

تمرین های درس بهینه سازی ترکیباتی - دکتر حسین کریمی

دانشجو: مصطفی کریمی

#===== تمرین اول =====#

✓ سه مورد از مثال های جایگشت ها را بنویسید

✦ روش تکنولوژی گروهی در طرح ریزی واحد های صنعتی
 ✦ شبکه های عصبی و الگوریتم های مرتبط بهش در طراحی سیستم ها در فصل 11 کتاب
 ✦ الگوریتم های تصمیم گیری چند معیاره یا چند شاخصه
 ✦ الگوریتم ژنتیک
 ✦ GNN - شبکه عصبی گرافی

#===== تمرین دوم =====#

✓ روش معاوضه یا سواپ

در این روش که مختص نمایش های جایگشتی است. جای دو جایگاه با هم عوض می شود.

First one we need **Python Library** for this method. i suggested **Numpy** and **Random** for matrix library in Python. for install python library package write **pip install numpy & pip install random** or for python 3 write **pip3 install numpy & pip3 install random** in CMD or Terminal.

M Karimi!

```

In [16]: # Swap Method in Combinatorial Optimizing Class - Dr Karimi
# import Numpy and Random Library
import numpy as np
import random as rd

# Number of Swap argument
n = int(input("How many argument from 0: "))

def swapPosition(n):
    '''
    Swap Position: swapPosition(integer number)
    swapPosition(n) -> change our random flow to new position

    this function try to change position from random position
    (simple flow by random number) to new position with choose
    2 position of simple place.

    Notic! We must install Numpy Library to run it
    Make shure to install numpy by CMD or Terminal with
    $ pip3 install numpy
    $ pip3 install random

    @@ Mostafa Karimi @@

    '''

    swap = np.random.permutation(n + 1)
    swap_rand = np.insert(swap, n + 1, swap[0])
    print("Our flow : {}".format(swap_rand))
    v = rd.sample(list(swap), 2)
    print("Random Swap Position Between: first random select pos {} in Swap list
    is {} and second pos {} is {}".format(int(v[0]+1), swap_rand[int(v[0])], int(v
    [1]+1), swap_rand[int(v[1])]))

    swap[v[0]], swap[v[1]] = swap[v[1]], swap[v[0]]
    swap = np.insert(swap, n+1, swap[0])

    print("New position is: {}".format(swap))
    return swap

c = swapPosition(n)

```

How many argument from 0: 8

Our flow : [4 1 5 7 3 0 6 8 2 4]

Random Swap Position Between: first random select pos 7 in Swap list is 6 and
second pos 8 is 8

New position is: [4 1 5 7 3 0 8 6 2 4]

#===== تمرین سوم =====#

روش معاوضه 2 جایگاه 

در این روش، جای دو ارتباط بین جایگاه ها با هم جابجا می شوند.

```

In [25]: # 2-Opt Method in Combinatorial Optimizing Class - Dr Karimi
# import Numpy and Random Library
import numpy as np
import random as rd

# Number of 2-opt argument
n = int(input("How many argument from 0: "))

def optTwo(n):
    '''
    2-Opt Position: optTwo(integer number)
    optTwo(n) -> change our random flow to new flow

    this function try to change graph from random position
    (simple flow by random number) to new flow with choose
    2 position of simple place.

    Notic! We must install Numpy Library to run it
    Make shure to install numpy by CMD or Terminal with
    $ pip3 install numpy
    $ pip3 install random

    @@ Mostafa Karimi @@

    '''

    optTwo = np.random.permutation(n + 1)
    swap_rand = np.insert(optTwo, n + 1, optTwo[0])
    print("Our flow : {}".format(swap_rand) )

    v = rd.sample(list(optTwo), 2)
    vMax = max(int(v[0]), int(v[1]))
    vMin = min(int(v[0]), int(v[1]))
    print("We change flow position [{} to {}] with [{} to {}]".format(swap_rand
[vMin], swap_rand[vMin+1], swap_rand[vMax], swap_rand[vMax+1]))

    # is not optimize, we use just 3 list for this function but use 5 list
    # we don't need opt min and max becuse is over need
    # develope and optimize for next time - Mostafa Karimi

    list1 = optTwo[0:vMin]

    optTwoMin = optTwo[vMin:vMin+1]

    reversList = list(reversed(optTwo[vMin+1:vMax]))

    optTwoMax = optTwo[vMax:vMax+1]

    list2 = optTwo[vMax+1:]

    finalList1 = list(list1) + list(optTwoMax) + reversList + list(optTwoMin) +
list(list2)

    finalList = np.insert(finalList1, n + 1, finalList1[0])
    print("And now, is new flow is {}".format(finalList))

    return optTwo

c = optTwo(n)

