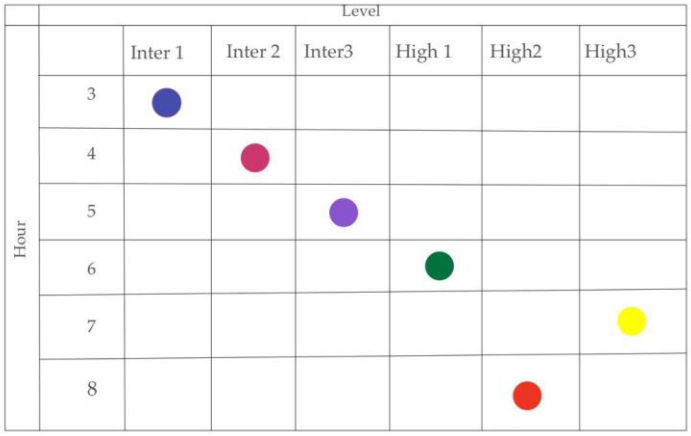
Al Imam Mohammad Ibn Saud Islamic University

College of Computer and Information Sciences

Computer Science Department

CS-340 Project Artificial Intelligent

problem scheduling math a tutor



Student :

1-Razan khalid bin zoba 439018661

2-Ethar Hassan suwaimil 439024017

3-Latefah Slman Abdallh 439018269

Supervised by : Dr.Areeb alowisheq

• Introduction :

Constrain in satisfaction problem (CSP) using to solve problem as searching algorithm can not solve large problem with constrain .

In CSP states and goal tests conform to a standard, and simple representation advantage of that: So general purpose heuristics rather than problem specific heuristics enable solving large problems [1]

There are many types, but we will cover these three types:

1- Arc Consistency AC-3 algorithm : it simplifies a constraint satisfaction problem using the constraints to prune out values from the variables domain.[5]

2- Backtracking algorithm : it uses the depth-first search method. When it starts exploring the solutions, a bounding function is applied so that the algorithm can check if the so-far built solution satisfies the constraints. If it does, it continues searching. If it doesn't, the branch would be eliminated, and the algorithm goes back to the level before.[5]

3- Forward Checking algorithm : it detects the inconsistency earlier than simple backtracking and thus it allows branches of the search tree that will lead to failure to be pruned earlier than with simple backtracking.[6]

المقدمة :

مشكلة (CSP):

مشكله لا تحل الا بتحقق الشروط الخاصة بها وخوارزميات البحث لا تحل مثل هذي المشاكل فلذلك تحتاج إلى خوارزميات تتعامل مع كمية كبيرة من الشروط . الهدف من هذي الخوارزميات هو الوصول الى جدول الأستاذ الذي يحقق الشروط هناك خوارزميات كثيره لكن سنغطي هنا الثلاث أنواع :

1- خوارزمية أقواس التناسق : يبسط مشكلة رضا القيد باستخدام القيود لاقتطاع القيم من مجال المتغيرات .

2- خوارزمية البحث بالتراجع : يستخدم طريقة البحث العميقة أولاً. عندما تبدأ في استكشاف الحلول ، يتم تطبيق وظيفة إحاطة بحيث يمكن للخوارزمية التحقق مما إذا كان الحل الذي تم إنشاؤه حتى الآن يفي بالقيود. إذا كان الأمر كذلك ، فإنه يستمر في البحث. إذا لم يحدث ذلك ، فسيتم حذف الفرع ، وتعود الخوارزمية إلى المستوى السابق .

3- خوارزمية الفحص الأمامي : يكتشف الفحص إلى الأمام عدم الاتساق في وقت أبكر من التراجع البسيط ، وبالتالي يسمح بفروع شجرة البحث التي ستؤدي إلى الفشل في التقليم في وقت أبكر من التراجع البسيط.

• problem scheduling math a tutor:

In this work will scheduling math a tutor for Intermediate and high School student.

The hour starting From 3PM to 9 PM. one hour for each level , tutor can't teach students to a different level in the same time.

