



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)  
دانشکده ریاضی و علوم کامپیوتر

پایان نامه کارشناسی ارشد  
گرایش ریاضی

هوش مصنوعی - گزارش ۰۲ - پیاده سازی الگوریتم  
های هیوریستیک روی یک مسئله پوشش راسی

پایان نامه

نگارش  
آترین حجت

استاد راهنما  
نام کامل استاد راهنما

استاد مشاور  
نام کامل استاد مشاور

فروردین ۱۴۰۰

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# صفحه فرم ارزیابی و تصویب پایان نامه - فرم تأیید اعضاء کمیته دفاع

در این صفحه فرم دفاع یا تأیید و تصویب پایان نامه موسوم به فرم کمیته دفاع - موجود در پرونده آموزشی - را قرار دهید.

## نکات مهم:

- نگارش پایان نامه/رساله باید به **زبان فارسی** و بر اساس آخرین نسخه دستورالعمل و راهنمای تدوین پایان نامه های دانشگاه صنعتی امیرکبیر باشد.(دستورالعمل و راهنمای حاضر)
- رنگ جلد پایان نامه/رساله چاپی کارشناسی، کارشناسی ارشد و دکترا باید به ترتیب مشکی، طوسی و سفید رنگ باشد.
- چاپ و صحافی پایان نامه/رساله بصورت **پشت و رو(دورو)** بلامانع است و انجام آن توصیه می شود.



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

به نام خدا

## تعهدنامه اصالت اثر

تاریخ: فروردین ۱۴۰۰

اینجانب **آترین حجت** متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است. در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مآخذ بلامانع است.

**آترین حجت**

امضا

# فهرست مطالب

صفحه

عنوان

۱	شرح مسئله و روند کار	۱
۱-۱	مقدمه	۲
۲	توابع ابتدایی	۳
۱-۲	تولید گراف تصادفی	۴
۲-۲	بررسی هدف	۴
۳	الگوریتم‌ها	۵
۱-۳	$A^*$	۶
۲-۳	تخمین و روش محاسبه	۶
۱-۲-۳	بررسی Admissibility	۶
۲-۲-۳	نمونه‌ها	۶
۳-۳	Greedy search best-first	۶
۱-۳-۳	نمونه‌ها	۱۰
۴-۳	Hill-climbing	۱۰
۱-۴-۳	تابع تخمین ارزش	۱۰
۲-۴-۳	همسایگی	۱۳
۳-۴-۳	شروع تصادفی	۱۳
۴-۴-۳	نمونه‌ها	۱۳
۵-۳	Search Annealing	۱۳
۱-۵-۳	نمونه‌ها	۱۳
۴	مقایسه روش‌های متفاوت	۲۰

## فهرست اشکال

صفحه

شکل

۷	.....	$P = 0.2$	۱-۳
۸	.....	$P = 0.3$	۲-۳
۹	.....	$P = 0.12$	۳-۳
۱۰	.....	$P = 0.2$	۴-۳
۱۱	.....	$P = 0.3$	۵-۳
۱۲	.....	$P = 0.12$	۶-۳
۱۴	.....	$P = 0.2$	۷-۳
۱۵	.....	$P = 0.3$	۸-۳
۱۶	.....	$P = 0.12$	۹-۳
۱۷	.....	$P = 0.2$	۱۰-۳
۱۸	.....	$P = 0.3$	۱۱-۳
۱۹	.....	$P = 0.12$	۱۲-۳
۲۱	.....	تعداد تکرارهای برنامه به نسبت احتمال وجود هر یال	۱-۴
۲۲	.....	درصد یافتن جواب درست با افزایش احتمال وجود هر یال	۲-۴

## فهرست جداول

صفحه

جدول

## فهرست نمادها

نماد	مفهوم
$n(G)$	تعداد رئوس گراف $G$
$\deg(v)$	درجه‌ی راس $v$ در گراف
$nei(v)$	همسایه‌های راس $v$ در گراف



# فصل اول

## شرح مسئله و روند کار

## ۱-۱ مقدمه

مسئله پوشش راسی مسئله انتخاب رئوسی از گراف است بطوری که هر راس یا انتخاب شده باشد یا همسایه‌ای انتخاب شده داشته باشد. این مسئله Complete NP و دارای الگوریتم تخمین می‌باشد. در این گزارش این مسئله را با استفاده از راه‌حل‌های Huristic حل خواهیم کرد.