```


How many argument from 0: 10

Our flow : [2 10 0 8 7 5 3 6 4 9 1 2]

We change flow position [2 to 10] with [3 to 6]

And now, is new flow is [3 5 7 8 0 10 2 6 4 9 1 3]

#===== تمرین چهارم =====#

روش معکوس کردن 

این روش هم مخصوص نمایش های جایگشتی است

```

In [58]: # Rverse (Inversion) Method in Combinatorial Optimizing Class - Dr Karimi
# import Numpy and Random Library
import numpy as np
import random as rd

# Number of Reverse argument
n = int(input("How many argument from 0: "))

def reverseList(n):
    '''
    Reverse Position: reverseList(integer number)
    reverseList(n) -> reverse 2 random position from first list

    this function try to reverse select random number from first
    list and reversed 2 position from our random select position
    by select random number

    Notic! We must install Numpy Library to run it
    Make shure to install numpy by CMD or Terminal with
    $ pip3 install numpy
    $ pip3 install random

    @@ Mostafa Karimi @@

    '''

    firstList = np.random.permutation(n + 1)
    print(firstList)
    v = rd.sample(list(firstList), 2)

    vMax = max(int(v[0]), int(v[1]))
    vMin = min(int(v[0]), int(v[1]))

    print("First position from {} to {}".format(vMin+1, vMax+1))

    list1 = firstList[0:vMin]
    list2 = firstList[vMax+1:]
    reveList = list(reversed(firstList[vMin:vMax+1]))
    finalList = list(list1) + reveList + list(list2)

    print("Our new flow is: ", list1, " | ", "and resolve ", reveList, " | and
else ", list2)
    print("And merge: ", finalList)


    return

c = reverseList(n)

How many argument from 0: 6
[4 2 5 3 1 6 0]
First position from 1 to 6
Our new flow is:  [] |  and resolve  [6, 1, 3, 5, 2, 4] | and else  [0]
And merge:  [6, 1, 3, 5, 2, 4, 0]

```

#===== تمرین پنجم =====#

روش درج کردن 

این روش هم مخصوص نمایش های جایگشتی است

```

In [65]: # Insertion Method in Combinatorial Optimizing Class - Dr Karimi
# import Numpy and Random Library
import numpy as np
import random as rd

# Number of Inverse argument
n = int(input("How many argument from 0: "))

def invers(n):
    '''
    Inverse Position: inverse(integer number)
    invers(n) -> insert number from position to new position

    this function try to insert number from first position to new
    position by select random number from first list and delet old
    position from first list

    Notic! We must install Numpy Library to run it
    Make shure to install numpy by CMD or Terminal with
    $ pip3 install numpy
    $ pip3 install random

    @@ Mostafa Karimi @@

    '''
    firstList = np.random.permutation(n + 1)
    print("First List is: {}".format(firstList))
    v = rd.sample(list(firstList), 2)

    vMax = max(int(v[0]), int(v[1]))
    vMin = min(int(v[0]), int(v[1]))

    invers = np.insert(firstList, vMin, firstList[vMax])
    invers = np.delete(invers, vMax + 1)

    print("Do, argument {} move to position {}".format(firstList[vMax], vMin +
1))
    print("Invers to {}".format(invers))

    return invers

c = invers(n)

How many argument from 0: 9
First List is: [7 8 1 9 6 5 4 0 2 3]
Do, argument 0 move to position 5
Invers to [7 8 1 9 0 6 5 4 2 3]

```

#===== تمرین ششم =====#

الگوریتم شبیه سازی تبرید 

به نظر شما چه روش‌هایی برای تولید یک جواب اولیه برای مسائل بهینه‌سازی ترکیباتی یا مسائل بهینه‌سازی مورد مطالعه در مهندسی صنایع وجود دارد؟

In []: