استخدام طرق الامثلية الذكائية في تقييم المسار

سـرى رمـزي شـريف
قسم علوم الحاسبات
كلية علوم الحاسبات والرياضيات
جامعة الموصل

تاريخ الاستلام	تاريخ القبول
2004/11/1	2005/5/10

ABSTRACT

Finding the path has a great importance in planning robot track with a certain environment ,finding the most suitable track between two cities in a city map and finally finding the suitable ways for the police and emergency board.

The suitable track between two sites has been evaluated by using the traditional algorithm which were used for such a purpose as one of the intelligence techniques with the technology of solving problems which include the ideal method to find the most suitable research . As a result, it appeared that the intelligence technique seemed to the best method as, it makes a balance between the shortest space and the weight of the obstacle .Further ,it is adequate for finding a solution , the most suitable solution –Unique optional solution . C++ was also used as a language in the research.

الخلاصة

إن إيجاد المسار له أهمية كبيرة في تخطيط مسار الإنسان الآلي ضمن بيئة معينـــة ، إيجـــاد المسار الاوفق بين مدينتين في خارطة مدن ، وكذلك إيجاد الطرق الاوفق لأجهزة الشرطة والإسعاف .

تم تقييم المسار ألاوفق ما بين موقعين باستخدام الخوارزميات التقليدية المعروفة والتي كـــانت تستخدم لهذا الغرض وإحدى التقنيات الذكائية ضمن تقنية حل المسائل متضمنة الطريقة المثلى لإيجـــاد بحث أوفق ونتيجة هذا التقييم ظهر إن التقنية الذكائية في الامثلية هي الأوفق وذلك لأنها توازن ما بيـــن المجال الأقصر ووزن العائق وكذلك تكتفي في إيجاد حـــل وحيـــد وهـــو الحـــل الاوفـــق (Unique Optional Solution). كما تم استخدام لغة (c++) كلغة برمجة في البحث.

المقدمة

في اغلب الأحيان نرى إن الكثير يبحث عن الخوارزمية المثلى لإيجاد أفضل مسار بين موقعين حيث إن الوصول إلى إيجاد المسار الأمثل غالبا ما يكون مكلف جدا ولغرض إجراء عملية البحث يجب إن يكون لدينا مخطط ويمكن أن تستخدم المخططات في خارطات الطرق (Road Map) وفي شبكات الاتصالات (Communication Network)(1). ففي خارطة المدن يعبر عن المدن بالعقد (nodes) ويعبر عن الطرق الواصلة بين تلك المدن بخطوط مستقيمة تصل تلك العقد في المخطط (2).

إن حل مشكلة إيجاد أفضل مسار تعني الحصول على مسار أولي بين العقدتين اللذان يمثلان المدينتين المعلومتين وان المسار الامثل هو المسار الذي تكون جميع النقاط المكونة له مختلفة ، أي لا يوجد تكرار لأي عقدة ضمن ذلك المسار.

من السهل جدا إيجاد المسار الأقصر بين عقدة المصدر (Source node) وعقدة الهدف (Goal node) أو إيجاد المسار إلى أي عقدة أخرى في مخطط المدن أو أي مخطط أخر بغض النظر عن البحث في أمثلية ذلك المسار بين أي عقدتين (1 و 3).

1.التنقيب (Heuristic) :

إن كلمة Heuristic هي كلمة إغريقية جاءت من الكلمة (heuriskein) وتعني (To discover) أي طريقة تعيين الهدف المطلوب .أن استعمال كلمة Heuristic يعتبر وصفاً للحالات التي يرغب في إيجاد الحل الأفضل لها تبعاً لطرق أفضل مقياس (4).

وبالإمكان أن تعرف بمجموعة من التجارب العملية التي تعتمد على مجموعة قوانين لاختبار الحالة اللاحقة (next state) للحالة الحالية(current state) أثناء عملية البحث.

كلما كانت طرق التنقيب (Heuristic) جيدة نستطيع الحصول على حل جيد للمشاكل الصعبة ، ومن الأمثلة التي يطبق فيها التنقيب (Heuristic) بشكل جيد هي مسالة (Nearest neighbor) المطبقة على خوارزمية (Sales man) والتي تعتبر تطوير لخوارزمية أفضل كلفة (Best cost search). تكون طرق التنقيب غير واضحة المعالم ما لم نستخدم دالة التقييم (Evaluation function) (5).

2. دالة التقييم (Evaluation function) :

أن لكل عقدة هنالك قيمة تمثل كلفة العقدة حيث يكون الانتقال من الحالة (العقدة) الحالية (Current state) إلى الحالة (العقدة) التالية (Next state) التي تملك اقل قيمة وتمسى أيضا معوقات الحركة من عقدة إلى عقدة أخرى ، وهكذا يكون الانتقال في المخطط اعتمادا على هذه القيم التي

تمثل قيمة كلفة المسارات للوصول إلى الهدف (Final state) ، ولقد تم استخدام هذه الدالة مع كل خوارزمية وخوارزمية البحث العرضي العمقي ابتداء (Depth first search) وخوارزمية البحث العرضي ابتداء (Breadth first search) للحصول على طريقة أكثر تطور هي (6) :

Best-first-search=Hill-Climbing=Depth-first-search + evaluation function.

