تمرین جستجوی آگاهانه عماد آقاجان زاده ۹۷۱۲۷۶۲۲۹۴ سوال) بررسی کنید که آیا نیاز به تغییر محل چک visited در BFSهست زمانی که بخواهیم * A را پیاده سازی کنیم ؟

در الگوریتم BFS هر بار از بالای یک صف یک state را برمیداشتیم و همان را گسترش می دادیم و این visited گسترش دادن با یک روش غیرآگاهانه و براساس داشتن کمترین عمق انجام میشد . این چک کردن state هم همواره پیش از گسترش دادن هر فرزند توسط پدر یا به عبارت دیگر پیش از قرار دادن یک state در داخل صف بررسی می شود .

در الگوریتم *A هم مشابه الگوریتم BFS یک صف داریم که تنها تفاوت آن در این است که عناصر آن برحسب تابع f(n) مرتب شده اند و انتخاب هر state همانند BFS همانند State و بررسی state بودن یک state هم همانند BFS پیش از قرار دادن State داخل f(n) صورت میگیرد .

در نتیجه بنا به توضیحات بالا نیازی به تغییر محل چک visited نیست .

انتخاب هيوريستيك

برای انتخاب یک هیوریستیک خوب ابتدا تمام فرض های سوال را مینویسیم:

۱ – حرکت در خانه های مجاور

۲ – عدم ورود به خانه های پر (*)

 $= - \sqrt{2}$ مکعب دارای یک قوانین خاص هستند مثلا اگر در حالت ایستاده هستیم در حرکت بعدی قطعا حالت مکعب خوابیده است .

حال برای پیدا کردن یک هیوریستیک خوب سعی میکنیم که فرض های بالا را در نظر نگیریم:

حالت اول : تقض فرض شماره ١

پس از هر action باید state دنیا را به تابع خاصی داد و آن تابع می تواند با نقض شروط خانه های مجاور تعداد خانه های رنگ نشده را برگرداند البته باید در نظر گرفت که در این شمارش نباید فرض سوم را نقض کنیم مثلا اگر چنین حالتی را داشتیم که سه خانه رنگ نشده پشت سر هم بودند : . . . در این صورت هیچ با حرکتی که شرط ۳ را نقض نکند نمیتوان آن را رنگ کرد .

این هیوریستیک از آن جا که تنها یک شرط را نقض می کند اما پیاده سازی آن زمان بر است چراکه در هر بار تغییر حالت دنیا باید یک مسئله دیگر حل شود که منطقی نیست .

حالت دوم: نقض فرض شماره ٢

در هر state باید مکان فعلی مکعب و اندازه دنیا را به یک تابعی داد تا آن جداگانه h را تخمین بزند .

این هیوریستیک هم مشکل بالایی را دارد چراکه در هر حالت دنیا باید یک مسئله حل کند که در آن مسئله تنها شرط عدم ورود به خانه های پر نقض شده است که مسئله سختی است .

حالت سوم : نقض فرض شماره ٣

در هر مکان که باشیم می توانیم به خانه های مجاور برویم بدون اینکه فرض ۳ را رعایت کنیم یعنی میتوانیم در حالت ایستاده به سمت راست حرکت کنیم و حالت بعدی هم ایستاده بماند .

پیاده سازی این هیوریستیک هم مشکلات دو هیوریستیک بالایی را دارد .

پس از بررسی هایی که انجام شد مشخص شد استفاده از هیوریستیکی که تنها یکی از فرض ها را نقض کند با وجود آنکه به هیوریستیک واقعی نزدیکتر است اما چندان مناسب نیست زیرا هم پیاده سازی دشواری دارد و هم آنکه زمان زیادی را صرف تخمین تابع هیوریستیک می کند .

اما نقض کردن ۲ تا از فرض ها انتخاب مناسبی به نظر میرسد:

حالت چهارم : نقض فرض های شماره ۱ و ۳

در این حالت هیوریستیک state دنیا را میگیرد و چون محدودیت حرکت در خانه و محدودیت های حرکتی ندارد تنها کافیست تا خانه های رنگ نشده را بشماریم البته بسته به اینکه حرکت مکعب را چگونه فرض میکنیم ۲ حالت خواهیم داشت :

h(n) = uncolored

این حالت overestimate خواهد شد یعنی در بعضی حالات از h واقعی بیشتر می شود .

و اگر حرکت مکعب را خوابیده در نظر بگیریم:

h(n) = uncolored/2

میتوان حالت های دیگری را هم در نظر داشت که در آن فرض های دیگر را نقض کنیم و چون در حالات دیگر هم ناچار به نقض دو فرض هستیم تفاوت آن چنانی با همین حالت ۴ نخواهد داشت و همین حالت چهار را به عنوان هیوریستیک خود انتخاب میکنیم.

admissibility اثبات

همان گونه که در بالا اشاره شد هیوریستیک مورد استفاده در هر State نصف خانه های رنگ نشده را می شمارد .



به عنوان مثال :

در حالت رو به رو هیوریستیک ما فرض می کند به همه خانه ها می شود بدون محدودیت مجاور بودن در حالت خوابیده مکعب رفت در این صورت مقدار آن برابر خواهد بود با نصف تعداد خانه های رنگ نشده.

حال اگر فرض کنیم در یک State تعدادی خانه رنگ نشده داریم تابع * که مقدار دقیق را برمیگرداند در کمترین حالت خود باز هم باید بیشتر از تابع * ما باشد :

در کمترین حالت h مکعب وارد هیچ خانه از پیش رنگ شده نمی شود و با هر حرکت تعدادی خانه رنگ نشده را رنگ می کند اما تعداد این حرکات همواره از h ما بیشتر خواهد بود چراکه برای رنگ خانه ها یکبار γ خانه را رنگ می کند و بار دیگر γ خانه را رنگ می کند در صورتی که در γ ما همواره فرض میکنیم که در هر مرحله γ خانه رنگ می شود .

پس نتیجه میگیریم که تابع هیوریستیک ما که ۲ فرض را از مسئله در نظر نمیگیرد همواره جواب کوچکتری نسبت به جواب اصلی مسئله دارد و از این رو همواره قابل قبول (admissible) است .

اثبات consistency

برای این کار از برهان خلف کمک میگیریم به شرح زیر :

