

تصميم خوارزمية بحث استنباطي هجين للوصول الى الهدف بأفضل المسارات

## Design Hybrid Search Algorithm to Reach

## The Goal with the Best Path

م.م انتصار ياسين الخزرجي  
كلية التربية الاساسية / جامعة ديالى

### Abstract خلاصة البحث

تم في هذا البحث بناء خوارزمية بحث استنباطي هجين Hybrid Algorithm مكونة من خوارزميتين ذكيتين هما Hill climbing وخوارزمية Best first Search وهما من خوارزميات البحث الاستنباطي او ما يطلق عليهما احيانا بخوارزميات البحث المعلوماتي Information Search حيث ان اساس عمل هذه الخوارزمية يبدأ بخوارزمية Hill Climbing فأن استطاعت هذه الخوارزمية التحقق من الهدف فيتوقف العمل ويعلن مسار البحث اما اذا ظهرت معلومة غير دقيقة فأنها سوف تبعد المسار عن الهدف وهي الحالة الارجح دائما فيتم استدعاء خوارزمية Best first Search حيث تواصل عملها ابتداء من Root المستوى الاول حيث لا تمر عمليات البحث بالنقاط التي غلقت في خوارزمية Hill climbing الا في حالة كون المستوى يحتوي على نقطة واحدة فقط ومغلقة فأن الخوارزمية مسموح لها بالمرور من تلك النقطة حيث تواصل الخوارزمية بعمليات البحث من اجل الوصول الى الهدف وفي حالة عدم الوصول الى الهدف تستمر الخوارزمية باستدعاء نفسها مرة اخرى حيث لم تمر بالنقاط المغلقة التي غلقتها بنفسها وكذلك النقاط التي غلقتها الـ Hill Climbing عدا النقاط الوحيدة في المستويات التي اشرنا اليها اعلاه حيث تستمر الخوارزمية بعمليات البحث الى ان تصل الى الهدف المطلوب. وبذلك تكون هذه الخوارزمية قد حققت العديد من الاهداف اولها الاستغناء عن تنفيذ خوارزميات البحث الاستنباطي Heuristic Search Algorithms بشكل مستقل اي كل خوارزمية تنفذ على حدة من اجل الوصول الى الهدف اما الهدف الثاني هو وثوقية الوصول الى الهدف اصبحت اكيدة بغض النظر عن طبيعة المعلومات المخمنة فأن كانت تلك المعلومات دقيقة فأن الوصول الى الهدف يكون سريع وبعبسه تستمر الخوارزمية بغلق نقاط فضاء البحث الواحدة تلو الاخرى الى ان تصل الى الهدف وبذلك تتحوا منحى عمل خوارزميات البحث المنتظم من حيث عدد نقاط البحث او اقل منها الى ان تصل الى الهدف المطلوب وبذلك سلكت هذه الخوارزمية سلوك عمل خوارزمية مستقلة وحققت الهدف المطلوب بتوفر معلومات دقيقة وسلكت سلوك هجين بمعلومات دقيقة او غير دقيقة الى ان يتم ايجاد الهدف المطلوب.

## 1- مشكلة البحث

أن مشكلة البحث هي البحث عن أفضل المسارات (Path) في فضاء البحث (State Space) من أجل الوصول إلى نقطة الهدف بأقل كلفة واقصر المسارات واقل وقت ممكن كون المشكلة التي يعمل فيها هذا النظام تمتلك درجة عالية من التعقيد بسبب الحجم الكبير من البيانات وكذلك دقة البيانات المخزنة حيث أن ظهور خطأ في أي نقطة سينحرف مسار البحث ويستمر هذا الانحراف وصولاً إلى آخر نقطة في المستويات الدنيا وبالتالي تضطر هذه الخوارزمية للرجوع إلى نقطة الانطلاق الأولى في عمليات البحث .

## 2- أهمية البحث

أن أهمية البحث تكمن كون مثل تلك الخوارزمية لم يتم التطرق لها في موضوع البحث الاستنباطي فكل الخوارزميات المتوفرة في مصادر الذكاء الاصطناعي تكون أما بحث ضمن خوارزمية (Hill Climbing) أو باستخدام خوارزمية (Best First Search) حيث لم نجد خوارزمية تجمع هاتين الخوارزميتين معا وعليه أي خطأ في البيانات المخزنة سوف تبعد عمليات البحث عن المسار الصحيح وبالتالي لا يمكن الوصول إلى الهدف المطلوب حيث لم يتم إيجاد خوارزمية هجينة كما في خوارزميات البحث المنتظم (Systematic Search Algorithms) . وعليه فأن بناء خوارزمية بحث استنباطي هجين أعطت دعم كبير للبحث الاستنباطي (Heuristic Search Algorithm) وفي نفس الوقت عندما تكون المعلومات المخزنة غير دقيقة فأنها تستطيع اجتياز هذه المشكلة والوصول إلى الهدف المطلوب.

## 3- هدف البحث

إن هدف البحث هو كيفية الوصول إلى النقطة المطلوبة (Goal State) ضمن فضاء البحث (State Space) بغض النظر إن كانت المعلومات المخزنة دقيقة أو يشوبها عدد من الأخطاء وقد تكون هذه الأخطاء قليلة أو قد تكون كبيرة في كلا الحالتين لا بد من الوصول إلى النقطة المطلوب الوصول إليها أو الإجابة عنها وبالتالي فأن هذه الخوارزمية قد حققت الهدف المطلوب فإذا كانت الأخطاء قليلة فهي تبحث ضمن هيكلية خوارزميات البحث الاستنباطي أما في حالة حجم الأخطاء كبيرة فأنها تبدوا وكأنها تبحث ضمن فضاء خوارزميات البحث المنتظم من حيث عدد النقاط التي بحثت عنها ولكنها في حقيقة الأمر أن سلوكها ليس كسلوك خوارزميات البحث المنتظم (Systematic Search Algorithms) لان كلف نقاط البحث هي التي تحدد مسار البحث.