## فصل دوم

### توابع ابتدایی

## ۱-۲ تولید گراف تصادفی

برای تولید گراف تصادفی دو تابع مورد استفاده قرار گرفته است. تابع `gen_graph_eq_prob_edges` گرافی با `nodes` راس تولید میکند که هر یال با احتمال  $p$  در آن حضور دارد. تابع `gen_graph_fix_set_edges` گرافی با `nodes` راس و `edges` یال تصادفی می‌کند.

## ۲-۲ بررسی هدف

تابع `utils` در `solutions` روی تمام یال‌های رئوس انتخاب شده می‌گردد و رئوس دیده‌شده را علامت می‌زند. زمان اجرای این تابع  $O(EV)$  و حافظه‌ی آن  $O(V)$  می‌باشد. به دلیل محدود بودن کل عملیات‌های چک کردن حالت بهینه به  $NP$  برای بهینه‌سازی این توابع تلاشی نشده‌است.

# فصل سوم

## الگوریتم‌ها

### ۱-۳ $A^*$

$A^*$  نوعی خاصی از search -firstBest است که در تابع تخمین وزن، وزن مسیر تا آن لحظه نیز محاسبه می‌شود.

### ۲-۳ تخمین و روش محاسبه

فرض کنید  $G$  گراف مورد نظر ما باشد. در حالتی که در آن مجموعه  $C = v_1, \dots, v_t$  انتخاب شده‌اند برای محاسبه تابع Heuristic از فرمول

$$h(C) = \frac{n(G - C)}{\max_{v \in G - C - \text{nei}(C)} \deg(v) + 1} \quad (1-3)$$

استفاده می‌کنیم

### ۱-۲-۳ بررسی Admissibility

به وضوح انتخاب هر راس انتخاب نشده حداکثر  $\max_{v \in G - C} \deg(v) + 1$  راس جدید را پوشش می‌دهد پس بوضوح داریم

$$h(C) \leq h^*(C) \quad (2-3)$$

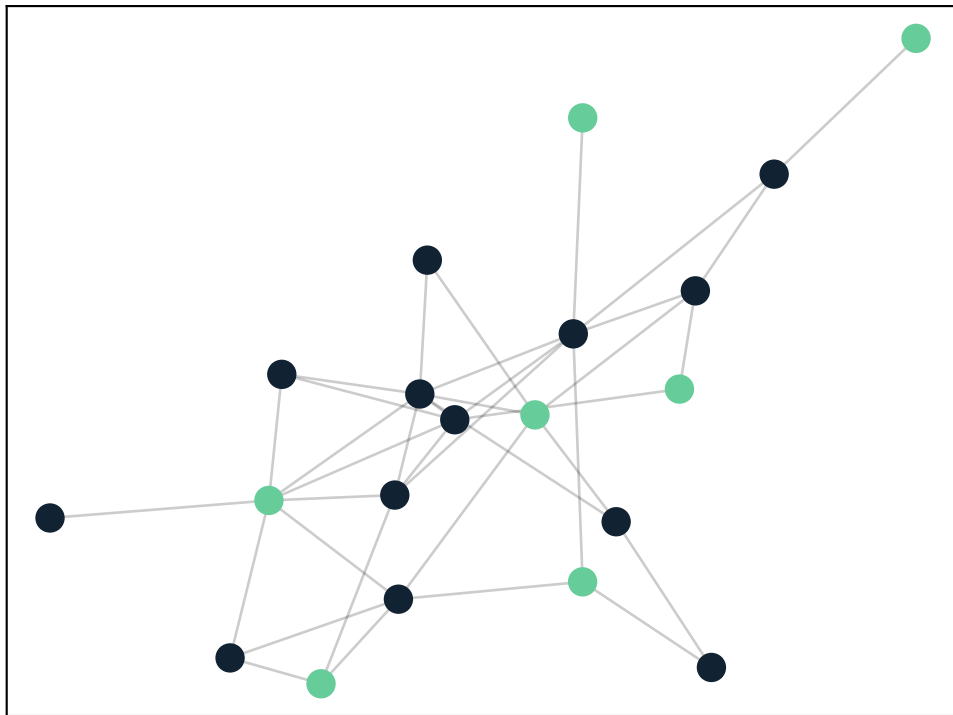
یعنی تابع هیوریستیکمان Admissible می‌باشد پس می‌توان نتیجه گرفت که  $A^*$  جواب بهینه می‌دهد و در نتیجه  $NP$  است.

