

# Project Operational Research

Gregory Filipe Lira da Silva  
Lincoln Wallace Valentim da Costa Silva

April 2020

## 1 Introdução

O projeto final referente a conclusão da disciplina pesquisa operacional, foi dado como um branch and bound para resolução de problemas lineares com restrição binária. Foi escolhido pela dupla para a criação do tal o solver da cvxpy.

## 2 Desenvolvimento

O cvxpy foi escolhido por facilitar a introdução de novas restrições para cada filho da árvore a qual usamos a biblioteca heapq, e também por aceitar a deepcopy da biblioteca copy.

Um dos grandes problemas enfrentados pela dupla foi criar uma heurística para verificar qual estava mais perto do 0.5, e a verificação se era fracionário ou não, visto que as variáveis do cvxpy são vista como numpy.float64.

Para a heurística foi utilizado uma lista com o intervalo entre os números e 0.5, para verificar qual estava mais próximo de 0.5 foi utilizado um if pra ver qual era o menor intervalo.

Na verificação se era fracionário foi utilizado a round da biblioteca numpy, assim resolvendo o problema dos flutuantes muito grandes.

## 3 Problemas e Árvores

O problema inicial que foi testado foi o que foi passado no projeto como teste (Figure 1 - problema 1), gerando a árvore (Figure 2 - arvore1).

foi criado pela dupla 3 novos problemas:

(Figure 3 - problema 2) com (Figure 4 - arvore 2)

(Figure 5 - problema 3) com (Figure 6 - arvore 3)

(figure 7 - problema 4) com (figure 8 - arvore 4)