## 4- المقدمة Introduction

إن فضاء البحث في الذكاء الاصطناعي إما أن يكون موجه أو غير موجه فالبحث الموجه يعني وجود نقطة بداية واحدة Start State ونقطة هدف وهي النقطة النهائية التي ينتهي عندها البحث Goal State وهذا ما يطلق عليه بالتمثيل الشجري Tree [1]. أما البحث الغير الموجه فهو يعني وجود أكثر من نقطة بداية Start State وهذا ما يطلق عليه بالمخطط الصوري Graph أما الهدف في البحث الغير موجه فهو هدف محدد ومعلوم كما هو في البحث الموجه. حيث تم التركيز في خوارزميات البحث الذكية العمل في فضاء البحث الموجه لعدة أسباب منها وجود نقطة بداية واحدة ولكي لا تمر الخوارزمية أثناء عمليات البحث في النقطة الواحدة أكثر من مرة وبالتالي يؤثر على سرعة البحث ويمكن أن تدخل الخوارزمية في حلقة تكرار ولم تخرج منها إضافة إلى ذلك وبسبب هذا السلوك فإن مسار البحث يصبح أكثر طولاً وهذا يحتاج إلى مساحة خزن كبيرة ولكون خوارزميات البحث في الذكاء الاصطناعي تركز على سرعة الوصول إلى الهدف وبأقصر مسار ولذلك تم التركيز على البحث الموجه حيث أن فضاء البحث مكون من العديد من نقاط البحث والعديد من المسارات التي تربط هذه النقاط فهناك فضاء خال من المعلومات أي نقاطه ومساراته لا تحتوي على كلف وهذا ما نطلق عليه بالبحث المنتظم Systematic Search ومن أشهر الخوارزميات في هذا النوع من البحث هي خوارزمية البحث العميق Depth Search وخوارزمية البحث الأفقي Breath search وخوارزمية البحث الهجين Hybrid Search. أما فضاء البحث الذي يمتلك معلومات أي الفضاء الذي تكون فيه كل نقطة بحث وكل مسار يحمل كلفة تلك النقطة (State) وكلفة ذلك المسار (path) وهذا ما يطلق عليه بالبحث الاستنباطي Heuristic search ومن أشهر خوارزميات هذا النوع من البحث هما خوارزمية Hill Climbing وخوارزمية Best First search حيث أن كل خوارزمية تعمل بشكل مستقل عن الخوارزمية الأخرى. وإذا ما تمت المقارنة بين البحث المنتظم والبحث الاستنباطي نجد أن سرعة الوصول إلى الهدف في البحث الاستنباطي أسرع بكثير من البحث المنتظم وأن الذاكرة التي نحتاجها في البحث الاستنباطي هي أصغر مما نحتاجها في البحث المنتظم أما المسار الذي نصل به إلى الهدف في البحث الاستنباطي فهو أقصر مما هو في البحث المنتظم. أما وثوقية الوصول إلى الهدف فهو يتحقق بكل تأكيد في البحث المنتظم كون خوارزميات البحث المنتظم لم تستثن أي State أثناء عمليات البحث أما عمليات البحث في خوارزميات البحث الاستنباطي فهي تستثن الكثير كونها تجري مقارنة بين كلف States وتأخذ الكلفة الأقل عند العمل بمفهوم Local Minimum. وعندما تكون أي State كلفتها غير دقيقة وكانت ضمن المسار فإن تلك State قد تحرف

المسار عن الهدف وبالتالي لم يتم الوصول الى الهدف وهذا ما يحصل عند خوارزمية Hill Climbing حيث لم اجد بين تلك الخوارزميات من يعالج تلك الاخطاء وكان لابد من صياغة مفهوم يربط هاتين الخوارزميتين معا ومعالجة النقاط التي تبعد المسار عن الهدف من خلال غلق تلك النقاط وعدم المرور بها من خلال استدعاء Best First search اكثر من مرة في حالة وجود كلف خاطئة لعدد من الـ States وبالتالي تتمكن من الوصول الى الهدف المطلوب فالبحث بهذه الطريقة هو بكل تأكيد اسرع بكثير من خوارزميات البحث المنتظم بالوصول الى الهدف [2].

## 5- خوارزميات البحث الذكية [3] Intelligent Search Algorithms

هنالك العديد من خوارزميات البحث في الذكاء الاصطناعي حيث ان طبيعة فضاء البحث هو الذي يحدد نوع الخوارزمية في البحث عن الهدف فعندما يكون فضاء البحث ممثل بمخطط شجري لا يحتوي على كلف فيتم اختيار احد خوارزميات البحث المنتظم Systematic Search algorithms اما اذا كان المخطط الشجري يحتوي على كلف فيتم اختيار احد خوارزميات البحث الاستباطي Heuristic Search algorithms وبذلك تكون خوارزميات كلا البحثين كالآتي:-

### 1-Systematic Search Algorithm

a- Depth Search Algorithm

b- Breath Search Algorithm

c- Hybrid Search Algorithm

### 2- Heuristic Search Algorithm

a- Hill Climbing Search Algorithm

b- Best First Search Algorithm

## 5-1- خوارزميات البحث المنتظم Systematic Search Algorithm

### 5-1-1 خوارزمية البحث العميق Depth Search Algorithm

نلاحظ ان خوارزميات البحث المنتظم Systematic Search Algorithm توجد علاقة مشتركة فيما بينها حيث ان كل خوارزمية في فضاء البحث تقوم بتدقيق كافة States المارة بها حتى يتم العثور على الهدف حيث ان الوصول الى الهدف في هذه الخوارزميات اكيد فلو تتبعنا عمل خوارزمية البحث العميق وتتبعنا خطوات هذه الخوارزمية من خلال الخطوات التالية نجدها اسرع خوارزميات البحث المنتظم :-

Procedure depth-first search;

Begin

Open:= [start ];

Close:= [ ];

While open  $\neq$  [ ] do

Begin

Remove leftmost state from open, call is X;

If X is a goal then return ( success)

Else begin

Generate children of X;

Put X on closed;

Eliminate children of X on open or closed;

Put remaining children on right end of open

End

End;

return (failure)

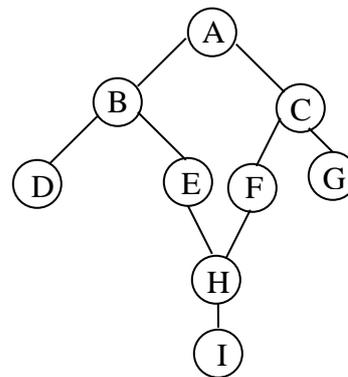
end.

نلاحظ ان هذه الخوارزمية تستمر بالنزول الى العمق باحثا عن الهدف المطلوب وفي

حالة عدم العثور على الهدف تقوم الخوارزمية بتدقيق كافة الـ States والتي تكون في حالة

open الى ان تصل الى الهدف.