### ۲-۲-۳ نمونه‌ها

### ۳-۳ Greedy search best-first

برای این الگوریتم نیز از همان تابع هیوریستیکی که در بالا استفاده شد استفاده می‌کنیم و پیاده سازی این الگوریتم کاملاً مشابه  $A^*$  است.

A★ Results graph 2.0-20

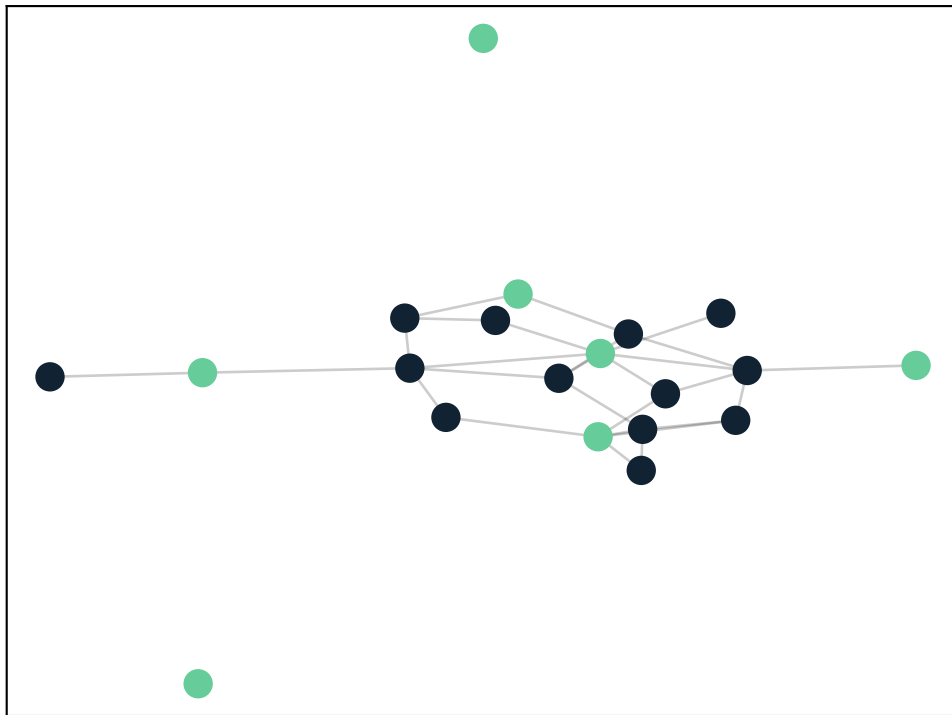


شکل ۳-۱:  $P = 0.2$

شکل ۳-۲:  $P = 0.3$

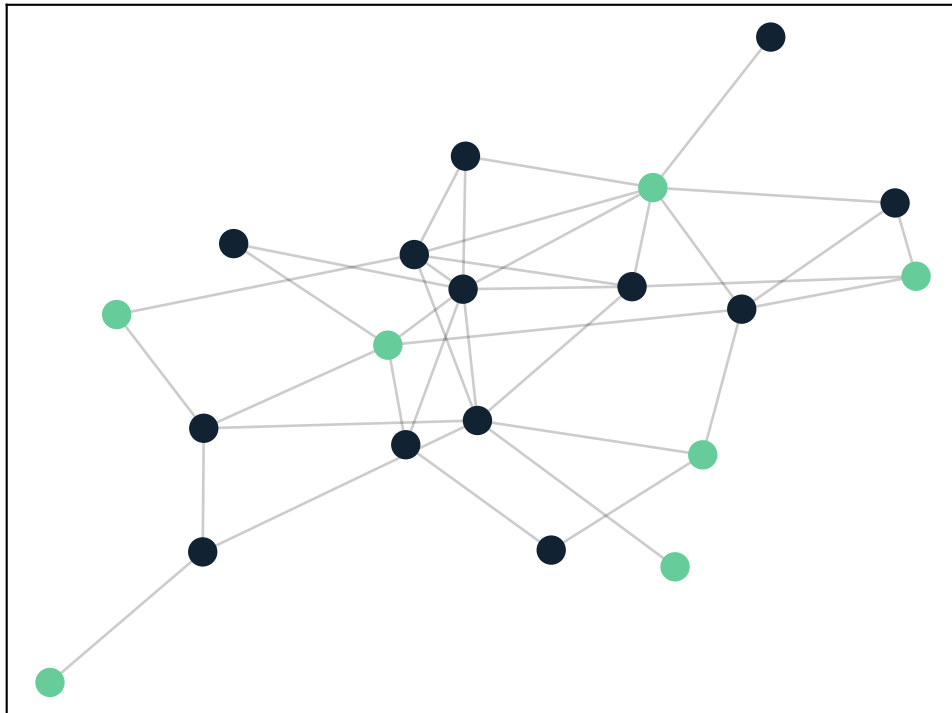


A★ Results graph 12.0-20



شکل ۳-۳:  $P = 0.12$

Greedy best-first search Results graph 2.0-20

شکل ۳-۴:  $P = 0.2$ 

۱-۳-۳ نمونه‌ها

