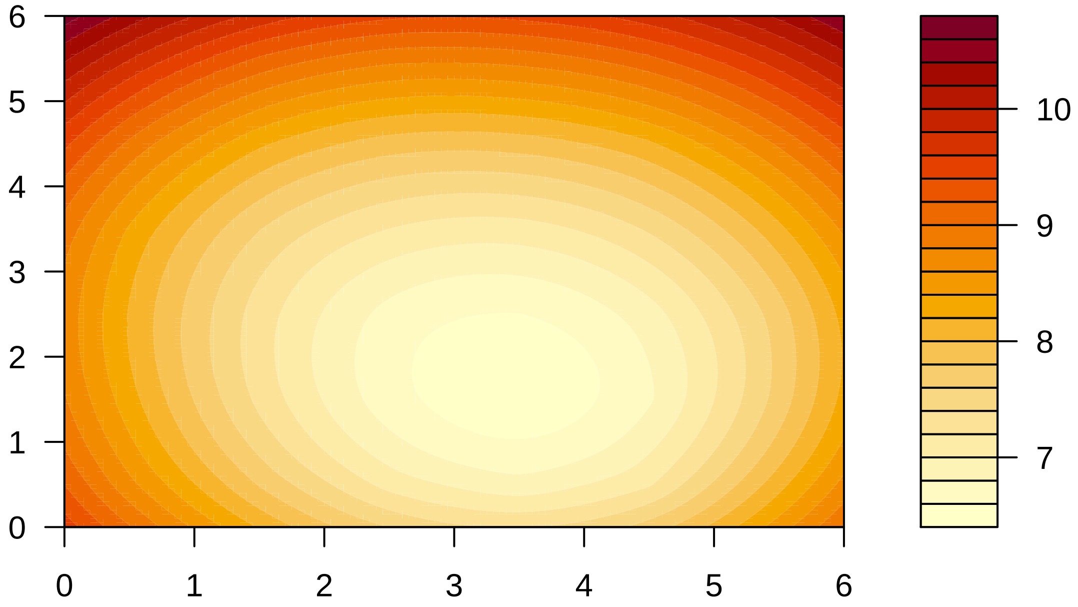
```r
persp(z)
```



```r
library(plot3D)
persp3D(xbase,ybase,z)
```



```
filled.contour(xbase,ybase,z)
```



```
contour(xbase,ybase,z)

#which.min(z)/length(xbase)
yi =as.integer(which.min(z)/length(xbase))
(x_min_trans = ybase[yi])
```

```
## [1] 1.45
```

```
xi = which.min(z)- yi*length(xbase)
(y_min_trans = 6-xbase[xi])
```

```
## [1] 2.5
```

```
(z_min_trans = min(z))
```

```
## [1] 6.447936
```

```
#Alternatively...
#Random search test
re=foreach(i = 1:8,.combine = cbind) %dopar% {
  R=Inf
  for(j in 1:125){
    pos=runif(2,0,6)
    Rnew =ztime(pos,gmap,gbase)
    if(Rnew<R){
      p = pos
      R = Rnew
    }
  }
  rbind(p[1],p[2],R)
}
result=re[,which.min(re[3,])]


(x_min_trans1 = result[2])
```

```
##
## 1.62866
```

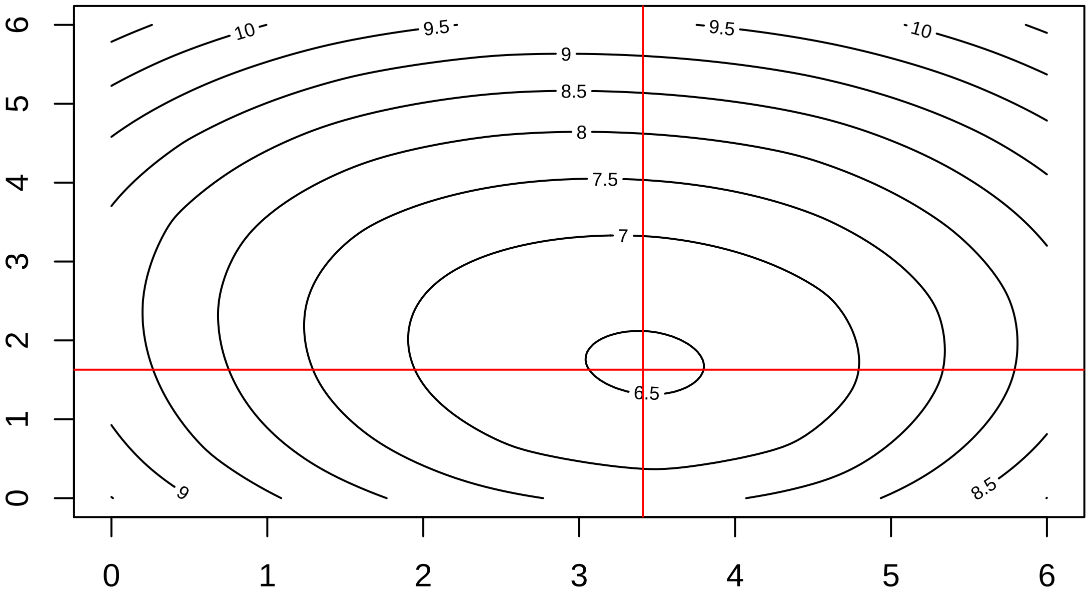
```
(y_min_trans1 = 6-result[1])
```

```
##
## 2.590985
```

```
(z_min_trans1 = result[3])
```

```
##        R
## 6.450983
```

```
abline(h = x_min_trans1,col= "red")
abline(v = result[1],col = "red")
```


10
9.5
9
8.5
8
7.5
7
6.5
9
8.5
9.5
10
0
1
2
3
4
5
6
0
1
2
3
4
5
6