ويمكن توضيح عمل هذه الخوارزمية من خلال المثال التالي:



1- Open= [A]; closed=[ ]

2- Open= [B,C]; closed=[A ]

3- Open= [D,E,C]; closed=[A,B]

4- Open= [E,C]; closed=[A,B,D]

- 5- Open= [H,C]; closed=[A,B,D,E]  
 6- Open= [I,C]; closed=[A,B,D,E,H]

نلاحظ ان مسار الوصول الى الهدف في هذه الخوارزمية كالاتي:-

Path=~~A~~ → ~~B~~ → ~~C~~ → A

وبذلك تكون الخوارزمية قد سلكت المسار ووصلت الى الهدف بأقصر المسارات على الرغم من وجود مسار اخر يمكن الوصول به الى الهدف ولكن طبيعة هذه الخوارزمية كونها تتحرك باتجاه العمق لذلك وصلت الى الهدف المطلوب باستخدام المسار اعلاه ولم تسلك المسار الثاني والمبين كالاتي:-

Path=~~A~~ → ~~B~~ → ~~C~~ → ~~D~~ → A

#### 5-1-2 خوارزمية البحث الافقي [4] Breath Search Algorithm

تعمل هذه الخوارزمية بشكل افقي اي يستمر تدقيق مستوى البحث بشكل كامل ثم الانتقال الى المستوى اللاحق الى ان يتم ايجاد الهدف حيث يتضح عمل هذه الخوارزمية من خلال تتبع خطوات الخوارزمية التالية:-

Procedure breath-first search;

Begin

Open:= [start ];

Close:= [ ];

While open ≠ [ ] do

Begin

Remove leftmost state from open, call is X;

If X is a goal then return ( success)

Else begin

Generate children of X;

Put X on closed;

Eliminate children of X on open or closed;

Put remaining children on right end of open

End

End;

return (failure)

end.

ولتوضيح عمل هذه الخوارزمية بشكل عملي نأخذ نفس فضاء البحث والممثل في خوارزمية

-: Depth Search

- 1- Open= [A]; closed=[ ]
- 2- Open= [B,C]; closed=[A ]
- 3- Open= [C,D,E]; closed=[A,B]
- 4- Open= [D,E,F,G]; closed=[A,B,C]
- 5- Open= [H,F,G]; closed=[A,B,C,D]
- 6- Open= [F,G,H]; closed=[A,B,C,D,E]
- 7- Open= [G,H]; closed=[A,B,C,D,E,F]
- 8- Open= [H]; closed=[A,B,C,D,E,F,G]
- 9- Open= [I]; closed=[A,B,C,D,E,F,G,H]

وبذلك تم العثور على الهدف I حيث نلاحظ ان هذه الخوارزمية قد دقتت كافة الـ States في فضاء البحث الى ان وجدت البحث عبر المسار التالي:-

Path=> A

حيث نلاحظ ان عدد الـ States التي تم تدقيقها في خوارزمية البحث الافقي هي اكثر مما هو عليه في خوارزمية البحث العميق.

### 3-1-5 خوارزمية البحث المنتظم الهجين Hybrid Search Algorithm

هذه الخوارزمية هي خليط من العمل المنتظم المتمثل بخوارزمية البحث العميق Depth Search Algorithm وخوارزمية البحث الافقي Breath Search Algorithm حيث يتم استخدام هذا النوع من الخوارزميات عندما يكون فضاء البحث كبير جدا اضافة الى وجود ترابط بين عدد من States من جهة اليمين الى اليسار او بالعكس في فضاء البحث حيث تعمل خوارزمية البحث العميق عندما تكون الـ States منتظمة او شبه منتظمة اما في حالة ظهور حالة انتقال او وجود حالة ربط او وجود انتقال سريعة في مستوى معين الى مستوى بعيد في العمق فيفضل استخدام البحث الافقي من اجل الوصول الى تلك العقد الناقلة لكي يأتي دور خوارزمية البحث العميق في تكملة خطوات المسار باتجاه الهدف.

### 2-5 خوارزميات البحث الاستنباطي Heuristic Search Algorithm [5]

هذا النوع من الخوارزميات تعمل في فضاء بحث معلوماتي اي كل State لها كلفة انتقال نحو الهدف وكذلك لكل مسار له كلفة انتقال بين State واخرى حيث ان هذا النوع من البحث يمتاز بالسرعة كونه لا يمر بكل نقاط البحث وانما فقط النقاط التي توجه مسار الهدف ومن اشهر هذه الخوارزميات:-

### 1-2-5 خوارزمية التلال Hill climbing

تعتبر هذه الخوارزمية من اسرع الخوارزميات الموجودة كونها تدقق مع الـ States التي تمتلك اقل كلفة وتهمل كل الـ State التي تم فتحها ثم تنتقل الى مستوى جديد الى ان تصل الى الهدف حيث يتوضح عمل هذه الخوارزمية من خلال الخطوات التالية:-

- 1- Generate the first proposed solution in the same way as would be done in the generate-and-test procedure. See if it is a solution. If so, quit.  
Else continue.
- 2- From this solution apply some number of applicable rules to generate a new set of proposed solutions.
- 3- From each element of the set do the following:-
  - a- send it to the test function , if it is a solution , quit.
  - b- if not, see if it is the closes to a solution of any of elements tested so far. If it is remember it. If it is not, forget it.
- 4- Take the best element found above and use it as the next proposed solution. This step corresponds to a move through the problem space in the direction that appears to be leading the most quickly toward a goal 3.
- 5- Go back to step 2.

### 2-2-5 خوارزمية أفضل أول بحث Best- First- Search

تعمل هذه الخوارزمية في فضاء بحث معلومات اي لكل State لها كلفة خاصة تم حسابها مسبقا وكذلك لكل مسار يتصل بها له كلفة خاصة به اصل عمل هذه الخوارزمية لا تهمل اي نقطة تم فتحها ولم تغلق الى ان تصل الى الهدف النهائي او الى نهاية البحث حيث ان حركة هذه الخوارزمية غير منتظمة يتحكم بها كلف الـ States حيث تنتقل عمليات البحث من الاعلى الى الاسفل او بالعكس اعتمادا على الكلف ويمكن الاطلاع على سير عمل هذه الخوارزمية من خلال الخطوات التالية:-

Procedure best-first search;

Begin

Open:= [start ];

Close:= [ ];

While open  $\neq$  [ ] do

Begin

Remove the leftmost state from open, call is X;

If X = goal then return the path from start to X;

Else begin

Generate children of X;

For each child of X do

Case

The child is not on open or closed;

Begin

Assign the child a heuristic value;

Add the child to open

End;

The child is already on open:

If the child was reached by a shorter path

Then give the state on open the shorter path

The child is already on closed:

If the child was reached by a shorter path then

Begin

Remove the state from closed;

Add the child to open

End;

End;

Put X on closed;

Re-order states on open by heuristic merit (best leftmost)

End;