CSP model :

- Variables: will be the level 3 for Intermediate student and 3 high School
  [Inter1, Inter2, Inter3, ,High1, High2, High3]
- Domains: will be hours [3,4,5,6,7,8]
- Constrains :
  Level 1 Intermediate student can't study with level 2 or level 3 and Au contraire
  Level 1 high school student can't study with level 2 or level 3 and Au contraire
  Intermediate student and high School doesn't study together in the same time
  Level 3 of Intermediate can't study in 6 PM
  Level 1 of Intermediate can't study in 4 PM
  Level 2 of high school can't study in 7 PM
- Relation:
  1- **R** $(I1, I2, I3)$:{ (3,4,5),(3,4,7),(3,4,8), (3,5,4),(3,5,7),(3,5,8), (3,6,5),(3,6,4),(3,6,7),(3,6,8), (3,7,5),(3,7,4),(3,7,8), (3,8,4),(3,8,7),(3,8,5), (5,4,3),(5,4,7),(5,4,8), (5,3,4),(5,3,7),(5,3,8), (5,6,3),(5,6,4),(5,6,7),(5,6,8), (5,7,3),(5,7,4),(5,7,8), (5,8,4),(5,8,7),(5,8,3), (6,4,3), (6,4,5), (6,4,7),(6,4,8),(6,3,4),(6,3,7),(6,3,5),(6,3,8),(6,5,3),(6,5,4),(6,5,7),(6,5,8),(6,7,3),(6,7,4),(6,7,5),(6,7,8), (6,8,4),(6,8,7),(6,8,3),(6,8,5), (7,4,3), (7,4,5),(7,4,8), (7,3,4),(7,3,5),(7,3,8), (7,5,3),(7,5,4),(7,5,8), (7,6,3),(7,6,4),(7,6,5),(7,6,8), (7,8,4) (7,8,3),(7,8,5),(8,4,3), (8,4,5), (8,4,7),(8,4,8),(8,3,4),(8,3,7),(8,3,5),(8,5,3),(8,5,4),(8,5,7),(8,5,8),(8,7,3),(8,7,4),(8,7,5),(8,7,8), (8,6,4),(8,6,7),(8,6,3),(8,6,5)}

2- **R($H1, H2, H3$)**:{(3,4,5),(3,4,6),(3,4,7),(3,4,8),(3,5,4),(3,5,7),(3,5,8),(3,5,6)
(3,6,5),(3,6,4),(3,6,7),(3,6,8), (3,8,4),(3,8,7),(3,8,5),(3,8,6) , (4,3,5),(4,3,6),(4,3,7),(3,4,8),
(4,5,3),(4,5,7),(4,5,8), (4,5,6) (4,6,5),(4,6,3),(4,6,7),(4,6,8), (4,8,3),(4,8,7),(4,8,5),(4,8,6),
(5,4,3),(5,4,7),(5,4,8),(5,4,6), (5,3,4),(5,3,7),(5,3,8),(5,3,6),(5,6,3),(5,6,7), 5,6,4),(5,6,8)
(5,8,4),(5,8,7),(5,8,3),(5,8,6), (6,4,3), (6,4,5), (6,4,7),(6,4,8), (6,3,4),(6,3,7),(6,3,5),(6,3,8),
(6,5,3),(6,5,4),(6,5,7),(6,5,8), (6,8,4),(6,8,7),(6,8,3),(6,8,5), (7,4,3), (7,4,5),(7,4,6),(7,4,8),
(7,3,4),(7,3,5),(7,3,6),(7,3,8), (7,5,3),(7,5,4), (7,5,6),(7,5,8), (7,6,3),(7,6,4),(7,6,5),(7,6,8),
(7,8,4),(7,8,6),(7,8,3),(7,8,5),(8,4,3),(8,4,5),(8,4,6),(8,4,7),(8,3,4),(8,3,7),(8,3,6),(8,3,5),(8,3,
8),(8,5,3),(8,5,4),(8,5,6),(8,5,7),(8,5,8),(8,6,4),(8,6,7),(8,6,3),(8,6,5)}

3- **R($I1, H1$)**:{(3,4),(3,5),(3,6),(3,7),(3,8),(5,4),(5,6),(5,7),(5,8),(5,3),(6,4),(6,5),(6,3),(6,7),(6,8),
(7,4),(7,5),(7,3),(7,6),(7,8),(8,4),(8,5),(8,3),(8,7),(8,6)}

