Tarefa 3

Gabriel Belém Barbosa RA: 234672

31 de Agosto de 2021

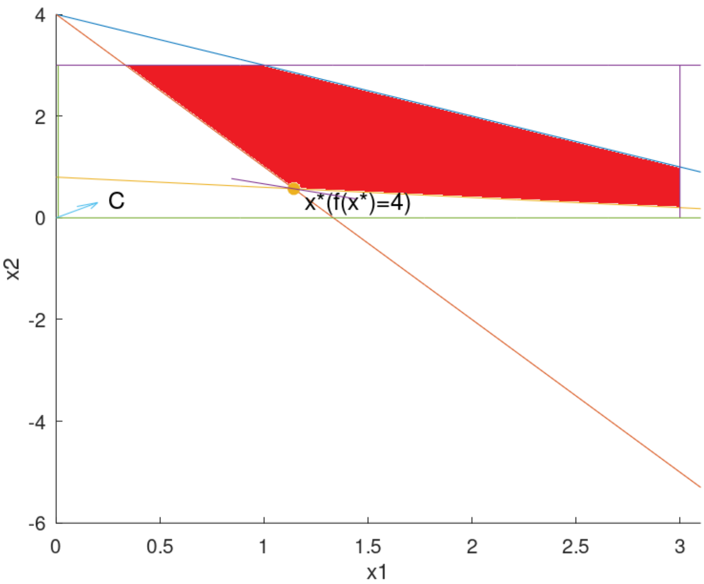
Conteúdo

1 Desenvolvimento

Sabendo que se um problema de otimização linear tem uma solução ótima, então existe um vértice ótimo (teorema mostrado em aula), o algoritmo testa todos os vértices (intersecção de pares de limitantes de restrições) aplicando a função objetivo (que se quer minimizar ou maximizar) e, se o valor do vértice atual é maior/menor que o anterior máximo/mínimo, o algoritmo checa se o vértice em questão obedece todas as restrições, e, em caso afirmativo, ele se torna o novo máximo/mínimo. Nas figuras abaixo, a região de factibildiade está colorida em vermelho, a solução ótima está denotada, caso exista, assim como o valor da função objetivo na solução. O vetor c é o vetor gradiente da função, e a curva de nível está posicionada sobre o ponto solução ótima (em caso de segmento solução, a curva de nível estaria sobreposta ao segmento solução), quando existe solução ótima.

2 Item (a)

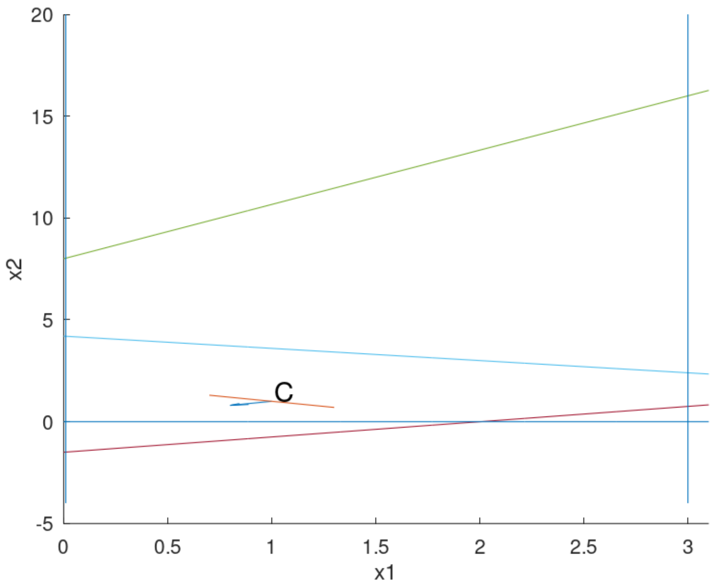
Figura 1: Item (a)



Solução única é $x* = \frac{1}{7}(8,4)$.

3 Item (b)

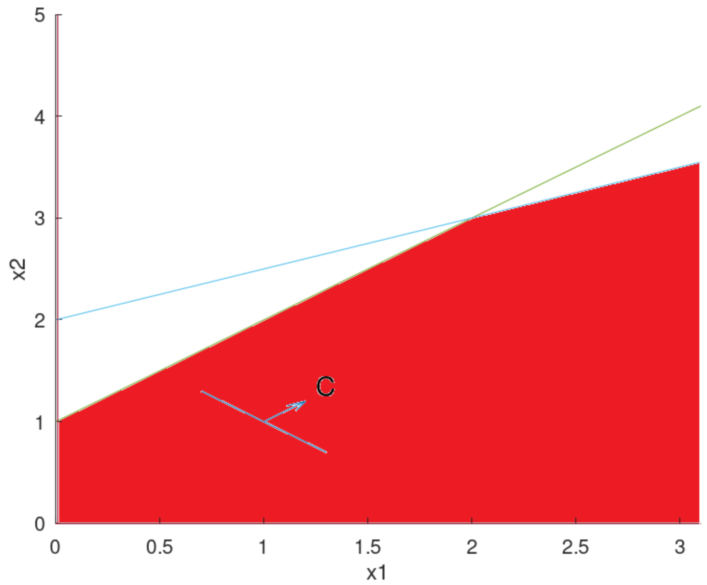
Figura 2: Item (b)



Região de factibilidade vazia: problema sem solução. Curva de nível posicionada junta ao vetor C.