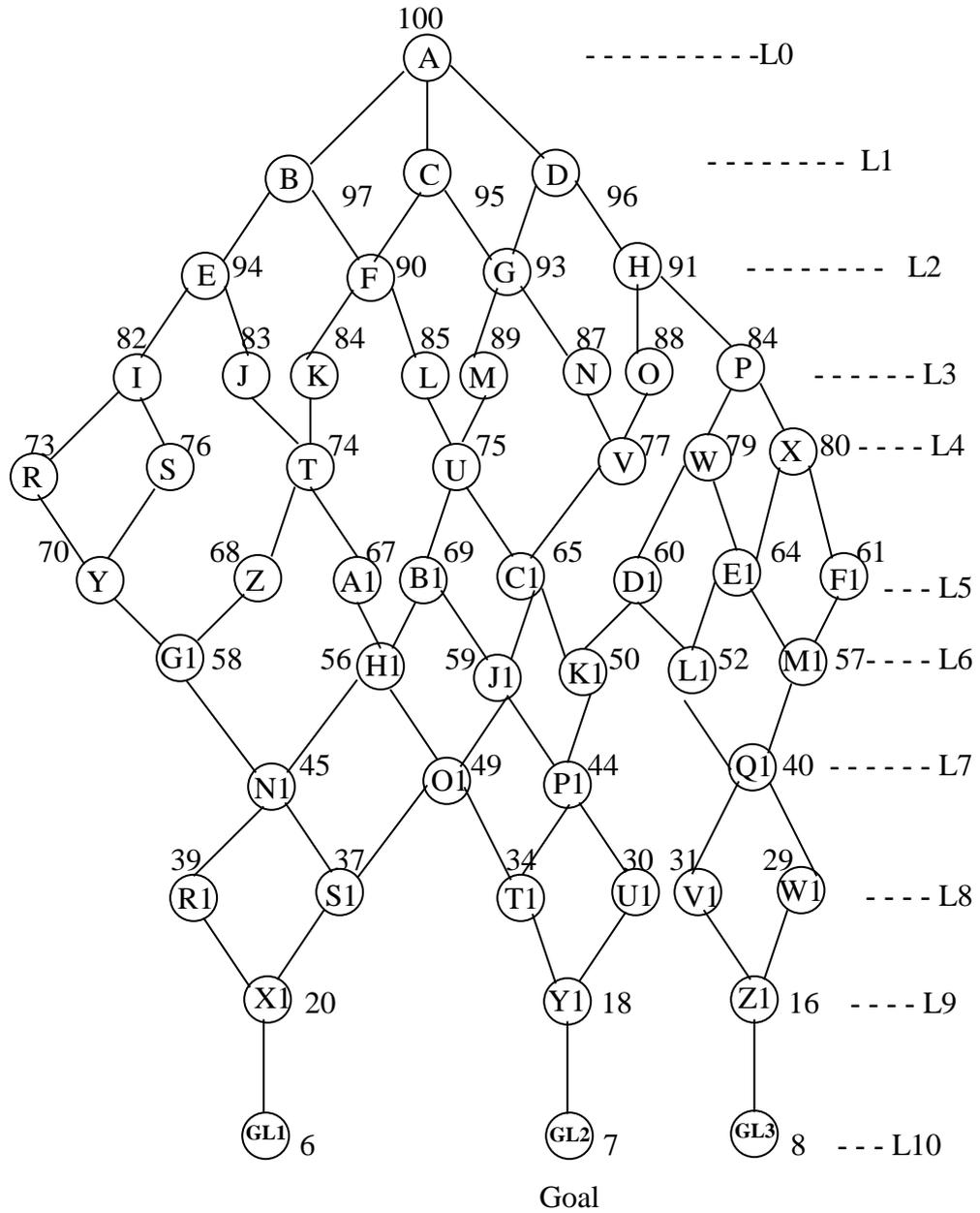
return (failure)

end.

### 3-2-5 الخوارزمية المقترحة للبحث الاستنباطي الهجين

#### Hybrid Heuristic Search Algorithm

ولتوضيح عمل خوارزمية البحث الاستنباطي الهجين نأخذ المثال التالي لشجرة لها نقطة بداية واحدة (Start State) ولها اكثر من نقطة نهاية (Goal State) حيث ان فضاء البحث مكون من (53) نقطة بحيث تؤلف هذه النقاط مجتمعة فضاء البحث الكلي ومن بين تلك النقاط نقطة الهدف Goal State كما موضحة في الشكل رقم ( 1 ):-



شكل رقم ( 1 )

حيث نلاحظ من الشكل اعلاه ان الكلف الفعلية لكل State قد تم حسابها حيث ان هذه

الكلف تمثل الكلفة الفعلية لافضل مسار من نقطة البداية الى نقطة الهدف حيث تم حسابها

$$E^*_F(N) = g^*(n) + h^*(n) \quad . \text{ باستخدام دالة التخمين [6] .}$$

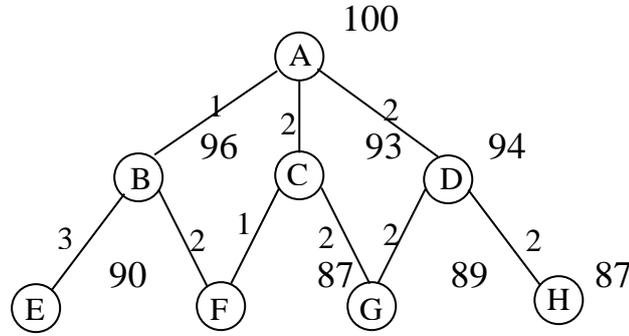
حيث  $g^*(n)$  هي كلفة اختبار اقصر مسار من نقطة البداية (root) الى النقطة (n) حيث ان n

تمثل الـ State المطلوب حساب كلفتها اما بالنسبة  $h^*(n)$  فهي تمثل الكلفة الفعلية لاختيار اقصر

مسار من النقطة (n) الى نقطة الهدف.

وبجمع هاتين الكلفتين نكون قد حصلنا او خمنا الكلفة الفعلية لافضل مسار من نقطة البداية الى نقطة الهدف.