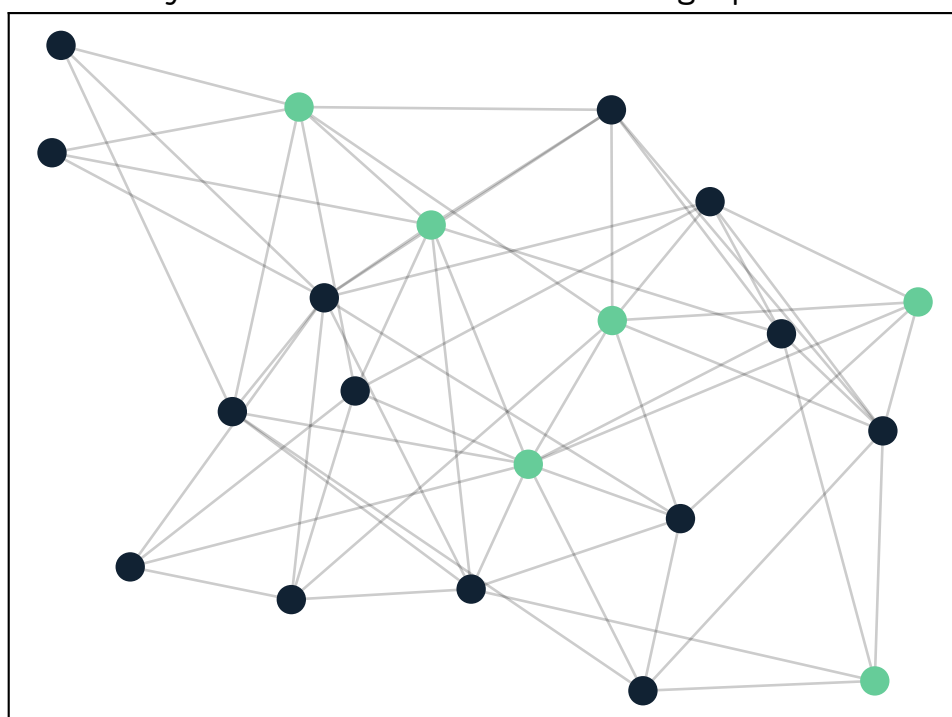
Hill-climbing ۴-۳

۱-۴-۳ تابع تخمین ارزش

مقدار این تابع باید طوری باشد که با افزایش راس‌های پوشش یافته زیاد شود و با افزایش تعداد رئوس انتخاب‌شده کاهش یابد. اگر ضریب این دو مقدار برابر باشند یعنی تابع به فرم  $-|C| - |G - C - nei(C)|$  باشد، الگوریتم لزومی در انتخاب یال‌های بدیهی نخواهد داشت. برای همین ضریب  $|G - C - nei(C)|$  را  $-2$  قرار دادیم و برای مثبت نگه داشتن کل عبارت را با  $2|G|$  جمع کردیم. پس تابع به فرم