ببساطة أل Best-first-search تدعى أل Hill-Climbing إذ ليس فقط تقيم اكبر سلسلة للحالة بل تقيم المسار المؤخذ من الحالة البدائية إلى أن نصل إلى حالة الهدف (4).

حيث عند البحث عن مسار ضمن المخطط الشجري (Tree graph) باستخدام طريقة التسلق (hill-climbing) حيث يكون التسلق أو الانتقال من المستوى الحالي (Current level) إلى المستوى اللاحق (Next level) باستخدام المسار الذي يمتلك اقل قيمة . أما الطريقة الثانية (first-best) فيتم الانتقال من المستوى الحالي إلى اللاحق باستخدام اقل قيمة للعقدة بعد ذلك يتم مقارنة المستوى الحالي بالمستوى اللاحق فإذا كانت قيمة المقارنة كبيرة فيتم الرجوع إلى نفس المستوى ويتم اختيار العقدة الأخت للعقدة الأولى تمتلك اقل قيمة ونستمر بهذه المقارنة حتى وصولنا إلى الهدف (Goal state) (2 ، 4 و 5).

ولغرض الحصول على طريقة بحث أكثر فعالية يتم تضمين دالة الكلفة (Cost function) للطرق المذكورة في أعلاه حيث من المتوقع أن تصبح من أفضل الطرق المستخدمة في المستقبل .

3. دالة الكلفة (Cost function) :

هي عبارة عن دالة تعطي كل مسار ابتداء من الحالة الابتدائية (Initial state) قيمة معرفة حيث يكون الانتقال إلى الحالة التالية التي سوف يمتلك المسار فيها اقل قيمة ، وهكذا يتم الانتقال إلى باقي العقد اعتمادا على اقل مجموع لقيم المسارات من الحالة الابتدائية والحالية.(7) .

وعند تطبيق هذه الدالة تم الحصول على الأتي :

Best-cost- Search =Breadth-first-Search +Cost Function

Best-first- Search = Best-cost-Search +Evaluation Function

إن الطريقة الأولى تسمى أيضا التفرع والاحتساب (Branch and Bound) وهي قريبة من خوارزمية (Dijksra).(1) ، والطريقة التي تعتبر من أهم طرق البحث الأفضل في إيجاد الحل الامثل بالامكان إن تعتبر :

A*=Best-Cost-Search + Evaluation function

وتسمى أيضا (A* Technique) . حيث إن الطريقة تحاول البحث للحصول على المسار الامثل وسوف يتم تقييمها ومقارنتها مع خوارزمية (Dijksra).(1)،(8).

1.المسار الأقصر (Dijksra Algorithm) Shortest path :

من أحدى التطبيقات الشائعة لحل مشكلة إيجاد المسار بين عقدتين في المخطط الموزون (Weighted graph) أو غير الموزون (Digraph) هو تطبيق (Dijksra) لإيجاد اقصر أو اقل كلف للمسار بين عقدتين في المخطط (1 ، 9 و 10).

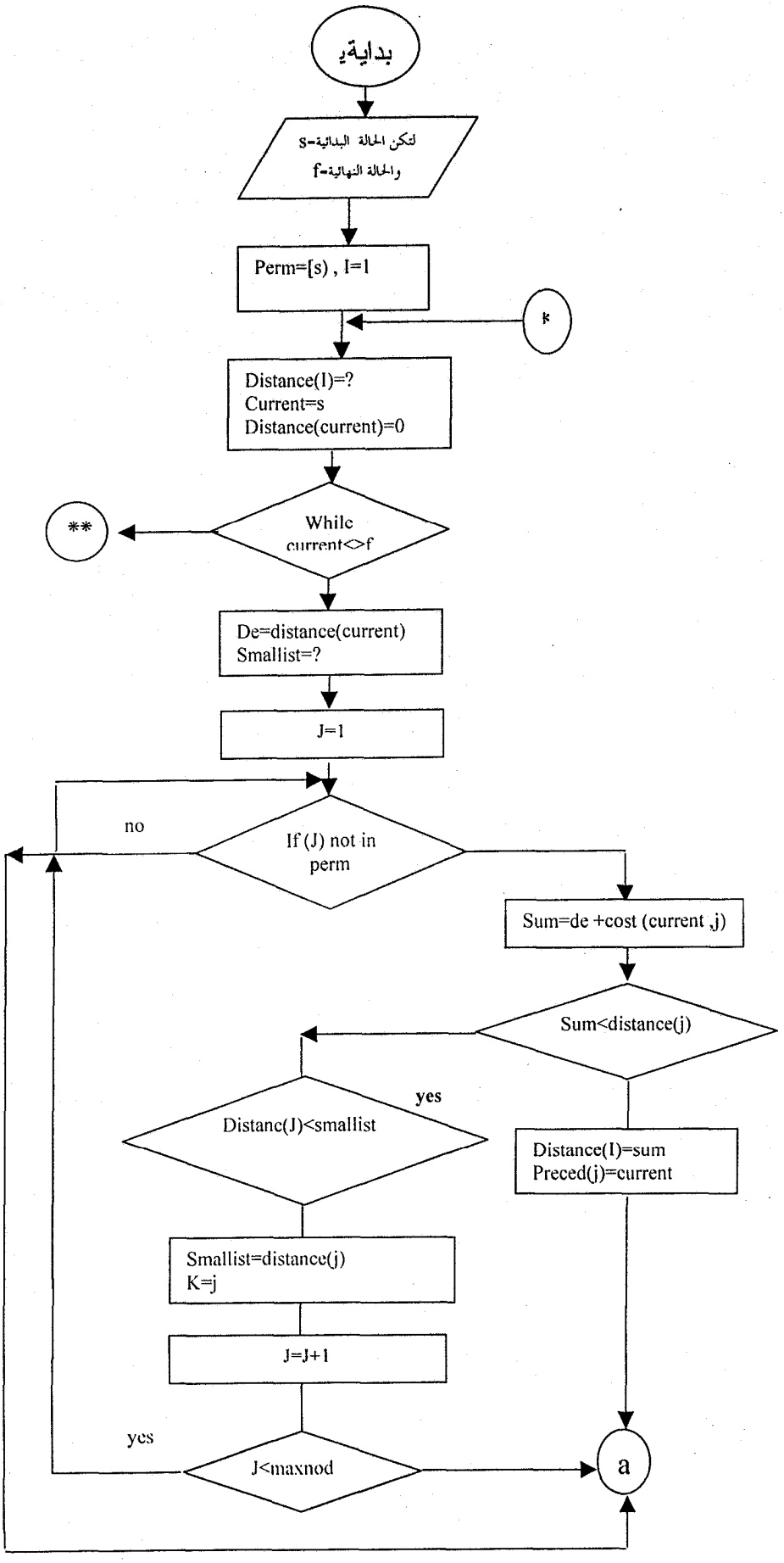
من خلال العلاقة :

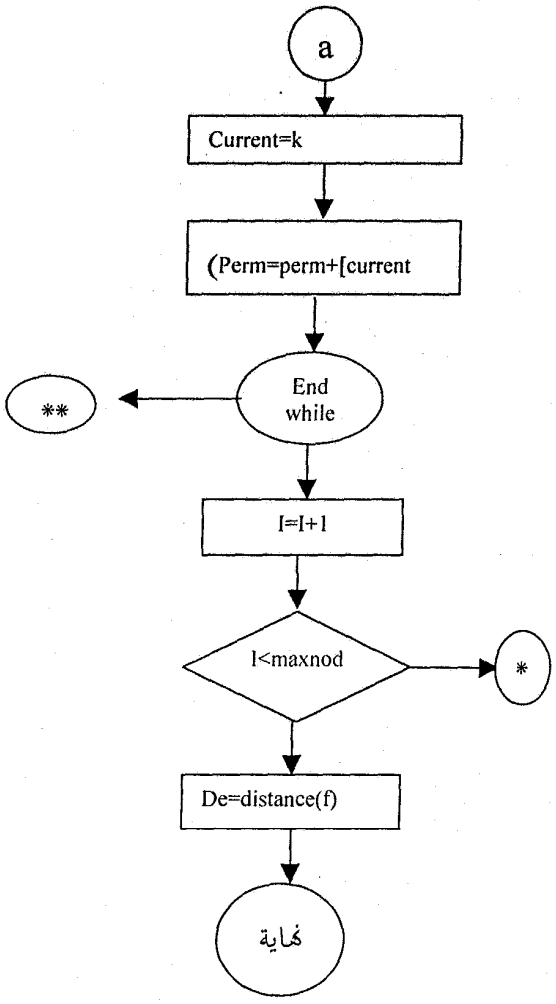
$$G = (V, E, W) \text{ is :}$$

$$\text{Cost function} = \sum_{i=1}^{k-1} w(v_i, v_{i+1})$$

تكون W هي مصفوفة تمثل قيمة المسار V_i إلى V_{i+1} أي قيمة المسار بين عقدتين والتي تمثل دالة الكلفة (Cost function) . ويكون من السهل إيجاد المسار الأقصر من عقدة المصدر إلى جميع العقد الموجودة في المخطط أو عقدة الهدف ، وان خوارزمية (Dijksra) تقوم بعملية جمع قيمة المسار بين عقدة البداية إلى العقدة الحالية والاستمرار في الجمع حتى الوصول إلى العقدة الهدف وباختيار اقل القيم وكما مبين في المخطط (1).

إذ يمثل المسار V_i إلى V_{i+1} هو اقصر مسار واقل كلفة إذا لم يوجد هنالك مسار أخر بين العقدتين يسلك بأقل كلفة من المسار الأول .


بداية
لتكن الحالة البدائية=s
والحالة النهائية=f
Perm=[s) , I=1
*
Distance(I)=?
Current=s
Distance(current)=0
**
While
current<>f
De=distance(current)
Smallist=?
J=1
no
If (J) not in
perm
Sum=de +cost (current ,j)
Sum<distance(j)
yes
Distanc(J)<smallist
Distance(I)=sum
Preced(j)=current
Smallist=distance(j)
K=j
J=J+1
yes
J<maxnod
a



مخطط(1):يوضح خوارزمية Dijksra

2. المسار الأفضل (A*Algorithm) Best-path :

المقصود بالمسار الأفضل الذي يوفر إمكانية التوازن بين قصر الطريق وقيمة اقل عائق عند كل عقدة وهي من الأمور المهمة جدا عند تخطيط مسار الإنسان الآلي ضمن بيئة معينة .وتعتبر تقنية من تقنيات البحث التي توجد الحلول بأقل كلفة وإمكانية الوصول مباشرة إلى الهدف .إذ تعطي إشارة (*) صفة مميزة للتقنية المستعملة

أل_ A* تستعمل القيمة الحدسية(Heuristic) ويمكن ملاحظة العلاقة آلاتية:

H(State)=Cost(State)+Estimate(State)

Cost(State):تمثل الكلفة الحقيقية التي تحقق الوصول للحالة.

Estimate(State):الكلفة التي نحتاجها لإعطاء الحالة حتى الوصول إلى الهدف حيث تضيف الـــ A* الكلفة التي نحتاجها مع الكلفة الحقيقية للوصول إلى الحالة الحالية وبهذا نوجد الحلول بأقل كلفة ممكنة.(1)،(3)،(4).ومن خلال ملاحظة العلاقة التالية .(5),(11):

$G = (C, V, E, W)$ is:

$$\sum_{i=1}^{k-1} w(c_i - v_i, c_{i+1} + v_{i+1})$$

W: W تمثل مصفوفة تمثل قيمة المسار من $C_i - V_i$ إلى $C_{i+1} + V_{i+1}$ بين عقدتين . كما تمثل (C) كلفة المسار في المخطط والـــ (V) تمثل قيمة كل عقدة من عقد المخطط(وزن العائق) .

إن هذه الخوارزمية لا تبحث عن قيم العقد في المخطط ما لم تكون بحاجة لها وهنا نلاحظ بأنها تطرق اقل عدد من العقد للوصول إلى الهدف وتقوم بالاحتفاظ بقيم حتى الوصول إلى الحالة النهائية والتي تمثل الكلفة للمسار (Total cost) ويتم حسابها كالآتي :

$S_1 = (C_1 - Vi)$ ..... (1)

إذا كانت الحالة (Current state) هي نفسها الحالة الابتدائية (initial state) فان قيم C_i و V_i تساوي (صفر) ويكون الناتج :

$S1=(0-0)=0$

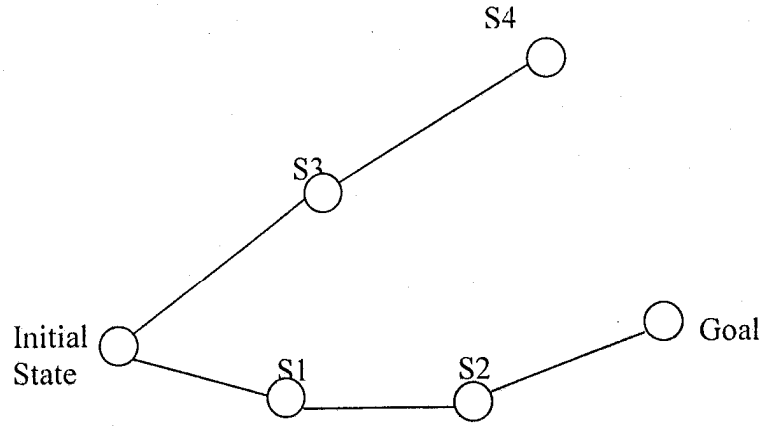
ثم نجد :

$S_2= (C_{i+1}+V_{i+1})$ .....(2)

للحالة القادمة (الجديدة) (next stage). ثم نقوم بعملية الجمع كالآتي :

Total cost $= s_1+s_2$ .....(3)

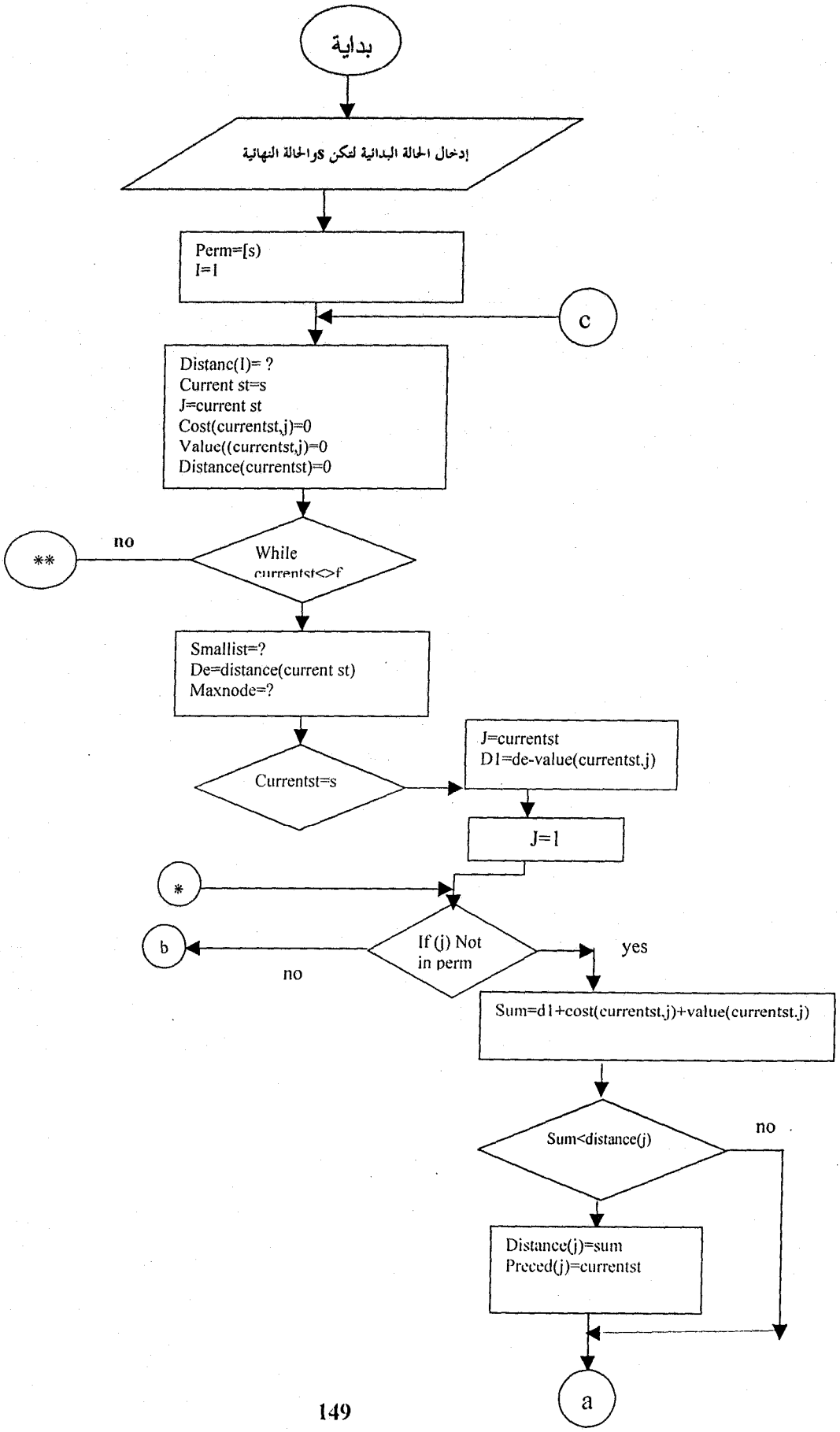
أن S_1 يمثل كلفة المسار الأقصر من عقدة البداية إلى العقد ة n في المعالجة و S_2 تعيد الكلفة الفعلية للمسار الأقصر من العقدة n إلى الهدف وأن الكلفة النهائية (total cost) تمثل الكلفة الفعلية للمسار الأمثل من عقدة البداية ولغاية عقدة الهدف كما مبين في المخطط(2). (4)