ولتوضيح ما ورد اعلاه لناخذ مقطعا افقيا للمخطط اعلاه ولتكن لثلاثة مستويات وكما مبينة بالشكل التالي:-



وبتطبيق دالة التخمين لحساب الكلف الفعلية لكل State :-

$$E^*_F(N) = g^*(n) + h^*(n)$$

$$\begin{aligned} E^*_F(A) &= g^*(A) + h^*(A) \\ &= 0 + 100 \\ &= 100 \end{aligned}$$

$$\begin{aligned} E^*_F(B) &= g^*(B) + h^*(B) \\ &= 1 + 96 \\ &= 97 \end{aligned}$$

$$\begin{aligned} E^*_F(C) &= g^*(C) + h^*(C) \\ &= 2 + 93 \\ &= 95 \end{aligned}$$

$$\begin{aligned} E^*_F(D) &= g^*(D) + h^*(D) \\ &= 2 + 94 \\ &= 96 \end{aligned}$$

$$\begin{aligned} E^*_F(E) &= g^*(E) + h^*(E) \\ &= (1 + 3) + 90 \\ &= 94 \end{aligned}$$

$$\begin{aligned} E^*_F(F) &= g^*(F) + h^*(F) \\ &= (1 + 2) + 87 \\ &= 90 \end{aligned}$$

هنالك مساران يتصلان بهذه الـ State ولكون المساران لهما نفس كلفة الانتقال وهي (1 + 2) وعليه نختار واحدة

$$E_F^*(G) = g^*(G) + h^*(G)$$

$$= (2 + 2) + 89$$

$$= 93$$

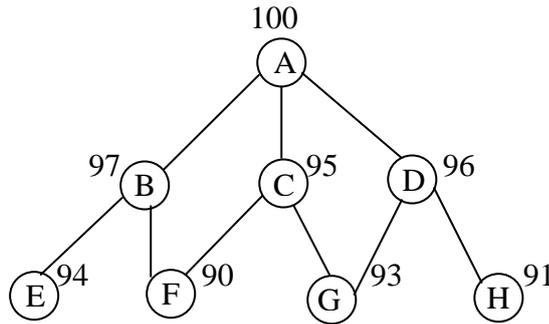
$$E_F^*(H) = g^*(H) + h^*(H)$$

$$= (2 + 2) + 87$$

$$= 91$$

حيث نحصل على المخطط التالي والذي يحتوي على كلف مطابقة للمستويات الثلاثة

التي تم اختيارها لغرض حسابها.



وبنفس الطريقة تستمر عملية حساب الكلف الفعلية لكافة الـ State في الشكل رقم (1).  
لا بد من الإشارة هنا في حالة اتصال State بأكثر من مسار فيجب حساب كافة الكلف وتثبيت  
أعلى الـ State أما في حالة التشابه فتأخذ قيمة واحدة فقط. أما كيف يتم التعامل مع الـ State  
التي لها أكثر من كلفة فعلية فيتم اختيار الكلفة الصغيرة أولاً وإذا ابعدتنا هذه الكلفة عن المسار  
الصحيح فيتم إهمالها وأخذ الكلفة التي تليها.  
وبعد أن أصبح لدينا مخطط كل State فيه قد تم حساب الكلف الفعلية لها نستطيع من  
خلال هذه الـ States الانتقال من مستوى إلى آخر وصولاً إلى الهدف المطلوب.

## 6- النتائج Results

لنأخذ على سبيل المثال الهدف (GL2) نجد أن عدد المسارات بين هذا الهدف ونقطة

البداية Start State هو (56) مساراً وهي كما يلي:-

- Path1=GL2 > Y1 > T1 > O1 > H1 > A1 > T > J > E > B > A
- Path2=GL2 > Y1 > T1 > O1 > H1 > A1 > T > K > F > B > A
- Path3=GL2 > Y1 > T1 > O1 > H1 > A1 > T > K > F > C > A
- Path4=GL2 > Y1 > T1 > O1 > H1 > B1 > U > L > F > B > A
- Path5=GL2 > Y1 > T1 > O1 > H1 > B1 > U > L > F > C > A
- Path6=GL2 > Y1 > T1 > O1 > H1 > B1 > U > M > G > C > A
- Path7=GL2 > Y1 > T1 > O1 > H1 > B1 > U > M > G > D > A
- Path8=GL2 > Y1 > T1 > O1 > J1 > B1 > U > L > F > B > A



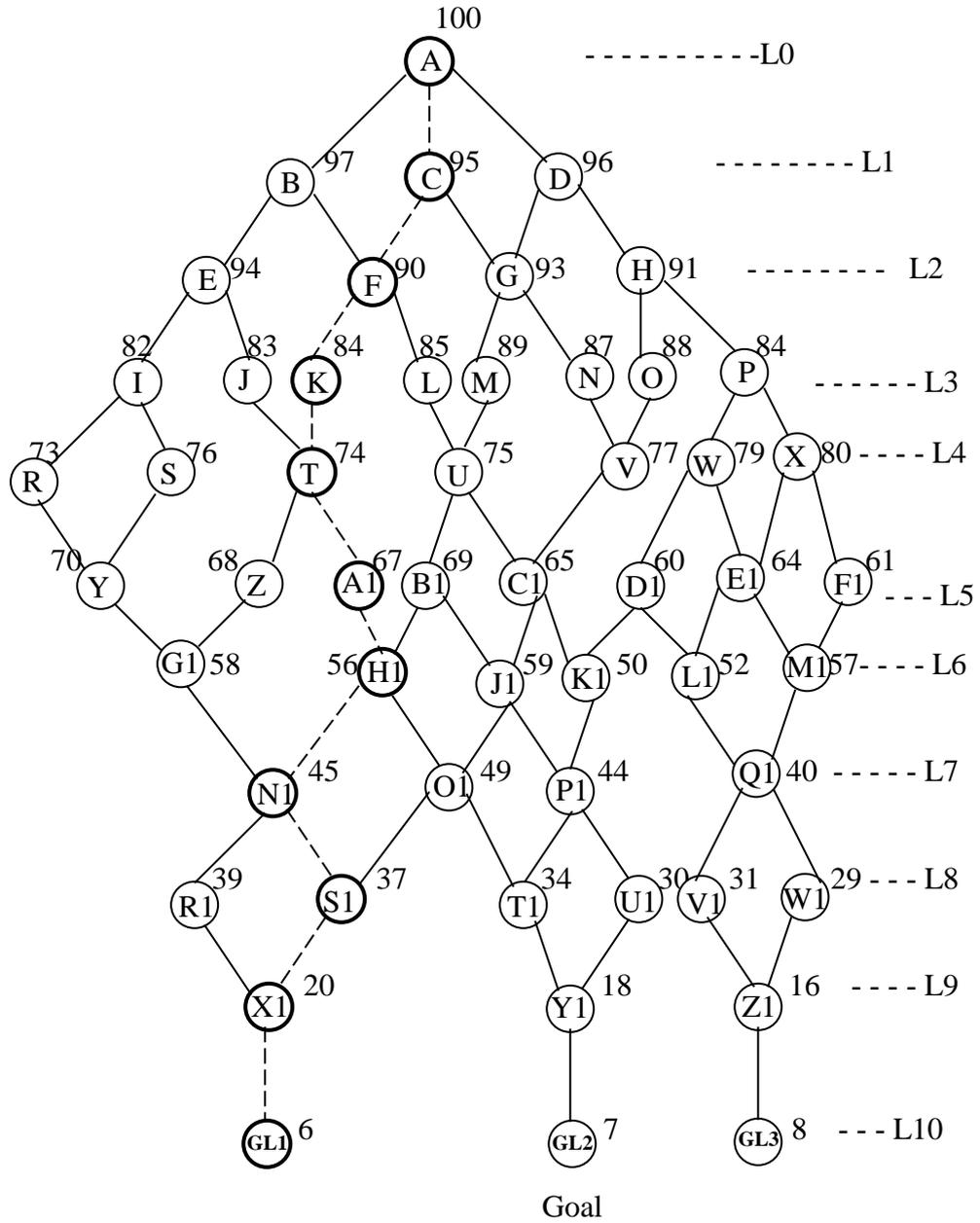
Path52=GL2> Y1> U1> P1> K1> C1> U> M> G> D> A  
 Path53=GL2> Y1> U1> P1> K1> C1> V> N> G> C> A  
 Path54=GL2> Y1> U1> P1> K1> C1> V> N> G> D> A  
 Path55=GL2> Y1> U1> P1> K1> C1> V> O> H> D> A  
 Path56=GL2> Y1> U1> P1> K1> D1> W> P> H> D> A