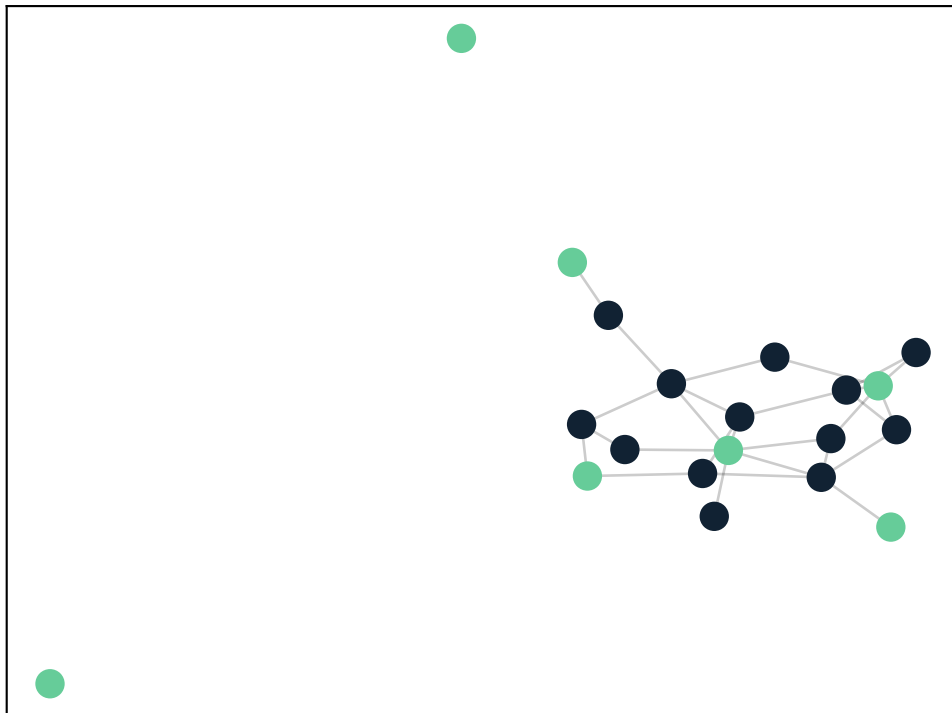
$$E(C) = 2|G| - 2|G - C - nei(C)| - |C| \quad (۳-۳)$$

Greedy best-first search Results graph 3.0-20



شکل ۳-۵:  $P = 0.3$

Greedy best-first search Results graph 12.0-20



شکل ۳-۶:  $P = 0.12$

خواهد بود.

### ۲-۴-۳ همسایگی

دو حالت را همسایه می‌گوییم هرگاه یکی با حذف دقیقاً یک عضو به دیگری تبدیل شود.

### ۳-۴-۳ شروع تصادفی

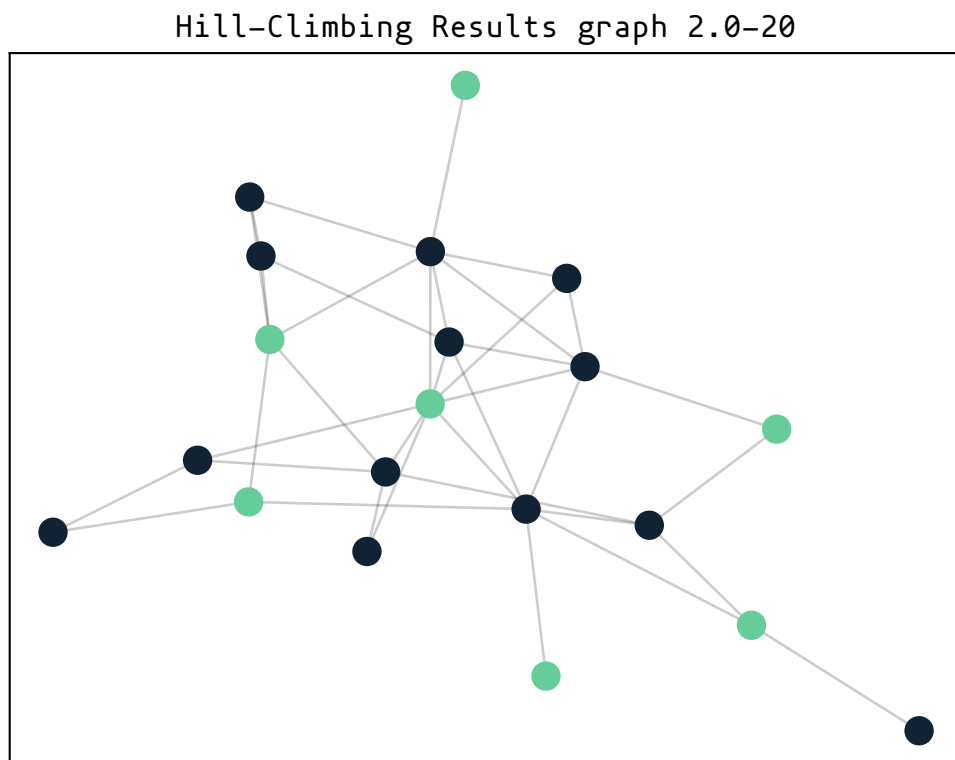
برای افزایش احتمال پیدا کردن جواب درست می‌توانیم بجای شروع از مجموعه خالی، از مجموعه‌ای از اعضای تصادفی انتخاب شده استفاده کنیم بطوری که احتمال حضور هر یک از آن‌ها در مجموعه‌ی اولیه برابر متغیر `rand_start` باشد.

