COMP6320: Advanced Assignment 1

Suraj Narayanan Sasikumar (u5881495)

August 5, 2016

1 Stronger Admissible Heuristics for the Multiple Way-point Navigation Search Problem

Problem Description: The best heuristic any search problem could use is the **exact** cost to achieving the goal, unfortunately computing such a function is extremely expensive and in most cases take exponential time. Hence we use heuristic functions that under-estimate (admissible) the exact cost of achieving a goal state from the given state. There is an inverse relationship between the computational complexity and the accuracy of a heuristic function, the better we estimate the actual cost, the harder it gets to compute those estimates. In this section we talk about this inherent trade-off between the accuracy and run-time of heuristic functions with the help of the red-bird search problem.

1.1 Formalise the Search Problem (1 TB Mark)

Definition 1.1.1 EAT-YELLOW ***Task***

- W is the width of the Grid World
- H is the height of the Grid World
- $X = \{1, ..., W\}$ is the set of x-coordinates in the Grid World
- $Y = \{1, ..., H\}$ is the set of y-coordinates in the Grid World
- $G = X \times Y$ is the set of coordinates representing the Grid World
- $r = \langle r_x, r_y \rangle$ is the coordinate of the ***red-bird*** in the Grid World. $r \in G$
- r_0 is the initial coordinate of the ***red-bird*** in the Grid World. $r_0 \in G$
- Y_b is a finite set of coordinates of all the yellow-birds in the Grid World. $Y_b \subset G$
- $iswall : G \rightarrow \{True, False\}$ is the ***check-for-wall*** function. We say that red-bird r may ***move*** to the coordinate, $\langle x, y \rangle \in G \quad iff. \quad iswall(\langle x, y \rangle) = False$
- $D = \{North, South, East, West\}$ is the set of ***directions*** the red-bird r can move

The coordinate of the red-bird should not be a member of the yellow-bird set, i.e. $r \notin Y_b$

Definition 1.1.2 EAT-YELLOW ***Domain***
The EAT-YELLOW domain maps EAT-YELLOW tasks in a Grid World G, with red-bird at r, and yellow-birds at Y_b to state spaces as follows:

STATES: Pairs $\langle r, Y_b \rangle$. The set of all possible states is denoted as S

INITIAL STATE: $\langle r_0, Y_b \rangle$, where r_0 is the initial coordinate of the ***red-bird***.

GOAL STATE: $\langle r, \emptyset \rangle$. The goal state is when the set of yellow-bird is a null-set

OPERATORS: $move : S \times D \nrightarrow S$

> The red-bird can ***move*** in a given direction to a new coordinate $\langle r_x^{'}, r_y^{'} \rangle$ if $iswall(\langle r_x^{'}, r_y^{'} \rangle) = False$ and $\langle r_x^{'}, r_y^{'} \rangle \in G$.
> For the ***move*** operation, following are the changes for the various directions:
> $East : r_x^{'} = r_x - 1$
> $West : r_x^{'} = r_x + 1$
> $North : r_y^{'} = r_y + 1$
> $South : r_y^{'} = r_y - 1$
> This action changes the value of r to $\langle r_x^{'}, r_y^{'} \rangle$. If $\langle r_x^{'}, r_y^{'} \rangle \in Y_b$, then remove $\langle r_x^{'}, r_y^{'} \rangle$ from Y_b. The cost of this action is always 1.

1.2 Propose and Implement a Stronger Admissible Heuristic (1 TB Mark)

We first lift the, $iswall(\langle x, y \rangle) = False$ constraint on the ***move*** operator in the EAT-YELLOW Domain. This creates a relaxed problem for which we can easily find an optimal solution. The optimal solution for the relaxed problem is strictly the sum of some Manhattan distances between birds. Due to the absence of walls it is guaranteed that traversal between birds in an optimal path would be a Manhattan path. This is an admissible heuristic because the cost of traversal in the original problem is higher, since the red-bird has to circumvent walls. The optimal solutions to this relaxed problem presents a more accurate, and hence a more informed approximation of the actual multiple yellow-bird navigation problem. Since for all states the values of PDB heuristic is less that the value of Bird-to-Bird Manhattan, it is more accurate than PDB.

1.3 Formalise Your Heuristic and Prove Admissibility (3 TB Marks)

Formalism of the relaxed problem

Definition 1.3.1 RELAXED-EAT-YELLOW ***Task***

- W is the width of the Grid World
- H is the height of the Grid World
- $X = \{1, ..., W\}$ is the set of x-coordinates in the Grid World
- $Y = \{1, ..., H\}$ is the set of y-coordinates in the Grid World
- $G = X \times Y$ is the set of coordinates representing the Grid World
- $r = \langle r_x, r_y \rangle$ is the coordinate of the ***red-bird*** in the Grid World. $r \in G$
- r_0 is the initial coordinate of the ***red-bird*** in the Grid World. $r_0 \in G$
- Y_b is a finite set of coordinates of all the yellow-birds in the Grid World. $Y_b \subset G$
- $D = \{North, South, East, West\}$ is the set of ***directions*** the red-bird r can move