ولضمان الوصول الى الهدف المطلوب وبأفضل المسارات نقوم بتنفيذ خوارزمية البحث الاستباطي الهجين حيث يبدأ العمل بخوارزمية Hill Climbing باستخدام مفهوم Open-Close اي اجراء عملية مطابقة الهدف مع المتولد من كل state وغلق تلك الـ State التي تم الانطلاق منها الى الـ States الجديدة وعادةً ما يطلق على المتولد بـ (Childs) [7].

#1 open ( A<sub>100</sub> )  
     Close ( [ ] )  
 # 2 open ( C<sub>95</sub>,D<sub>96</sub>,B<sub>97</sub> )  
     Close ( A<sub>100</sub> )  
 # 3 open ( F<sub>90</sub>,G<sub>93</sub> )  
     Close ( A<sub>100</sub>,C<sub>95</sub> )  
 # 4 open ( K<sub>84</sub>,L<sub>85</sub> )  
     Close ( A<sub>100</sub>,C<sub>95</sub>, F<sub>90</sub> )  
 # 5 open ( T<sub>74</sub> )  
     Close ( A<sub>100</sub>,C<sub>95</sub>, F<sub>90</sub>, K<sub>84</sub> )  
 # 6 open ( A<sub>167</sub>,Z<sub>68</sub> )  
     Close ( A<sub>100</sub>,C<sub>95</sub>, F<sub>90</sub>, K<sub>84</sub>, T<sub>74</sub> )  
 # 7 open ( H<sub>156</sub> )  
     Close ( A<sub>100</sub>,C<sub>95</sub>, F<sub>90</sub>, K<sub>84</sub>, T<sub>74</sub>, A<sub>167</sub> )  
 # 8 open ( N<sub>145</sub>,O<sub>149</sub> )  
     Close ( A<sub>100</sub>,C<sub>95</sub>, F<sub>90</sub>, K<sub>84</sub>, T<sub>74</sub>, A<sub>167</sub>, H<sub>156</sub> )  
 # 9 open ( S<sub>137</sub>,R<sub>139</sub> )  
     Close ( A<sub>100</sub>,C<sub>95</sub>, F<sub>90</sub>, K<sub>84</sub>, T<sub>74</sub>, A<sub>167</sub>, H<sub>156</sub>, N<sub>145</sub> )  
 # 10 open ( X<sub>120</sub> )  
     Close ( A<sub>100</sub>,C<sub>95</sub>, F<sub>90</sub>, K<sub>84</sub>, T<sub>74</sub>, A<sub>167</sub>, H<sub>156</sub>, N<sub>145</sub>, S<sub>137</sub> )  
 # 11 open ( GL<sub>16</sub> )  
     Close ( A<sub>100</sub>,C<sub>95</sub>, F<sub>90</sub>, K<sub>84</sub>, T<sub>74</sub>, A<sub>167</sub>, H<sub>156</sub>, N<sub>145</sub>, S<sub>137</sub>, X<sub>120</sub> )

وبذلك تكون هذه الخوارزمية في هذا المثال لم تصل الى الهدف المطلوب (GL2) بل ابتعدت عنه بعد ان سلكت مسار اخر حيث انحرف المسار لظهور State ذات كلفة اقل وهي N<sub>145</sub> حيث غيرت اتجاه البحث الى نهاية ليست هي الهدف المطلوب فلو كانت كلفة هذه الـ State بدل كلفة الـ State (O<sub>149</sub>) لأصبح اتجاه المسار صحيح ووصلت الى الهدف المطلوب GL2 وهذا ما تم التأكيد عليه حيث ان مساوى هذه الخوارزمية هو ظهور كلفة صغيرة لـ

state ما تبعتها عن المسار الصحيح وبالتالي لم تصل الى الهدف المطلوب بل وصلت الى احد  
 نهائيات فضاء البحث وبالتحديد (GL2) كما هو موضح بالمسار المنقط التالي في الشكل رقم  
 ( 2 ) :-



الشكل رقم ( 2 )

حيث ان المسار الذي سلكته الخوارزمية هو كما يلي:-

Path= GL2 → X1 → S1 → N1 → H1 → A1 → T → K → F → C → A

ولتفادي ما حدث في خوارزمية Hill-Climbing لابد من تفعيل خوارزمية Best-First-Search وهي احدى خوارزميات البحث الاستنباطي ايضاً ولكن طبيعة عمل هذه الخوارزمية لا تهمل اي State تم فتحها في عمليات سابقة او لاحقة حيث تقوم بأجراء المقارنة وترتيبها تصاعديا حسب الكلف واخذ الـ State ذات الكلفة الاقل لغرض الانطلاق الى مستوى لاحق حيث لا تتعامل هذه الخوارزمية مع الـ States التي تم غلقها في الخوارزمية السابقة Hill-Climbing الا اذا كانت State واحدة مرتبطة بـ root مستقلة حيث تعمل هذه الخوارزمية Best-First-Search بالشكل التالي حيث تنطلق من الـ (A<sub>100</sub>) State كون المستوى الاول يحتوي على State واحدة وتم غلقها وهي A<sub>100</sub> ثم تدقق كافة الـ States في المستوى الثاني حيث تجد هذه الخوارزمية ان اقل الكلف للـ State هي D<sub>96</sub> كون الـ State ذو الكلفة 95 وهي C<sub>95</sub> قد تم غلقها في لخوارزمية السابقة وبذلك تنطلق هذه الخوارزمية من الـ State D<sub>96</sub> وصولا الى الهدف المطلوب (GL2) وكما مبينة في الخطوات التالية:-

#1 open ( A<sub>100</sub>)

Close ( [ ] )

