

## ۱- الگوریتم AND - OR - GRAPH - SEARCH:

باید چک کنیم که هر گره فقط یکبار دیده شود، و از آنجاییکه گره های OR پدران گره های AND هستند پس یکبار دیده شدن گره های OR یکبار دیده شدن گره های AND را تضمین میکند.  
علاوه بر این باید مقدار plan محاسبه شده هر گره OR که تا کنون دیده شده است را نیز نگهداریم.  
پس فقط عملکرد در مورد گره های OR عوض میشود.  
با توجه به توضیحات بالا یک دیکشنری به نام Plans میسازیم که کلیدهای آن شماره های گره های OR هستند و مقادیر به ازای هر کلید بصورت زیر است:

- اگر گره مربوطه موفق نباشد: 1-
  - اگر گره مربوطه موفق باشد: plan محاسبه شده
- توجه شود که اگر گره ای هنوز دیده نشده، پس شماره آن در مجموعه کلیدها وجود ندارد.

با استفاده از این ابزار هر بار به یک گره OR میرسیم بصورت زیر عمل میکنیم:

- اگر شماره گره در مجموعه کلیدهای دیکشنری Plans وجود داشت:
- مقدار متناظر با آن را برمیگردانیم.
- اگر شماره گره در مجموعه کلیدهای دیکشنری Plans وجود نداشت:
- ابتدا مقدار plan را برای آن محاسبه میکنیم (پیمایش گره های AND که فرزند هستند)  
سپس شماره و مقدار plan گره را به دیکشنری Plans اضافه میکنیم.  
درنهایت مقدار plan را برمیگردانیم.

## ۲- الگوریتمی که از هر یک از حالت های زیر حاصل می شود را نام ببرید:

(آ) local beam search در صورتی که  $k = 1$  باشد:

Hill-climbing search

(ب) stimulated annealing با  $T = 0$  در همه زمان ها (با حذف تست پایانی):

Hill-climbing search

(پ) الگوریتم ژنتیک با اندازه جمعیت  $N = 1$ :

First choice hill-climbing