### ۴-۴-۳ نمونه‌ها

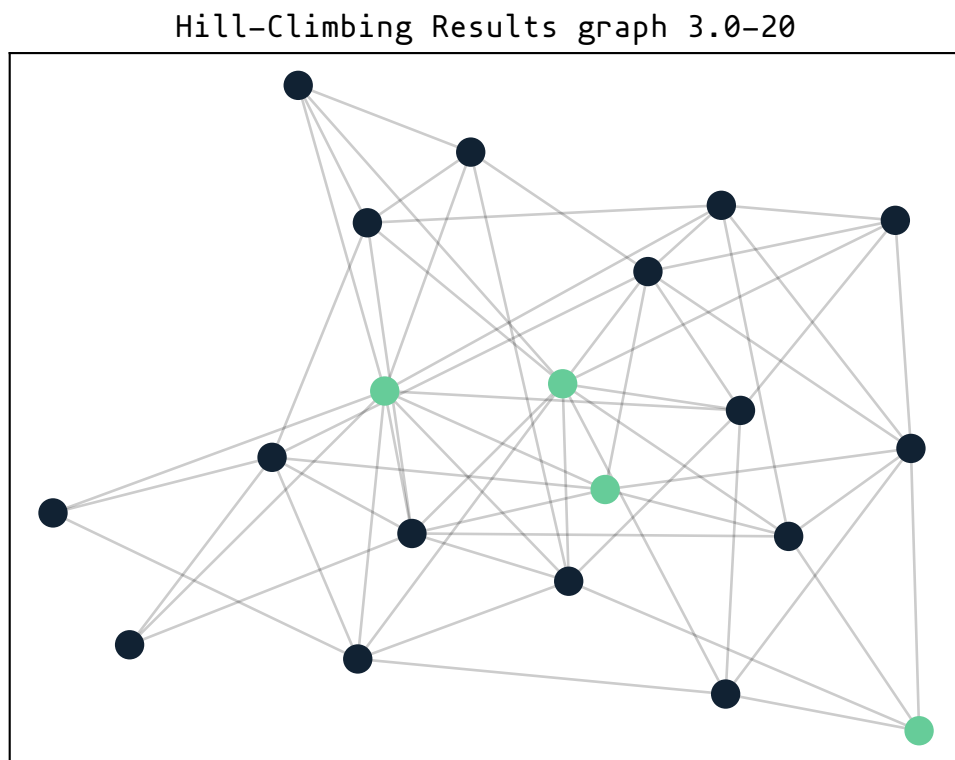
## ۵-۳ Search Annealing

برای پیاده سازی این الگوریتم از تابع ارزش‌گذاری مانند بالا استفاده می‌کنیم. تغییرات دما به فرم  $T_{next} = 0.98T_{now}$  با حداقل دمای 0.000001 و دمای اولیه ۱ ثبت شده‌اند.