# 2 open (D<sub>96</sub>,B<sub>97</sub>)

Close ( A<sub>100</sub>)

# 3 open (H<sub>91</sub>,G<sub>93</sub>,B<sub>97</sub>)

Close ( A<sub>100</sub>,D<sub>96</sub>)

# 4 open (P<sub>84</sub>,O<sub>88</sub>,G<sub>93</sub>,B<sub>97</sub>)

Close ( A<sub>100</sub>,D<sub>96</sub>, H<sub>91</sub>)

# 5 open (W<sub>79</sub>,X<sub>80</sub>,O<sub>88</sub>,G<sub>93</sub>,B<sub>97</sub>)

Close ( A<sub>100</sub>,D<sub>96</sub>, H<sub>91</sub>, P<sub>84</sub>)

# 6 open (D<sub>160</sub>,E<sub>164</sub>,X<sub>80</sub>,O<sub>88</sub>,G<sub>93</sub>,B<sub>97</sub>)

Close ( A<sub>100</sub>,D<sub>96</sub>, H<sub>91</sub>, P<sub>84</sub>,W<sub>79</sub>)

# 7 open (K<sub>150</sub>,L<sub>152</sub>,E<sub>164</sub>,X<sub>80</sub>,O<sub>88</sub>,G<sub>93</sub>,B<sub>97</sub>)

Close ( A<sub>100</sub>,D<sub>96</sub>, H<sub>91</sub>, P<sub>84</sub>,W<sub>79</sub>,D<sub>160</sub>)

# 8 open (P<sub>144</sub>,L<sub>152</sub>,E<sub>164</sub>,X<sub>80</sub>,O<sub>88</sub>,G<sub>93</sub>,B<sub>97</sub>)

Close ( A<sub>100</sub>,D<sub>96</sub>, H<sub>91</sub>, P<sub>84</sub>,W<sub>79</sub>,D<sub>160</sub>,K<sub>150</sub>)

# 9 open (U1<sub>30</sub>,T1<sub>34</sub>,L1<sub>52</sub>,E1<sub>64</sub>,X<sub>80</sub>,O<sub>88</sub>,G<sub>93</sub>,B<sub>97</sub>)

Close ( A<sub>100</sub>,D<sub>96</sub>, H<sub>91</sub>, P<sub>84</sub>,W<sub>79</sub>,D1<sub>60</sub>,K1<sub>50</sub>,P1<sub>44</sub>)

# 10 open (Y1<sub>18</sub>,T1<sub>34</sub>,L1<sub>52</sub>,E1<sub>64</sub>,X<sub>80</sub>,O<sub>88</sub>,G<sub>93</sub>,B<sub>97</sub>)

Close ( A<sub>100</sub>,D<sub>96</sub>, H<sub>91</sub>, P<sub>84</sub>,W<sub>79</sub>,D1<sub>60</sub>,K1<sub>50</sub>,P1<sub>44</sub>,U1<sub>30</sub>)

# 11 open (GL2<sub>7</sub>,T1<sub>34</sub>,L1<sub>52</sub>,E1<sub>64</sub>,X<sub>80</sub>,O<sub>88</sub>,G<sub>93</sub>,B<sub>97</sub>)

Close ( A<sub>100</sub>,D<sub>96</sub>, H<sub>91</sub>, P<sub>84</sub>,W<sub>79</sub>,D1<sub>60</sub>,K1<sub>50</sub>,P1<sub>44</sub>,U1<sub>30</sub>,Y1<sub>18</sub>)

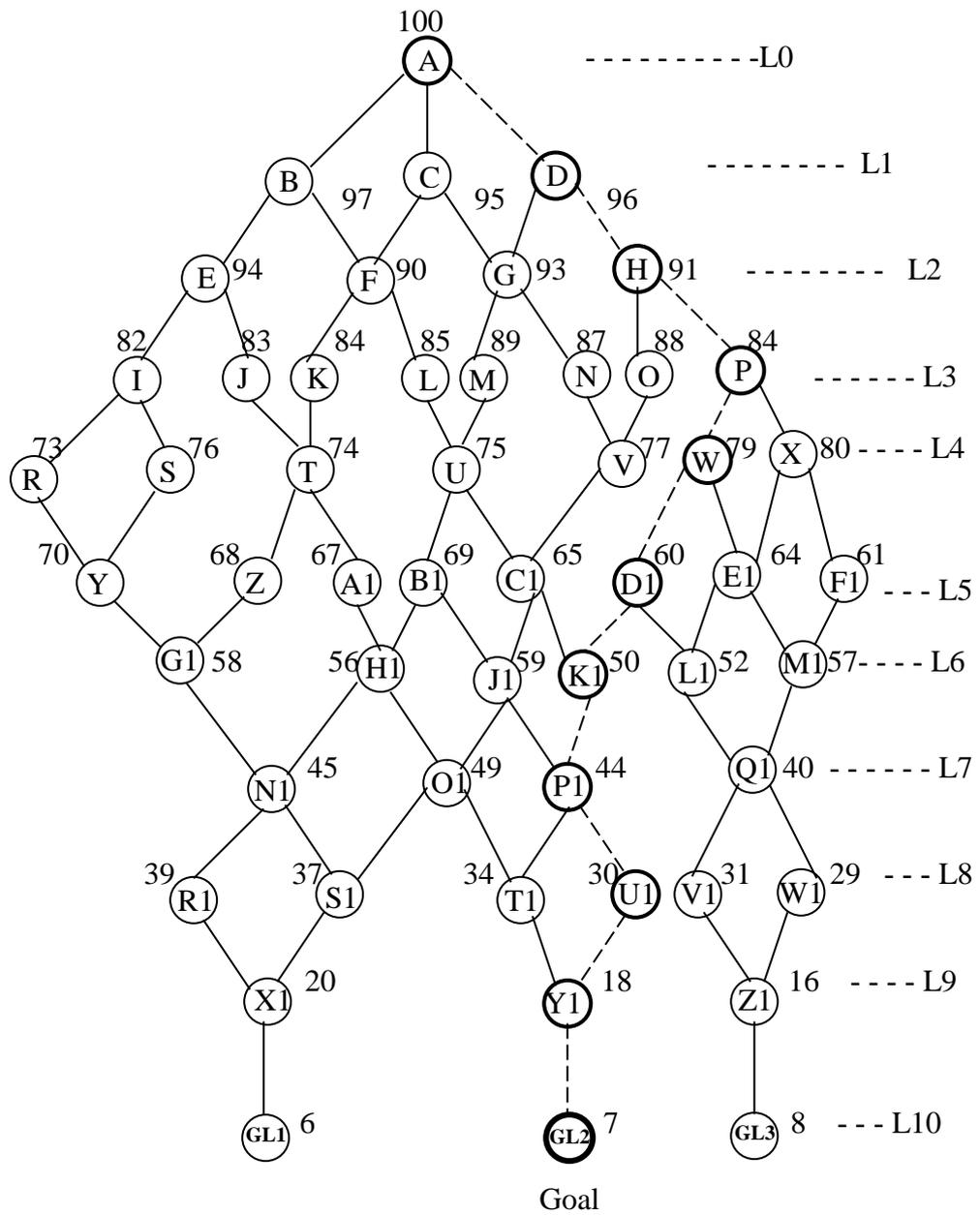
وبذلك تحقق الهدف المطلوب وهو GL2<sub>7</sub>