## 4 Figures

3	2
5	10 8
3	5 2 6
4	4 4 7

Figure 1: problema 1

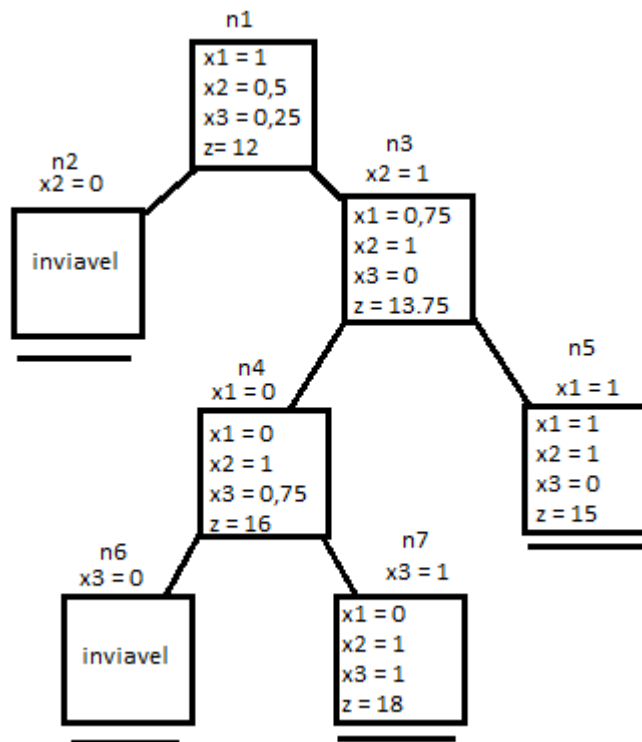


Figure 2: árvore 1

4	3
50	60 130 30
5	5 1 4 5
9	5 17 20 9
10	10 12 18 1

Figure 3: problema 2

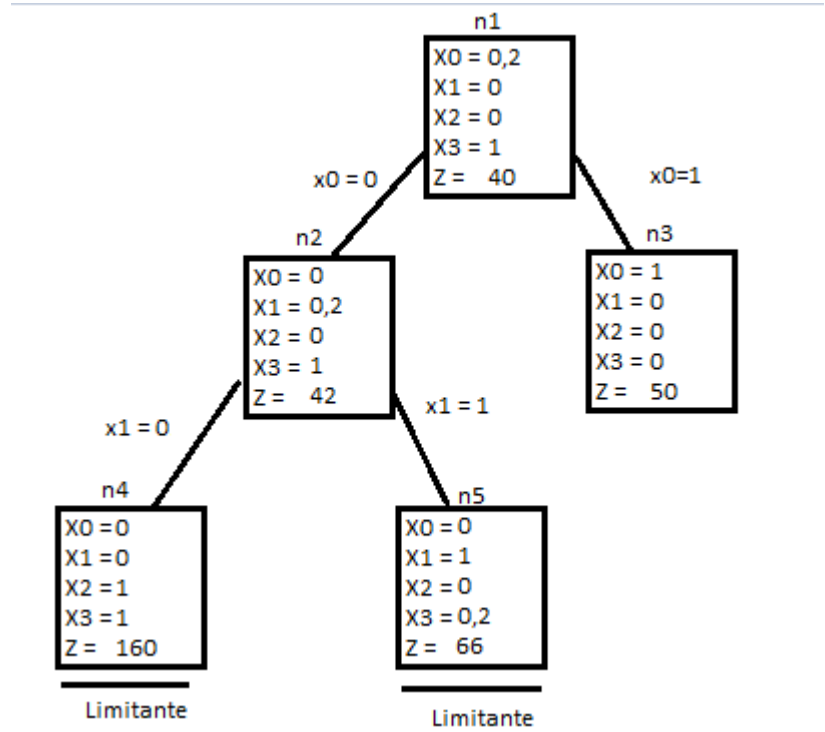


Figure 4: arvore 2

```

5 2
140 180 120 160 110
9 7 7 1 2 15
17 12 1 5 1 19
  
```

Figure 5: problema 3

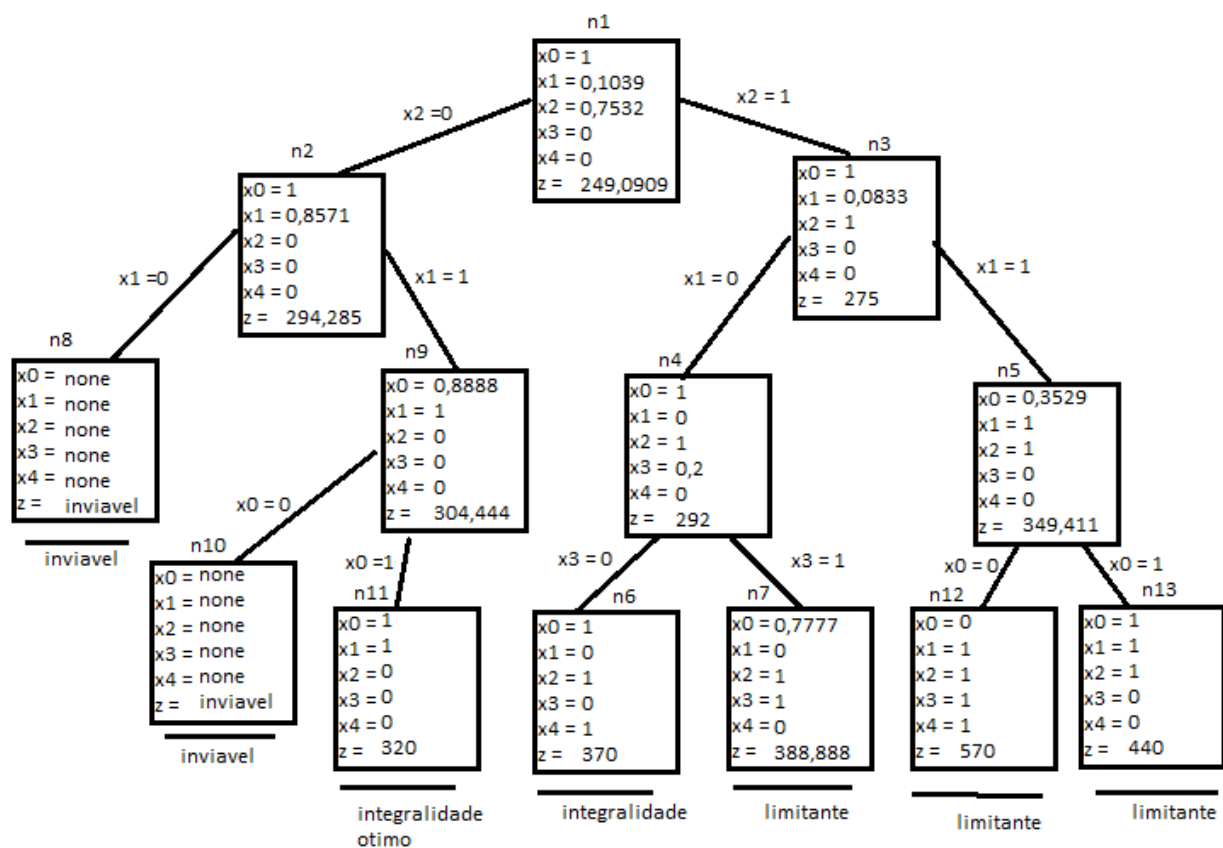


Figure 6: arvore 3

```

4 2
1.50 2.00 3.00 0.80
7 8 30 6 3
550 300 400 250 50

```

Figure 7: problema 4

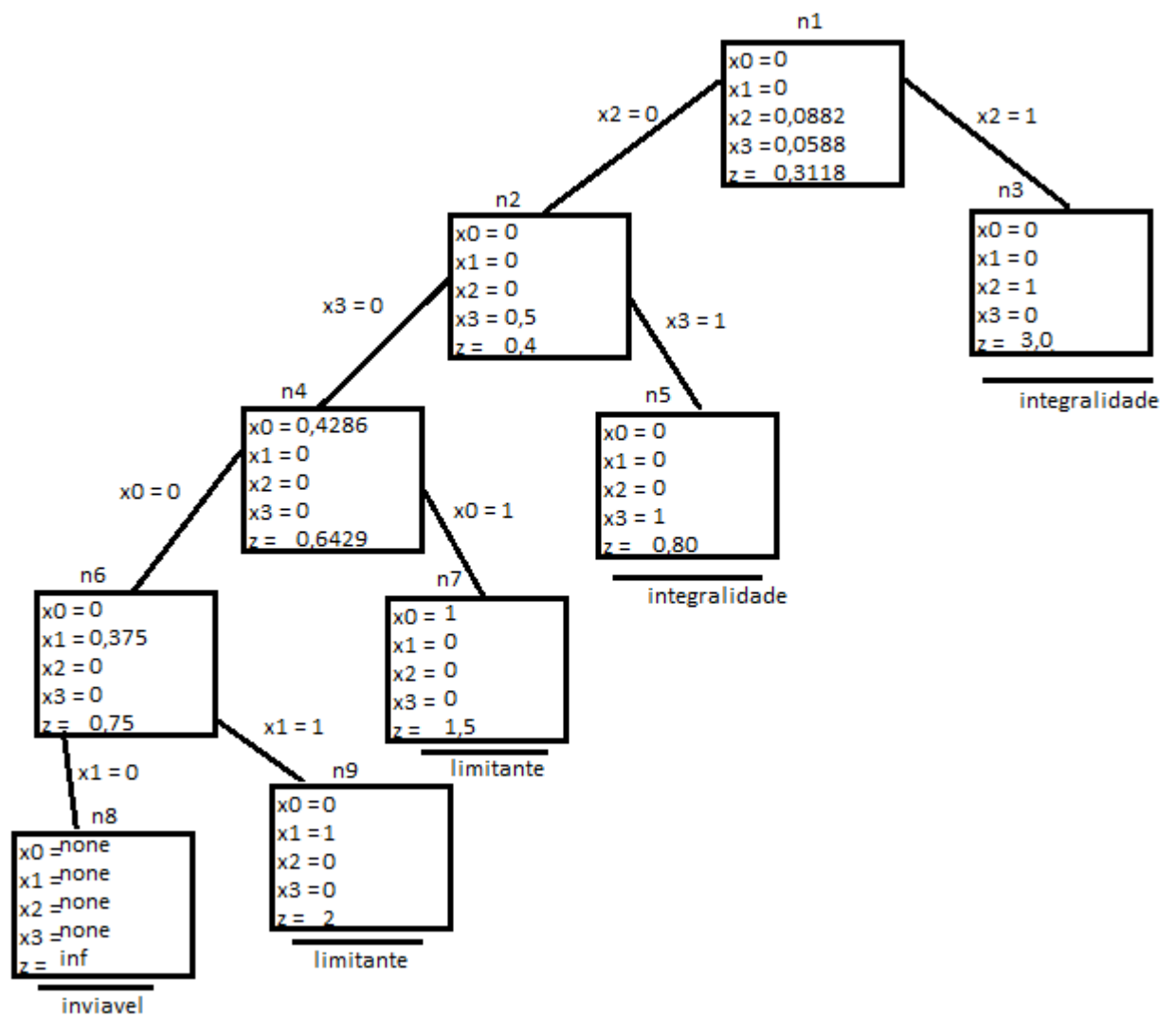


Figure 8: arvore 4

## 5 Conclusão

para resolução dos problemas e montar as arvores foram utilizados a calculadora online do simplex (PHPSimplex) e foram comparados com os resultados obtidos no algoritmo branch and bound do projeto, e as soluções ótimas com o próprio resolvidor do solver do cvxpy mostrado no algoritmo.

## 6 Referencias

"<http://www.phpsimplex.com/simplex/simplex.htm>"  
"https://github.com/ghackebeil/pybnb/blob/33b75b5251285a8318bc6d21f037050c7dbe65de/src/pybnb/solver.py"  
"https://docs.python.org/3/library/heapq.html"  
"https://docs.python.org/3/library/copy.html"  
"http://cvxr.com/cvx/doc/"