4- **R($I2, H2$)**:{(3,4),(3,5),(3,6),(3,8),(4,3),(4,5),(4,6),(4,8),(5,4),(5,6),(5,8),(5,3),(6,4),(6,5),(6,3),
(6,8), (7,4),(7,5),(7,3),(7,6),(7,8),(8,4),(8,5),(8,3),(8,6)}

5- **R($I3$, $H3$)**:{(3,4),(3,5),(3,6),(3,7),(3,8),(4,3),(4,5),(4,6),(4,7),(4,8),(5,4),(5,6),(5,7),(5,8),(5,3)
,(7,4),(7,5),(7,3),(7,6),(7,8),(8,4),(8,5),(8,3),(8,6),(8,7)}

-Solution:{ (3,4,5,6,8,7),(3,4,7,6,8,5),(3,5,4,7,8,6),(3,8,4,7,5,6),(5,3,4,6,8,7),(5,4,3,6,8,7)(3,5,4,
8,6,7),(3,5,7,6,8,4),(3,5,8,7,6,4),(3,6,5,8,4,7),(3,6,4,7,4,8),(3,6,7,4,8,5),(3,6,8,4,5,7),(3,7,5,4,6,
8),(3,7,4,6,5,8),(3,7,8,5,6,4),(3,8,4,5,6,7),(3,8,7,4,5,6),(3,8,5,4,6,7),(7,8,5,4,6,3),(7,6,5,4,6,3),(
7,5,6,4,8,3),(8,7,5,6,4,3),(8,7,5,6,3,4),(8,6,7,5,4,3),(8,6,7,5,3,4)(8,5,7,6,3,4),(8,5,7,6,4,3),(8,7,
4,5,3,6)(8,7,4,5,6,3)}

(جدولة جدول أستاذ رياضيات) :

في هذا العمل ,سوف نجدول جدول استاذ رياضيات خصوصي للمرحلة المتوسط و الثانوية و سوف يبتدأ عمله من الساعة 3 عصرا الى 9 مساء ,ساعة لكل مرحله في المتوسط او الثانوية , بحيث لا يدرس اكثر من مرحلة في الوقت نفسه .

نموذج CSP

المتغيرات:

متوسط1 (م1) , متوسط2 (م2) , متوسط3 (م3) , الثانوية1 (ث1) , الثانوية2 (ث2) , الثانوية3 (ث3)

المجال :

(8-7-6-5-4-3)

الشروط :

م1 لا يدرس مع م2 و م3 معا

ث1 لا يدرس مع ث2 و ث3 معا

م1 م2 م3 لا يدرسون مع ث1 ث2 ث3 في نفس الوقت

م3 لا يدرس الساعه 6

م1 لا يدرس الساعه 4

ث2 لا يدرس الساعه 7

العلاقات :

R (م1, م2, م3):{ (3,4,5),(3,4,7),(3,4,8), (3,5,4),(3,5,7),(3,5,8), (3,6,5),(3,6,4),(3,6,7),(3,6,8), (3,7,5),(3,7,4),(3,7,8), (3,8,4),(3,8,7),(3,8,5), (5,4,3),(5,4,7),(5,4,8), (5,3,4),(5,3,7),(5,3,8), (5,6,3),(5,6,4),(5,6,7),(5,6,8), (5,7,3),(5,7,4),(5,7,8), (5,8,4),(5,8,7),(5,8,3), (6,4,3), (6,4,5), (6,4,7),(6,4,8),(6,3,4),(6,3,7),(6,3,5),(6,3,8),(6,5,3),(6,5,4),(6,5,7),(6,5,8),(6,7,3),(6,7,4),(6,7,5),(6,7,8), (6,8,4),(6,8,7),(6,8,3),(6,8,5), (7,4,3), (7,4,5),(7,4,8), (7,3,4),(7,3,5),(7,3,8), (7,5,3),(7,5,4),(7,5,8), (7,6,3),(7,6,4),(7,6,5),(7,6,8), (7,8,4) (7,8,3),(7,8,5),(8,4,3), (8,4,5), (8,4,7),(8,4,8),(8,3,4),(8,3,7),(8,3,5),(8,5,3),(8,5,4),(8,5,7),(8,5,8),(8,7,3),(8,7,4),(8,7,5),(8,7,8), (8,6,4),(8,6,7),(8,6,3),(8,6,5)}