نلاحظ ان عمل هاتين الخوارزميتين معا ضمن خوارزمية واحدة قد حققت الهدف المطلوب حيث عملت الخوارزمية الاولى Hill-Climbing على تدقيق احد المسارات من اجل ايجاد الهدف المطلوب ولكنها ذهبت الى هدف ابعد حيث قامت بغلق كافة الـ States المارة بها مما سهل للخوارزمية الثانية Best-First-Search بعدم المرور بتلك الـ States المغلقة وبالاخص State C<sub>95</sub> مما جعل هذه الخوارزمية ان تسلك مساراً اخر وتحقق الهدف المطلوب.

لا بد من الاشارة هنا ان الخوارزميتين لو نفذتا على انفراد لما وصلنا الى الهدف المطلوب. وبهذه الطريقة استطعنا ان نحل احد المشاكل القائمة في عمليات البحث الاستباطي (Heuristic search) باستخدام خوارزمية البحث الاستباطي الهجين (Hybrid Heuristic search) حيث تمكنت هذه الخوارزمية من الوصول الى الهدف GL2 عبر المسار التالي:-

Path=GL2<sub>7</sub> > Y1<sub>18</sub> > U1<sub>30</sub> > P1<sub>44</sub> > K1<sub>50</sub> > D1<sub>60</sub> > W<sub>79</sub> > P<sub>84</sub> > H<sub>91</sub> > D<sub>96</sub> > A<sub>100</sub>

كما ورد في المسار 56 . كما هو موضح بالمسار المنقط التالي في الشكل رقم ( 3 ) :-



الشكل رقم ( 3 )

## 7- الاستنتاجات والتوصيات (Discussion and Conclusion)

- 1- تعتبر هذه الخوارزمية اسرع خوارزمية في ايجاد الهدف ولكونها تستدعي خوارزمية الـ Hill Climbing اولا وعندما تكون الكلفة محسوبة بدقة فأنها تصل الى الهدف بأقصر مسار وبالتالي لا يحتاج الى تنفيذ خوارزميات البحث الاستباطي لمعرفة اقصر مسار وبذلك قيل عنها اسرع خوارزمية.
- 2- في حالة وجود مشاكل في حساب الكلف هذا ما يؤدي الى ظهور States ذات كلف محظورة وعليه فأن خوارزمية الـ Best-First-Search وبمساعدة الشروط التي تمت مراعاتها في هذه الخوارزمية فانها سوف تصل الى الهدف ولو بمسار اطول ولكن بالنتيجة سوف تحقق الهدف المطلوب.
- 3- اذا كانت غالبية الكلف غير دقيقة في فضاء البحث فأن هذه الخوارزمية سوف تصل الى الهدف ايضا بعد ان تقوم بغلق كافة الـ States التي تسبق الهدف حتى تصل الى الهدف المطلوب حيث يشابه عملها عمل خوارزميات البحث المنتظم من حيث عدد نقاط البحث التي تم تدقيقها.
- 4- تم التوصل في هذه الخوارزمية Hybrid Algorithm الى معالجة الاخطاء التي تقع بها خوارزمية Hill-Climbing والتي تبعد المسار عن الهدف وذلك بأبعاد مثل تلك النقاط عن فضاء البحث لكي تتمكن خوارزمية الـ Best-First-Search من سلك مسارات اخرى كي تصل الى الهدف المطلوب.
- 5- اغلب الخوارزميات التي يتم اشتقاق الهدف عن طريق البيانات (Data Driven) او عن طريق الهدف (Goal Driven) يمكن ايجاد علاقات مشتركة بينها كما هو معمول في Production Rule حيث توجد خوارزمية هجينة تجمع عملية الاشتقاق لكلا الحالتين.

6- نوصي ببناء خوارزمية اخرى من نوع Hybrid Algorithm بطرق اخرى من طرق البحث ضمن خوارزميات تمثيل المعرفة [8] Knowledge Representation وذلك بايجاد علاقات مشتركة بين بعض الخوارزميات التي تعمل بإتجاهين في اشتقاق الهدف وهما الاشتقاق من جهة الهدف Goal Driven او الاشتقاق من جهة البيانات Data Driven وهذا ممكن تحقيقه في خوارزمية الاستدلال Deduction System ومن اشهر الخوارزميات التي بالامكان ربطها معا هما خوارزميات التمثيل الدلالي Semantic Network والتمثيل الصوري Conceptual Graph و خوارزمية الأطر Frame [9].

7- نوصي بايجاد علاقات مشتركة في اشتقاق الهدف في خوارزمية الاستدلال Deduction System بين كل من Forward Deduction System وبين Backward Deduction System حيث لم يتم التطرق في بناء خوارزميات هجينة في هذا الميدان.

## References المصادر

- 1- Berliner .H.J, "The B\* Tree Search algorithm A Best First proof procedure" Artificial Intelligence, Vol 12, No 1, May 1989.
- 2- Martelli, A., "On the Complexity of admissible search algorithm " , Artificial Intelligence, Vol 8, 1997.
- 3- George .F. Luger. " Structure and Strategies for Complex Problem Solving". Artificial Intelligence, Vol 3, 1997.
- 4-Elaine Rich, " Artificial Intelligence", Computer Science Series, the university of Texas at Austin, 2002.
- 5- Newell, A., "The heuristic of George Playa and its Relation to Artificial Intelligence", in methods of heuristics-R. Groner, W.F.Bischoof (Eds), Erlbaum, Hillsdale,N.J.,1983.
- 6- Berliner .H.J, "On the Construction of Evaluation Function for Large Domains", in proc. IJCAI 6. 1999.
- 7- Simon.H.A. & J.B.Kadane, " Optimal Problem – Solving Search: All-or-non Solutions", Artificial Intelligence, Vol 6, No 3, 1995.
- 8- Bobrow D.G. & T. Winograd " An Overview of KRL. A knowledge Representation Language" Cognitive Science, Vol 1, No 2001.
- 9- Brachman.R.J. "On the Epistemological status of Semantic Networks". In Associative networks N.V. Findler(Ed.). Academic press, Newyork 2000.