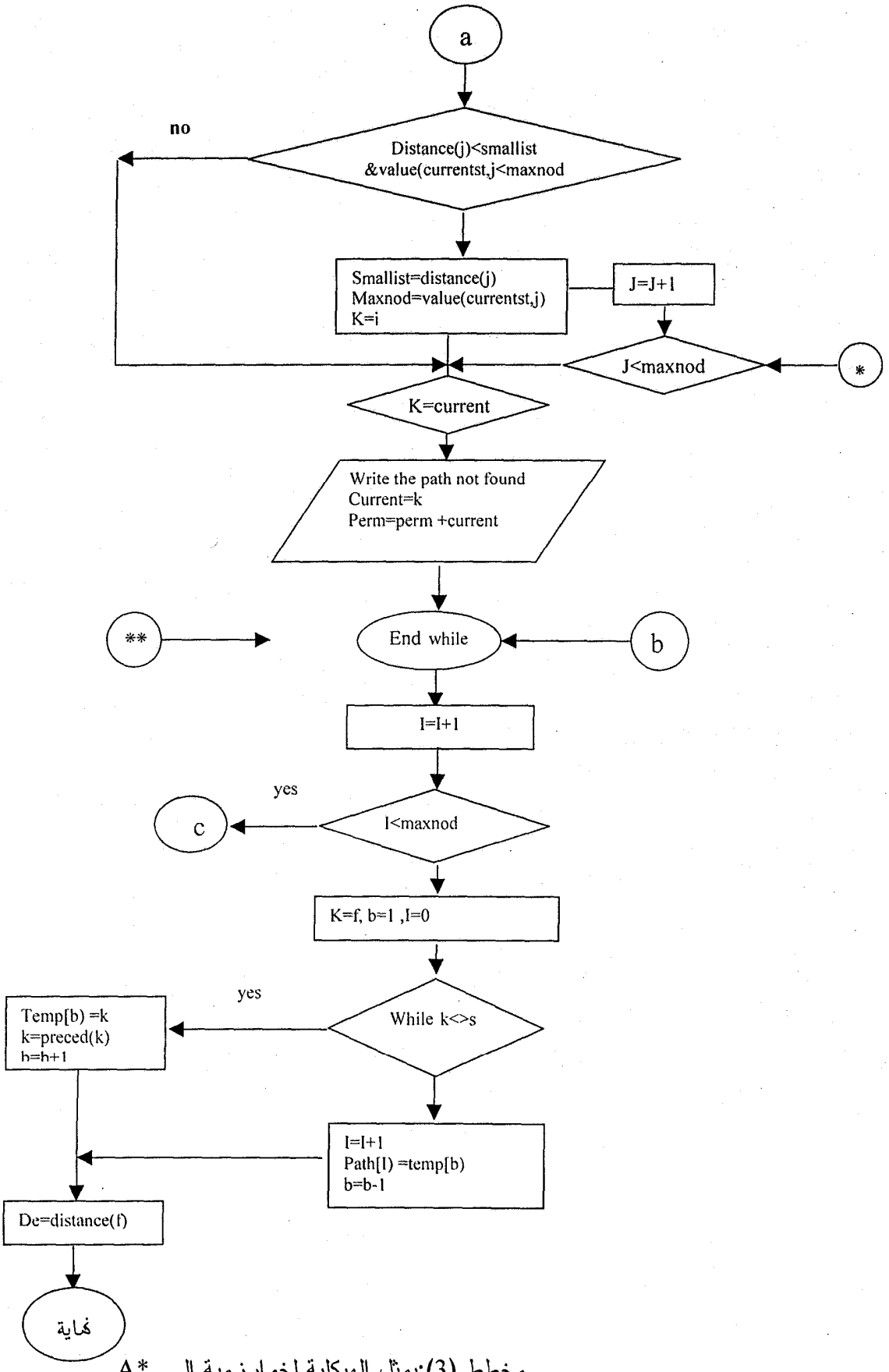


مخطط(2):يمثل الكلفة الفعلية للمسار الأمثل من عقدة البداية ولغاية عقدة الهدف

```
A * Algorithm
Let S is the start state and F is the final state
Perm = [S)
For I ← 1 to maxnode do
Distance (i) ← ?
current ← S
j ← current
Cost (current, j) ← 0
Value (current, j) ← 0
distance (current) ← 0
While current <> F do
        Begin
        Smalldist ← ?
        de ← distance (current)
        maxev ← ?
        If current = S then
                Begin
                        j ← current
                        d_I ← de-value (current, j)
                End
        If current <> S then
                Begin
                        j ← current
                        d_I ← de-value (current, j)
                        End
                For j ← 1 to maxnode do
                        Begin
                                If not (j in perm) then
        Begin
                sum ← d_I+cost (current, j)+value (current, j)
                        If sum <=distance (j) then
                                Begin
                                        distance (j) ← sum
                                Preced (j) ← current
                        End
                                If (distance (j) < smalldist) and
                                (value(current, j) < maxev) then
                                begin
                                        smalldist ← distance (j)
                                        maxve ←value (current, j);
                                        k ← j
```

```
                    end
                    end – if
                    if (k=current) then
                    write (The Path is not Found)
                    current ← k
                    perm ← perm+[current)
              end-while
              k ← f
              b ← 1,        I ← 0
              while k < > s do
                    begin
                          temp[b) ← k
                          k ← preced (k)
                          b=b+1
                    end
              I ← i+1
                    path[i) ←temp[b)
              b ← b-1
              End
        de ← _istance (f)
End-algorithm
```


بداية
إدخال الحالة البدائية لتكن sوالحالة النهائية
Perm=[s)
I=1
c
Distanc(I)= ?
Current st=s
J=current st
Cost(currentst,j)=0
Value((currentst,j)=0
Distance(currentst)=0
**
no
While
currentst<>f
Smallist=?
De=distance(current st)
Maxnode=?
Currentst=s
J=currentst
D1=de-value(currentst,j)
J=1
*
b
no
If (j) Not
in perm
yes
Sum=d1+cost(currentst,j)+value(currentst,j)
Sum<distance(j)
no
Distance(j)=sum
Preced(j)=currentst
a



مخطط (3):يمثل الهيكلية لخوارزمية الـ A*

الجانب العلمي

لقد تم بناء حقيبة برمجية تعتمد الخوارزميتين أعلاه لغرض إجراء التقييم للمسار الأفضل لأي مخطط ، حيث يكون المدخل هو شكل المخطط والناتج مصفوفة عناصرها المسار من عقدة البداية (initial state) وحتى عقدة الهدف(goal state).وفر النظام واجهة مستفيد تتكون من العناصر التالية :

1- Graph creation.
2- Graph displaying.
3- Find path using (Dijksra) shortest path.
4- Find path using (A^*) best path.
5- Exit

1-يمثل الخيار الأول :عملية تكوين المخطط الشجري(Tree Graph) الذي سيتم البحث فيه ويكون الإدخال على شكل مصفوفة متجاورة(adjacency matrix) لخزن معالم المخطط.حيث تكون طريقة إدخال المخطط الشجري على شكل مصفوفة مؤشرات (Array of Pointers) لجميع العقد وتكون صيغة الإدخال على شكل سؤال وجواب وكما يلي :

Q: Numbers of total nodes in the graph:عدد العقد في المخطط الشجري
A: 10: **10** ليكن عدد العقد