R(ث1, ث2, ث3):{(3,4,5),(3,4,6),(3,4,7),(3,4,8),(3,5,4),(3,5,7),(3,5,8),(3,5,6) (3,6,5),(3,6,4),(3,6,7),(3,6,8), (3,8,4),(3,8,7),(3,8,5),(3,8,6) , (4,3,5),(4,3,6),(4,3,7),(3,4,8), (4,5,3),(4,5,7),(4,5,8), (4,5,6) (4,6,5),(4,6,3),(4,6,7),(4,6,8), (4,8,3),(4,8,7),(4,8,5),(4,8,6), (5,4,3),(5,4,7),(5,4,8),(5,4,6), (5,3,4),(5,3,7),(5,3,8),(5,3,6),(5,6,3),(5,6,7), 5,6,4),(5,6,8) (5,8,4),(5,8,7),(5,8,3),(5,8,6), (6,4,3), (6,4,5), (6,4,7),(6,4,8), (6,3,4),(6,3,7),(6,3,5),(6,3,8), (6,5,3),(6,5,4),(6,5,7),(6,5,8), (6,8,4),(6,8,7),(6,8,3),(6,8,5), (7,4,3), (7,4,5),(7,4,6),(7,4,8), (7,3,4),(7,3,5),(7,3,6),(7,3,8), (7,5,3),(7,5,4), (7,5,6),(7,5,8), (7,6,3),(7,6,4),(7,6,5),(7,6,8), (7,8,4),(7,8,6),(7,8,3),(7,8,5),(8,4,3),(8,4,5),(8,4,6),(8,4,7),(8,3,4),(8,3,7),(8,3,6),(8,3,5),(8,3,8), (8,5,3),(8,5,4),(8,5,6),(8,5,7),(8,5,8),(8,6,4),(8,6,7),(8,6,3),(8,6,5)}

R(م1, ث1):{(3,4),(3,5),(3,6),(3,7),(3,8),(5,4),(5,6),(5,7),(5,8),(5,3),(6,4),(6,5),(6,3),(6,7),(6,8), (7,4),(7,5),(7,3),(7,6),(7,8),(8,4),(8,5),(8,3),(8,7),(8,6)}

R(م2, ث2):{(3,4),(3,5),(3,6),(3,8),(4,3),(4,5),(4,6),(4,8),(5,4),(5,6),(5,8),(5,3),(6,4),(6,5),(6,3),(6, 8), (7,4),(7,5),(7,3),(7,6),(7,8),(8,4),(8,5),(8,3),(8,6)}

R(م3 , ث3):{(3,4),(3,5),(3,6),(3,7),(3,8),(4,3),(4,5),(4,6),(4,7),(4,8),(5,4),(5,6),(5,7),(5,8),(5,3),(7,4),(7,5),(7,3),(7,6),(7,8),(8,4),(8,5),(8,3),(8,6),(8,7)}

الحل:

{(3,4,5,6,8,7),(3,4,7,6,8,5),(3,5,4,7,8,6),(3,8,4,7,5,6),(5,3,4,6,8,7),(5,4,3,6,8,7)(3,5,4,8,6,7),(3,5,7,6,8,4),(3,5,8,7,6,4),(3,6,5,8,4,7),(3,6,4,7,4,8),(3,6,7,4,8,5),(3,6,8,4,5,7),(3,7,5,4,6,8),(3,7,4,6,5,8),(3,7,8,5,6,4),(3,8,4,5,6,7),(3,8,7,4,5,6),(3,8,5,4,6,7),(7,8,5,4,6,3),(7,6,5,4,6,3),(7,5,6,4,8,3),(8,7,5,6,4,3),(8,7,5,6,3,4),(8,6,7,5,4,3),(8,6,7,5,3,4)(8,5,7,6,3,4),(8,5,7,6,4,3),(8,7,4,5,3,6)(8,7,4,5,6,3)}

- **Problem formulation:**

-Initial state: The scheduling of the tutor is empty (have not the student select hour) or has more than level in the same time.

-Action : Assignment hour ,keep hour ,none available hour or multi hour.

-State Space:

1- check if scheduling is complete (all level have one hour and not in same time) is true and done scheduled

2- if not done go any level then assignment hour

3- if no constrain false constrain as segmented

4- if not will change hour to get hour correct

5- then will iterative for all level

-Goal state: all levels in scheduling have one hour and all not all same time.