The coordinate of the red-bird should not be a member of the yellow-bird set, i.e. $r \notin Y_b$

Definition 1.3.2 RELAXED-EAT-YELLOW ***Domain***
The RELAXED-EAT-YELLOW domain maps RELAXED-EAT-YELLOW tasks in a Grid World G, with red-bird at r, and yellow-birds at Y_b to state spaces as follows:

STATES: Pairs $\langle r, Y_b \rangle$. The set of all possible states is denoted as S

INITIAL STATE: $\langle r_0, Y_b \rangle$, where r_0 is the initial coordinate of the ***red-bird***.

GOAL STATE: $\langle r, \emptyset \rangle$. The goal state is when the set of yellow-bird is a null-set

OPERATORS: $move : S \times D \nrightarrow S$

The red-bird can ***move*** in a given direction to a new coordinate $\langle r_x^{'}, r_y^{'} \rangle$ if $\langle r_x^{'}, r_y^{'} \rangle \in G$.
For the ***move*** operation, following are the changes for the various directions:
$East : r_x^{'} = r_x - 1$
$West : r_x^{'} = r_x + 1$
$North : r_y^{'} = r_y + 1$
$South : r_y^{'} = r_y - 1$
This action changes the value of r to $\langle r_x^{'}, r_y^{'} \rangle$. If $\langle r_x^{'}, r_y^{'} \rangle \in Y_b$, then remove $\langle r_x^{'}, r_y^{'} \rangle$ from Y_b. The cost of this action is always 1.

Proof of Admissibility

Lemma 1. *The optimal solution of RELAXED-EAT-YELLOW Domain is a lower bound of the cost of the optimal solution of the EAT-YELLOW Domain.*

Proof. In RELAXED-EAT-YELLOW, we calculate the sum on the least bird-to-bird Manhattan distances. Without loss in generality let $cost_{rlx}$ represent the optimal cost of some instance of RELAXED-EAT-YELLOW. We can represent them as sum of Manhattans:

$$cost_{rlx} = \sum_{i}^{len(yellow-bird)} (Manhattan_i) \tag{1}$$

For the same instance of the REALAXED-EAT-YELLOW, re-apply the lifted constraint to get the problem as an instance of EAT-YELLOW. Now let $cost_{orig}$ represent the optimal cost. Because of the constraint, and if there exists walls in-between the Manhattan path, the optimal cost would increase by say some value k_i for each yellow-bird i, giving us:

$$cost_{orig} = \sum_{i}^{len(yellow-bird)} (Manhattan_i + k_i) \tag{2}$$

$(2) - (1)$ gives:

$$cost_{orig} - cost_{rlx} = \sum_{i}^{len(yellow-bird)} k_i$$

Since any additional distance to circumvent walls is either zero or a positive quantity, we have:

$$\sum_{i}^{len(yellow-bird)} k_i \geq 0$$

$$\Rightarrow cost_{orig} - cost_{rlx} \geq 0$$

$$\Rightarrow cost_{rlx} \leq cost_{orig}$$

Hence the optimal solution of RELAXED-EAT-YELLOW is a lower bound of the cost of the optimal solution of the EAT-YELLOW problem. ∎

2 Epidemics Simulator v. 2.0

2.1 Formalisation as a Search Problem (1 TB Mark)

Definition 2.1.1 KILL-EPIDEMIC ***Task***

- $G = \langle L, E \rangle$ is the ***epidemic graph***. Its vertices L is a finite set of ***locations***, its edges E are ***connections*** between locations.
- $t \in \mathbb{R}$ is the ***threshold*** value above which disease from a location starts to spread into its connected locations.
- $g \in [0, 1]$ represents the ***growth rate*** of a disease at a particular location
- $s \in [0, 1]$ represents the fraction of the disease at a particular location that ***spread*** to connected locations.
- $h \in L$ is the current location of the ***health agent***
- $h_0 \in L$ is the ***initial location*** of the health agent
- $disease : L \to \mathbb{R}$ is the ***disease*** function that returns the intensity of the epidemic spread in the given location.

Definition 2.1.2 KILL-EPIDEMIC ***Domain***
The KILL-EPIDEMIC domain maps KILL-EPIDEMIC tasks with location L, and health agent at h to state spaces as follows:

STATES: Pairs $\langle h, disease \rangle$. The set of all possible states is denoted as S

INITIAL STATE: $\langle h_0, disease_0 \rangle$, where h_0 is the initial coordinate of the ***health-agent***, and $disease_0$ is the initial ***disease function***.

GOAL STATE: Any state $\langle h, disease \rangle$ such that $\sum_{l \in L} disease(l) = 0$

OPERATORS: $move : S \times L \nrightarrow S$
A health-agent at location h can ***move*** to location h' in state $\langle h, disease \rangle \quad iff. \quad \{h, h'\} \in E$, i.e. locations h and h' are connected. This action changes the value of h to h', and assign 0 to $disease(h')$. The cost of this action is always 1.