Q: Enter First node and it's value:يتم إدخال أول عقدة وقيمتها
A: 1,0

Q: Enter Second node and it's value: يتم إدخال ثاني عقدة وقيمتها
A: 2,11

Q: Enter path cost between these nodes: إدخال المسافة بين العقدتين
A: 6

Q: Enter Third node and it's value: يتم إدخال ثاني عقدة وقيمتها
A: 3,13

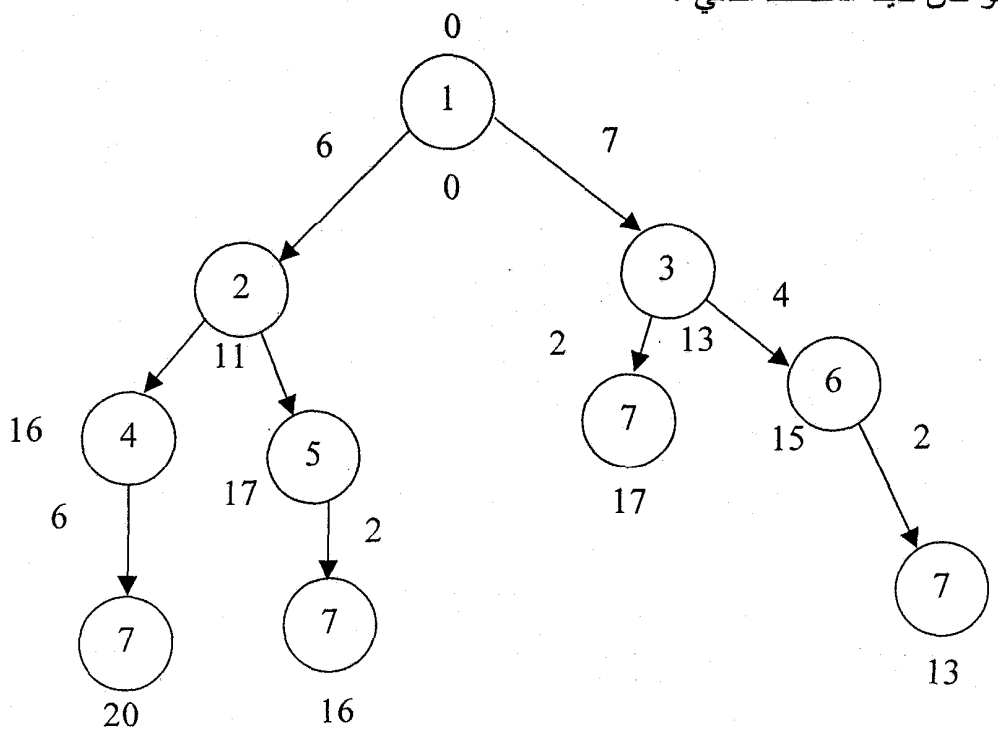
Q: Enter path cost between these nodes: إدخال المسافة بين العقدتين
A: 7

وهكذا نستمر بالإدخال بنفس الأسلوب لكافة العقد في المخطط الشجري والقيم الخاصة بذلك . بعدها يتم السؤال على العقدة الهدف وكما مبين في المثال آلاتي:

Q: Enter Goal node :
A: 7

مثال :

لو كان لدينا المخطط التالي :



2-الخيار الثاني يمثل طريقة لعرض المخطط الذي قمنا بإدخاله على شكل جدول مفهرس لكل مدخل من مداخله يتم خزن طول المسافة أو قيمة (cost) بين عقدتين في المخطط التي قد تكون مسافة أو زمن وغيرها وكذلك يخزن قيمة العقد (value) وهي قيمة معرفة .وكما مبين في الجدول (1).

الجدول (1): يخزن طول المسافة بين عقدتين والقيمة المعرفة (value) للعقد

I-Row		1		2		3		4		5		6		7	
J-Coolum	1	0	10	6	11	7	13	?	16	?	17	?	15	?	17
	2	?	10	0	11	?	13	2	16	4	17	?	15	?	17
	3	?	10	?	11	0	13	?	16	?	17	4	15	?	17
	4	?	10	?	11	?	13	0	16	?	17	?	15	?	17
	5	?	10	?	11	?	13	?	16	0	17	?	15	?	17
	6	?	10	?	11	?	13	?	16	?	17	0	15	?	17
	7	?	10	?	11	?	13	?	16	?	17	?	15	?	17

إن هذه المصفوفة الموزونة تمثل المخطط أعلاه وهي بالأبعاد (7x7) ومداخلها أحد القيم الثلاثة التالية :

1-تكون قيمة المسار تساوي صفر (0) إذا كان الانتقال من العقدة إلى نفسها مع بقاء وزن العقدة فيها قيمة معرفة .

2- إذا لم يكن هنالك مسار بين العقدتين فإن قيمة المسار تمثل بالرمز (?) ، ويتم إعطاء رقم ذات قيمة كبيرة مثلاً (99999) في البرنامج مع بقاء وزن العقدة ثابت .

3- في حالة وجود مسار بين العقدتين فإن قيمة المسار هي وزنه مع بقاء وزن العقدة ثابت.وكما مبين في الجدول (1) .