4 Item (c)

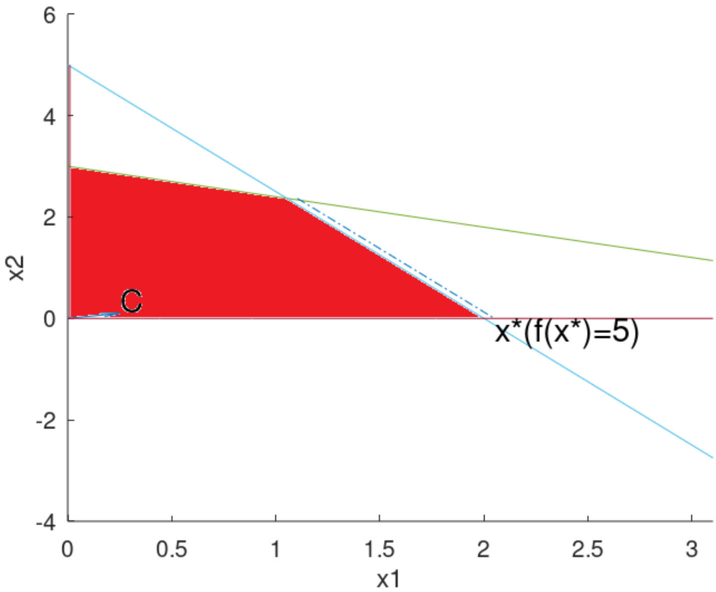
Figura 3: Item (c)



Região de factibilidade ilimitada na direção de interesse ($c^t x \rightarrow \infty$, e busca-se o máximo). Não existe solução ótima finita. Curva de nível posicionada junta ao vetor C.

5 Item (d)

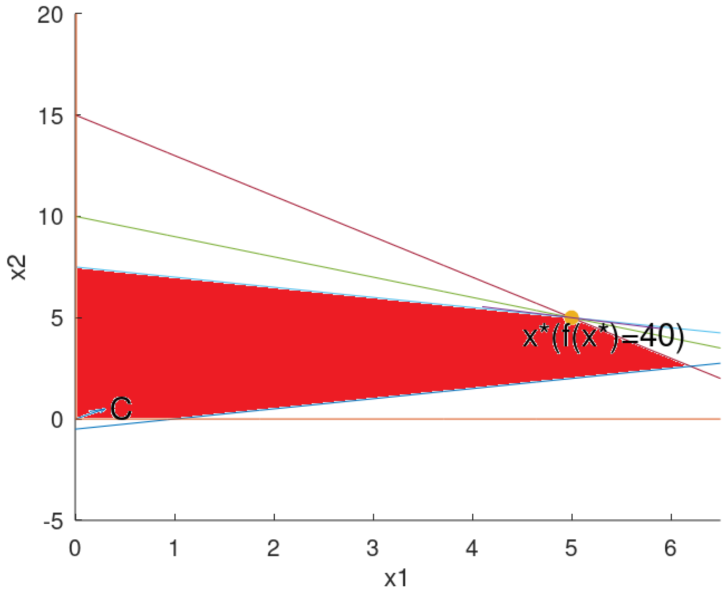
Figura 4: Item (d)



Segmento de reta pontilhado é a solução (infinitas soluções, conjunto limitado de soluções): $x* = (1-\lambda)(2,0) + \frac{\lambda}{19}(20,45)$, $\lambda \in [0,1]$.

6 Item (e)

Figura 5: Item (e)



Solução única é $x* = (5,5)$. Vale ressaltar que o ponto é degenerado, sendo intersecção de $x_1 + x_2 = 10$, $x_1 + 2x_2 = 15$ e $2x_1 + x_2 = 15$.

7 Algoritmo

A seguir está o algoritmo (em `Octave`) usado no Item (e).

```
f=@(x,y) 3*x+5*y;
grad=[3, 5];
agrad=[1,-grad(1)/grad(2)];
restri=[1,1;1,2;2,1;1,-2;1,0;0,1]
res=[10;15;15;1;0;0]
sines=[-1,-1,-1,-1,1,1]
corners=[]
a=[]
min=[0;0;-10000]; #valor inicial dummy
hf=figure;
xlabel ("x1");
ylabel ("x2");
```

```
hold on
quiver(0,0,grad(1)/10,grad(2)/10);
for i=1:rows(restri)
    for j=i+1:rows(restri)
        a=[restri(i,:);restri(j,:)]\[res(i);res(j)];
        corners=[corners;transpose(a)];
        if (f(a(1),a(2))>=min(3))
            key=1;
            for p=1:rows(restri)
                if ((restri(p,:)*a>res(p) && sines(p)==-1) ||
                (restri(p,:)*a<res(p) && sines(p)==1))
                    key=0;
                    break;
                end
            endfor
            if key
                min=[a;f(a(1),a(2))];
            end
        end
    endfor
    if (restri(i,2)!=0)
        fplot(@(x) (res(i)-restri(i,1)*x)/restri(i,2),[0,6.5]);
    else
        [restri(i,1),restri(i,2),res(i)]
        if (res(i)==0)
            line([0.01,0.01],[0,20]);
#altura escolhida à mão de acordo com o gráfico
        else
            line([res(i),res(i)],[0,20]);
#altura escolhida à mão de acordo com o gráfico
        end
    end
endfor
text(0.35,0.55,'C','FontSize', 14);
scatter(min(1),min(2), "filled");
text(min(1)-0.5,min(2)-1,strcat('x*(f(x*)=',
mat2str(min(3)),')'),'FontSize', 14);
line([min(1)-0.9*agrad(1),min(1)+0.9*agrad(1)],
[min(2)-0.9*agrad(2),min(2)+0.9*agrad(2)]);
legend("hide");
hold off
```