entrar valores:listas A,b,c ; valores: m,n

bibliotecas usadas:numpy (resolver sistema linear, multiplicar e redimensionar matriz)

funções definidas:

* check\_fun():

Ver valor da função objetivo dado X básico com seu custo

* particao():

Procura identidade para base na matriz A vendo se cada coluna é eletiva (apenas 1 um e o resto de zeros) e , se for, guarda o indice da linha do 1 e o índice dessa coluna, vai vendo as outras colunas e se forem eletivas, ve se ja nao tem a linha do 1 guardada, se não tiver, guarda essa coluna e a linha desse 1. Com essas colunas eletivas, é feita uma matriz. O processo continua até que seja possível formar uma identidade de tamanho m ou até acabar as colunas de A. As colunas que não forem selecionadas são guardadas em uma matriz como a particao nao básica com seus índices referentes a A inicial.

Se o número de colunas selecionadas for m, então essa pode ter forma de uma identidade e é uma base factível para iniciar o problema.

Se o número de colunas selecionadas não for m, recorremos ao problema de duas fases (função prob\_artf()). É introduzido um indicador para alertar que se chegou a solução ótima deste problema artificial, enquanto esse indicador não mudar, realizamos a função custo\_rel() para chegarmos na solução ótima, se o valor da função objetivo na solução ótima for zero, achamos uma partição básica factível para o problema inicial, então se remove os índices artificiais e seus colunas da matriz nao básica. O custo do problema inicial é então atualizado e particionado referente às colunas que ficaram na base e não base do problema artificial. Se o valor da função objetivo do problema artificial for diferente de zero, não temos uma partição básica factível para o problema inicial, então se retorna um indicador de problema infactível.

Se factível, retorna a partição básica e seus índices de posição.

* prob\_artf():

Para matriz das colunas eletivas, se adicionam colunas de zeros com 1 um nas posições de linhas faltantes para completar uma identidade e um vetor custo artificial com custo 1 sobre os índices das colunas criadas e zero para as colunas de A. Para esse problema artificial, a matriz de colunas eletivas de A e colunas adicionadas será uma base factível, a matriz não básica é a já estabelecida e o custo será este custo artificial.

* custos\_rel():

Recebe a partição básica, faz a solução básica, calcula o vetor multiplicador simplex, calcula os custos relativos e analisa se o custo mínimo é menor que zero.

Se for maior que zero, estamos na ótima ótima, então retorna um indicador que chegamos na sol ótima, sua partição básica e a solução ótima.

Se for menor que zero, não estamos na solução ótima, então ve qual o index da coluna não básica que gerou o menor custo relativo. Calcula o vetor direção simples com essa coluna.

Se a maior componente deste vetor for menor ou igual a zero, temos um problema limitado, então é retornado um indicador que não temos solução ótima finita.

Se houver componentes maiores que zero no vetor direcao, calculamos calculamos os passos possíveis para a iteração e selecionamos o menor e o índice de sua coluna base referente.

Troca a coluna base e a coluna não base e seus custos, atualizando também a lista de seus índices. Retorna a partição da iteração, os índices e a solução básica.

* simplex(m,n,c,b,A):

Redimensiona as listas, chama a função para partição (particao()), se for retornado indicador de infactibilidade, retorna "Não há solução factível". Se achar partição factível, é introduzido o indicador de solução ótima, chama funcao custos\_rel() até o indicador mudar, indica então se há solução finita, se não houver, retorna “Problema não tem solução ́ ótima finita”. Se tiver solução ótima, é calculado o valor da função objetivo e retorna a solução ótima com o valor ótimo da função.