الحالة الابتدائي:

الجدول الاستاذ من دون الطلاب ولا الساعة

ما ستفعله الخوارزمية:

تسند لكل مستوى(طلاب) ساعة لا يأخذها غيرهم

مساحة الحل:

1- تأكد انه الجدول اكتمل

2- اسند ساعة للفصول بدون ساعه

3- اذا لم يوجد شرط لم يتحقق اسند له الساعة

4- اذا كان لا يحقق الشروط لا تسنده

5- ثم اعيد الخطوات السابقة

الحل المطلوب: كل المستويات تم إسناد لها ساعة

- **Heuristics for CSPs:** Most constrained variable: choose the variable with the fewest legal values Least constraining value: Given a variable, choose the least constraining value, the one that rules out the fewest values in the remaining variables. Then when time complexity will low with heuristics.[1]

اختيار المتغير ذات القيود الكثيرة و القيم الذي يحتوي على أقل عدد من الشروط و اسندها إلى لمتغير الذي لديه قيود كثيره , القيمة التي تستبعد أقل القيم في المتغيرات المتبقية. بتالي ينخفض الوقت مع الاستدلال .

• Algorithm:

1- Arc Consistency:

function AC-3(*csp*) **returns** false if an inconsistency is found and true otherwise
  inputs: *csp*, a binary CSP with components (X, D, C)
  local variables: *queue*, a queue of arcs, initially all the arcs in *csp*

  while *queue* is not empty **do**
    $(X_i, X_j) \leftarrow$ REMOVE-FIRST(*queue*)
    if REVISE(*csp*, X_i, X_j) **then**
      if size of $D_i = 0$ **then return** *false*
      for each X_k **in** X_i.NEIGHBORS - $\{X_j\}$ **do**
        add (X_k, X_i) to *queue*
  return *true*

---

function REVISE(*csp*, X_i, X_j) **returns** true iff we revise the domain of X_i
  revised $\leftarrow$ *false*
  for each x **in** D_i **do**
    if no value y in D_j allows (x,y) to satisfy the constraint between X_i and X_j **then**
      delete x from D_i
      revised $\leftarrow$ *true*
  return *revised*

الأسهم المتناسقة (ARC- Consistency) :

راس الدالة يحتوي على المشكلة **يرجع** الحل او لا يوجد حل

المدخلات :المشكلة تحتوي على (المغيرات و المجال و الشروط)

المتغيرات المحلية : متغير نوعه صف يحتوي على المشكلة

ندخل في عملية تكرار تمشي على طول المتغير المحلي

نأخذ اول عنصر في المتغير المحلي ونضعه في متغير X

اذا كانت القيمة القادمة من دالة الريسيف (المستقبل) التي تأخذ المشكلة و اول عنصر في المتغير المحلي اذا كان صحيح ندخل الشرط التالي

اذا كان حجم المجال = 0 بتالي لا يوجد حل

يكرر بحيث يمشي على كل الجيران المتغير X

إرجاعX للمتغير المحلي

بتالي هناك حل

راس دالة الريسيف (المستقبل)(يستقبل المشكلة والمتغير و جارته) ارجع صحيح اذا كان المجال صحيصح لي المتغير

متغير من نوع قائمة تحتوي على خطاء

دالة التكرار يمشي على كل المجال

اذا كان لا يوجد حل يحقق من الشروط على المجال المعطى لي المتغير و جارتها

يتم حذفه من المجال

يجعل الخانة في المتغير = صحيح

ارجع لي المتغير من نوع قائمة

2- Backtracking with Ac-3

```
function RECURSIVE-BACKTRACKING( assignment, csp) returns a solution, or
failure
    if assignment is complete then return assignment
    var ← SELECT-UNASSIGNED-VARIABLE( Variables[csp], assignment, csp)
    for each value in  CSP domains  do
        if value is consistent with assignment according to Constraints[csp] then
            add { var = value } to assignment
            result ← RECURSIVE-BACKTRACKING(assignment, csp)
            if result ≠ failue then return result
            remove { var = value } from assignment
    return failure
```

```
function is consistent(assegment, csp constrain )
    counter=0 ,to count the inter in constrain
     for con in each csp constrain
       if con is constrain in assegment
         return false
```

return true

البحث بالتراجع:

راس الدالة يحتوي على المشكلة و متغير من نوع قائمة يحتوي على بعض من الحل

اذا كان المتغير يحتوي على كامل الحل ارجع لي هذا المتغير

المتغير يحتوي على المتغيرات التي لا تحتوي على قيم

دالة التكرار تمشي على المجال في المشكلة المعطاة

اذا كان كانت هذه القيمة من المجال تحقق دالة الشروط

اسندها لي المتغير الذي لا يملك قيمه

متغير الناتج نوع لسته اسند له الناتج من استدعى دالة البحث بالتراجع

اذا كان الناتج لا يسوي خطاء ارجع لي الناتج

امسح من المتغير المرسل

وارجع خطاء

راس دالة الشروط الذي يحتوي على الحل المرسل و الشروط لهذه المشكلة

العداد =0 لعد عدد المرات دخول على هذه الدالة

دالة التكرار تمشي على الشروط في المشكله المرسله

اذا كان لا يحققه ارجع لي خطاء

ارجاع القيمه صحيحه

- Implementation in jupyter Notebook:

  https://mybinder.org/v2/gh/ipython/ipython-in-depth/7e5ce96cc9251083979efdfc393425f1229a4a68?filepath=binder%2FUntitled.ipynb

Implementation Arc Consistency : [3]

```
88
89  def ac3(arcs):
90      """
91      Update `domains` such that each variable is arc consistent.
92      """
93      # Add all the arcs to a queue.
94      queue = arcs[:]
95
96      # Repeat until the queue is empty
97      while queue:
98          # Take the first arc off the queue (dequeue)
99          (x, y) = queue.pop(0)
100
101         # Make x arc consistent with y
102         revised = revise(x, y)
103
104         # If the x domain has changed
105         if revised:
106             # Add all arcs of the form (k, x) to the queue (enqueue)
107             neighbors = [neighbor for neighbor in arcs if neighbor[1] == x]
108             queue = queue + neighbors
109     if domains["High2"] == [7]:
110             domains["High2"]=None
111     if domains["Inter1"] == [4]:
112             domains["Inter1"]=None
113     if domains["Inter3"] == [6]:
114             domains["inter3"]=None
115
```

```
65
66  def revise(x, y):
67      revised = False
68
69      x_domain = domains[x]
70      y_domain = domains[y]
71
72      all_constraints = [
73          constraint for constraint in constraints if constraint[0] == x and constraint[1] == y]
74
75      for x_value in x_domain:
76          satisfies = False
77          for y_value in y_domain:
78              for constraint in all_constraints:
79                  constraint_func = constraints[constraint]
80                  if constraint_func(x_value, y_value):
81                      satisfies = True
82          if not satisfies:
83              x_domain.remove(x_value)
84              revised = True
85
86      return revised
87
88
```

def ac3(arcs):

1. Get all the constraints and turn each one into two arcs.

2. Add all the arcs to a queue.

3. Repeat until the queue is empty:

    3.1. Take the first arc (x, y), off the queue (dequeue).

    3.2. For every value in the x domain, there must be some value of the y domain.

    3.3. def revise(x, y): Make x arc consistent with y. To do so, remove values from x

domain for which there is no possible corresponding value for y domain.

3.4. If the x domain has changed, add all arcs of the form (k, x) to the queue (enqueue).

1-احصل على جميع القيود وقم بتحويل كل واحد الى قوسين

2-اضف جميع الاقواس الى قائمة الانتظار

3-كرر حتى تصبح قائمة الانتظار فارغه

4-اخرج من المتغير الصفي و خذ القوس الأول

5-يجب أن يكون هناك بعض قيمة المجال ل قيمة في المجال

6.-مراجعة الاقواس اجعل اكس قوسًا متسقًا مع واي للقيام بذلك ، قم بإزالة القيم من المجال الذي لا توجد له قيمة مقابلة محتملة للمجال

7- إذا تغير المجال ، أضف جميع أقواس النموذج (اكس .كاي) إلى قائمة الانتظار (قائمة الانتظار).