3 The Transport Problem

3.1 Heuristics for the Transport Domain (4 TB Marks)

Definition 2.1.1 RELAXED-TRANSPORT ***Task***

- $G = \langle V, E\rangle$ is the ***roadmap graph*** or ***roadmap***. Its vertices V are called ***locations***, its edges ***roads***
- M is a finite set of ***mobiles***
- P is a finite set of ***portables***
- $cap : M \to \{1, ...|P|\}$ is the ***capacity*** function. We say that mobile m has ***unbounded capacity*** $iff.\ cap(m) = |P|$
- $w_m : M \times E \to \mathbb{N}_1 \cup \{\infty\}$ is the ***movement cost*** function. If $w_m(m, e) \neq \infty$, we say that mobile m ***may access*** road e. The ***roadmap graph*** or ***roadmap*** G_m of m is the weighted graph obtained by restricting G to those roads which m may access, weighted by the corresponding movement cost. We require that each road maybe accessed by some mobile.
- $w_p : P \to \mathbb{N}_0$ is the ***pickup cost*** function
- $l_0 : (M \cup P) \to V$ is the ***initial location*** function.
- $fuel_0 : M \to \infty$ is the ***initial unrestricted fuel*** function. This is the lifted constraint that makes this task a relaxed problem. All mobiles are assumed to have infinite fuel supply.
- $l_* : P \to V$ is a function from portables to locations called ***goal locations***.

Definition 2.1.2 RELAXED-TRANSPORT ***Domain***
The RELAXED-TRANSPORT domain maps RELAXED-TRANSPORT tasks with location V, mobiles M, and portables P to state spaces as follows:

STATES: Pairs $\langle l, fuel\rangle$, where $l : M \cup P \to V \cup M$ is the ***location*** function and $fuel : V \to \infty$ is the ***fuel reserve*** function. All original states that have restricted fuel quantity is mapped to relaxed state spaces that have infinite fuel supply. Only portables may have mobiles as there location. The set of all possible states is denoted as S

INITIAL STATE: $\langle l_0, fuel_0\rangle$, where l_0 is the initial location function and $fuel_0$ is the initial fuel function of the task.

GOAL STATE: Any state $\langle l, fuel\rangle$ such that $\forall p \in P.\ l(p) = l_*(p)$

OPERATORS:
- $move : S \times M \times V \nrightarrow S$
  This operation allows a mobile m to ***move*** from its current location $l(m)$ to a new location $v^{'}$. This operation is defined only when $[(\{l(m), v^{'}\} \in E) \wedge w_m(m, \{l(m), v^{'}\}) \in \mathbb{N}_1]$ holds. Since this is a relaxed problem the fuel restriction on the mobile is lifted. The cost of this operation is given by the movement cost function applied to mobile m and edge $\{l(m), v^{'}\}$. This action changes $l(m)$ to $v^{'}$

- $pickup : S \times M \times P \nrightarrow S$
  This operation allows a mobile m to ***pickup*** a portable p. This operation can be applied in state $\langle l, fuel\rangle$ $iff.$ $[l(p) = l(m) \wedge cap(m) > |\{p \in P : l(p) = m\}|]$ holds. The cost of this operation is given by the pickup cost function applied to m and p. This action changes $l(p)$ to m

- $drop : S \times M \times P \nrightarrow S$
  This operation allows a mobile m to ***drop*** a portable p. This operation can be applied in state $\langle l, fuel\rangle$ $iff.$ $[l(p) = m]$ holds. The cost of this operation is given by the pickup cost function applied to m and p. This action changes $l(p)$ to $l(m)$.

Since we are relaxing the problem by lifting the constraint on fuel requirements, the state space has dropped a dimension since $fuel$ in $\langle l, fuel\rangle$ is now a constant, namely ∞.

Proof of Admissibility

Lemma 2. *The optimal solution of RELAXED-TRANSPORT domain is a lower bound of the cost of the optimal solution of the TRANSPORT Domain.*

Proof. We know that the RELAXED-TRANSPORT domain can be solved in polynomial time [1], the optimal solution to a transportation problem where fuel constraints are lifted would be to use the mobiles that have the minimal transportation cost, i.e. movement+pickup+drop costs over all the roads. When fuel constraints and re-applied, zero or more least cost mobiles would be unavailable for transportation all the time. This would mean that the optimal cost of transportation under fuel constraints would be greater that or equal to the the cost when fuel constraints are lifted. Hence the optimal cost of our RELAXED-TRANSPORT domain is a lower bound of the cost of the optimal solution of the TRANSPORT domain ■

References

[1] Helmert, M., 2006. Solving planning tasks in theory and practice. Albert-Ludwigs-Universitat Freiburg Doctoral thesis. pp.54-55.