3-الخيار الثالث يمثل طريقة استخدام خوارزمية (Dijksra) لإيجاد المسار الأقصر مع ملاحظة إن هذه الخوارزمية تهتم بقصر المسافة بغض النظر عن قيمة العائق عند كل عقدة. فعند تطبيق هذا الخيار في النظام نلاحظ أن المسار الأقصر من العقدة (1) إلى العقدة (7) يكون ذات وزن 9 ويحتوي العقد (1,3,7). حيث عند التنفيذ نعطي العقدة البدائية

(initial state) وهي العقدة (1) والعقدة النهائية (final state) وهي العقدة (7) .

من الملاحظ أن هذه الخوارزمية تبحث في جميع عقد المخطط وأوجدت أقصر مسار فعلي بين العقدة البدائية وعقدة الهدف ولكن لم تراعي وزن العائق عند كل عقدة ، وينحصر عمل الخوارزمية على قيمة واحدة فقط تمثل قيمة المسار (cost) ومؤشرة في الحقل الثاني في الجدول (1).

4-الخيار الرابع يمثل طريقة استخدام خوارزمية (A*) لإيجاد افضل مسار وهنا تم الأخذ بنظر الاعتبار وزن العائق لكل عقدة مزاره.فعند تطبيق هذا الخيار من النظام أي تشغيل (A*) فنلاحظ أن المسار الأمثل من العقدة البدائية(1) ولغاية عقدة الهدف (7) يكون بالوزن24 وأن عقد المسار هي (1,3,6,7) .

إن هذه الخوارزمية تعمل بقيمة الحقلين في الجدول المذكور في أعلاه قيمة المسار (cost) ووزن العقدة (value) أي العائق .

من الملاحظ أن هذه الخوارزمية أعطت توازنا بين قصر المسار ووزن أقل عائق وهو الأمر المهم لتخطيط مسار الإنسان الآلي . كذلك تم زيارة العقد التي تحتاجها في تخطيط المسار فقط .

إن الكلفة النهائية :

Total cost = 24

ناتجة من : S1 = 11

و S2 = 13 ناتجة من كلفة (cost=0) ووزن عائق (value=13).

المناقشة والاستنتاجات

1- إن خوارزمية المسار الأفضل (Best path algorithm) قد لا تعطي المسار الأقصر دائماً مقارنة بخوارزمية المسار الأقصر(Shortest path algorithm) ولكنها تهتم بقيمة العائق عند كل عقدة حيث تضع موازنة بين قصر المسار وأقل قيمة للعائق والتي تمثل لنا طريقة جيدة لتخطيط مسار الإنسان الآلي والذي يمثل موضع اهتمام الباحثين .

2- لا تكتفي خوارزمية المسار الأقصر بحل واحد لذا تقوم باختبار جميع العقد في المخطط بينما لا تستخدم الطريقة الأولى إلا العقد التي تحتاجها لبناء المسار وتكتفي بحل أمثل واحد فقط(Optimalsolution).

3- لم يصمم النظام البرمجي لبيئة مسارات ثابتة وإنما يسمح بتحديث المخرجات اعتمادا على طبيعة المخططات المدخلة إلى النظام .

4- وقت تنفيذ البرنامج يتناسب طردياً مع زيادة عدد العقد عندما يكون هنالك مخطط واحد للبيئة ككل بالإضافة إلى التعقيدات الحاصلة أثناء عملية التحديث ، أم عند افتراض المخطط كونه عدد من المخططات الجزئية فإن وقت المعالجة لا يتأثر بعدد العقد ولكن بعدد المداخل .

المصادر

1. Ariel Felner .,‹‹Finding The Shortest Path With A* In An Unknown Physical Environment,, .Department Of Information System Engineering ,Ben - Gurion University of the Negev ,Beer_ sheva.(2004). http://www.cs.biu.ac.il/~felner

2. Freeman J. A. and Skapura D. M.,‹‹Neural Networks: Algorithm Application and Programming,, .New York, AddisonWesley. (1991).

3. Stern R.,‹‹optimal path search in unknown physical Environments,, .MSC Thesis, Department of computer science, Bar_ Ilan University.(2001). http://www.cs.biu.ac.il/~felner.

4. Cognitive Science 108b Lecture Notes. ‹‹Heuristic Search".(2004). http://www.search.yahoo.com/search.

5. Lugar F .George,. ‹‹Artificial Intelligence, Structure and Strategies For Complex Problem, Solving,, . Addison Wesley (1998).

6. Aidepot .‹‹Optimal Search Methods" .Heuristically Informed Methods. http://www.ai-denot.com/Tutorial/Path Finding-Optimal.(2003)

7. Haussler D., AI Learning Algorithm and Valiant Learning Frame Work, Artificial Intelligence, 36, 177-222(1988).

8. Moret B. M. E. and Shapiro H .O ., Algorithm From P to N P, Vol. Redwood City CA. Benjamin / Cummings (1991).

9. نجلاء أكرم الساعاتي . رسالة ماجستير -جامعة الموصل(**1996**).

10. Turing A., ‹‹Logics for Artificial Intelligence,, . ChiChester, Ennis Harwood.(1984).

11. Yuasa and Hagias M., ‹‹Introduction to Common Lisp,, , Boston Academic Press.(1988).