- Implementation Backtracking : [4]

```python
def is_complete(assignment):
  return None not in (assignment.values())

def select_unassigned_variable(variables, assignment):
  for var in variables:
    if assignment[var] is None:
      return var

def is_consistent(assignment, constraints):
  global counter
  counter += 1
  for constraint_violated in constraints:
    if constraint_violated(assignment):
      return False
  return True

def init_assignment(csp):
  assignment = {}
  for var in csp[VARIABLES]:
      # dead stat
      V = select_unassigned_variable(csp[VARIABLES], domains)
      if V is not None:
         assignment[var]=None
      # backtacing without AC-3
      else:
         assignment[var] =None
  return assignment
```

```python
def recursive_backtracking(assignment, csp):
  if is_complete(assignment):
    return assignment
  var = select_unassigned_variable(csp[VARIABLES], assignment)
  for value in csp[DOMAINS]:
    assignment[var] = value
    if is_consistent(assignment, csp[CONSTRAINTS]):
      result = recursive_backtracking(assignment, csp)
      if result != FAILURE:
        return result
    assignment[var] = None
  return FAILURE
```

def recursive_backtracking(assignment, csp):

1. def init_assignment(csp): The Algorithm begins to build up a solution, starting with an empty solution set (Assignment) . Assignment = {}

2. is_complete(assignment): Add to (Assignment) the first move that is still left (All possible moves are added to (Assignment) one by one).

3. def select_unassigned_variable(variables, assignment): Assign values to variables .

4. def is_consistent(assignment, constraints): Check if (Assignment) satisfies each of the constraints .

If Yes, then the so-far built solution satisfies the constraints and it continues searching.

Else, the branch would be eliminated, and the algorithm goes back to the level before.

دالة البحث بالتراجع (مهمة ,المشكلة):

1- {} = تعيين دالة أولية : تبدأ الخوارزمية في بناء حل ، بدءًا من مجموعة حل فارغة (التعيين). واجب

2- هل هي مكتمله (مهمة): أضف إلى (التعيين) الحركة الأولى المتبقية (تتم إضافة جميع الحركات الممكنة إلى (التعيين) واحدة تلو الأخرى.

3- المتغيرات بدون قيم (المتغيرات ، التخصيص): قم بتعيين القيم إلى المتغيرات.

4- التحقق من القيود (مهمة ، قيود): تحقق مما إذا كان (التعيين) يفي بكل من القيود .إذا كانت الإجابة بنعم ، فإن الحل الذي تم إنشاؤه حتى الآن يلبي القيود ويستمر في البحث.

..عدا ذلك ، سيتم التخلص من الفرع ، وتعود الخوارزمية إلى المستوى السابق.

- Result:

```
The Arc3 is :
{'Inter1': [3], 'Inter2': [4], 'Inter3': [5], 'High1': [6], 'High2': None,
    'High3': [8]}
the time is 353.81317138671875 MicroSecond
The Backtacing is :
{'Inter1': 3, 'Inter2': 6, 'Inter3': 4, 'High1': 5, 'High2': 8, 'High3': 7}
39
the time is: 189.06593322753906 MicroSecond
>
```

	Arc constrain الاسهم المتناسقه	Backtracking with AC-3 البحث بتراجع
Time الوقت	353.8 microsecond	189 microsecond
Big oh مدى التعقيد	$O(ed^3)$1	$O(d^n)$

(1) e is the number of constraints and d is the size of the maximum domain in a problem

It turns out that the time of Arc constrain higher than Backtracking and Arc constrain not complete solution may be have dead state.

اتضح لنا أن الاسهم المتناسق تستغرق وقتاً اكبر من البحث بالتراجع ، وايضا من الممكن عدم ظهور حل كامل (حاله مميته) في الاسهم المتناسقة.

- **Reference:**

[1] slied the course AI unveracity imam Mohamed bin Saud Islamic 6.1 (Dr.Areeb alowisheq)

Lecture 6.1.pptx

[2] slied the course AI unveracity imam Mohamed bin Saud Islamic 6.2 (Dr.Areeb alowisheq)

Lecture 6.2 Updated.pptx

[3] Cesar William Alvarenga https://medium.com/swlh/how-to-solve-constraint-satisfaction-problems-csps-with-ac-3-algorithm-in-python-f7a9be538cfe

[4] Steven Collins https://nextjournal.com/lomin/constraint-satisfaction-problems-and-functional-backtracking-search

[5] https://www.baeldung.com/cs/backtracking-algorithms , https://www.geeksforgeeks.org/backtracking-introduction/

[6] https://ktiml.mff.cuni.cz/~bartak/constraints/propagation.html