### ۱-۵-۳ نمونه‌ها

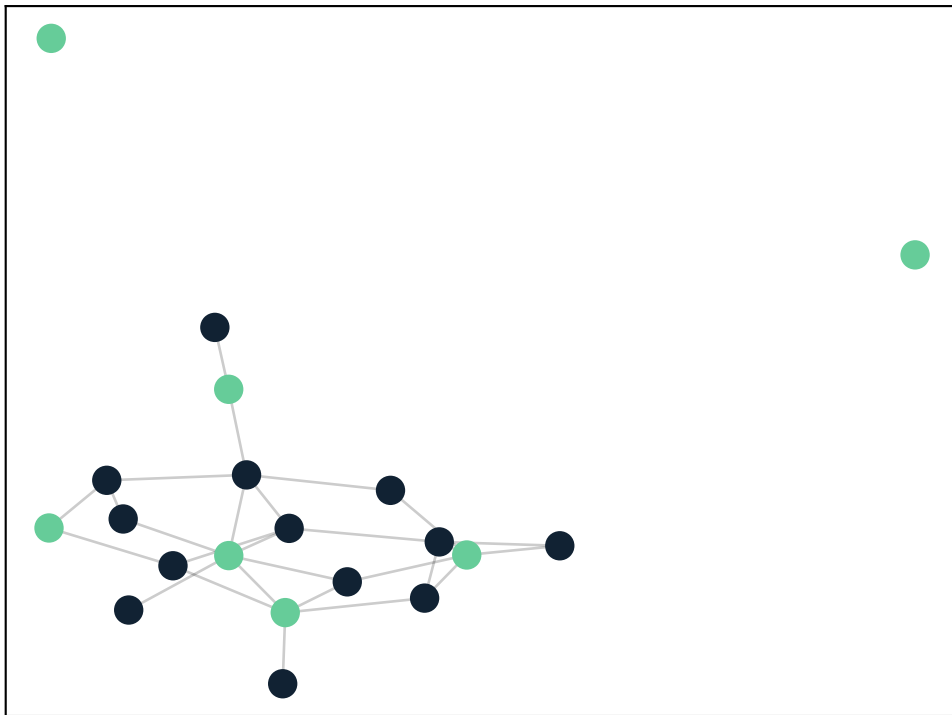


شکل ۳-۷:  $P = 0.2$



شکل ۳-۸:  $P = 0.3$

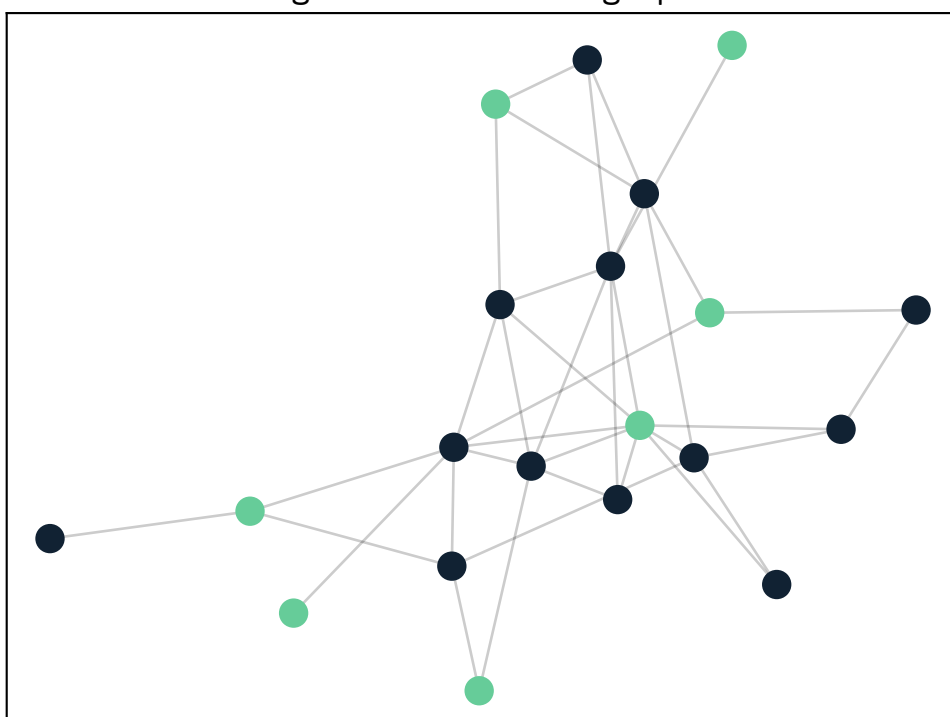
Hill-Climbing Results graph 12.0-20



شکل ۳-۹:  $P = 0.12$

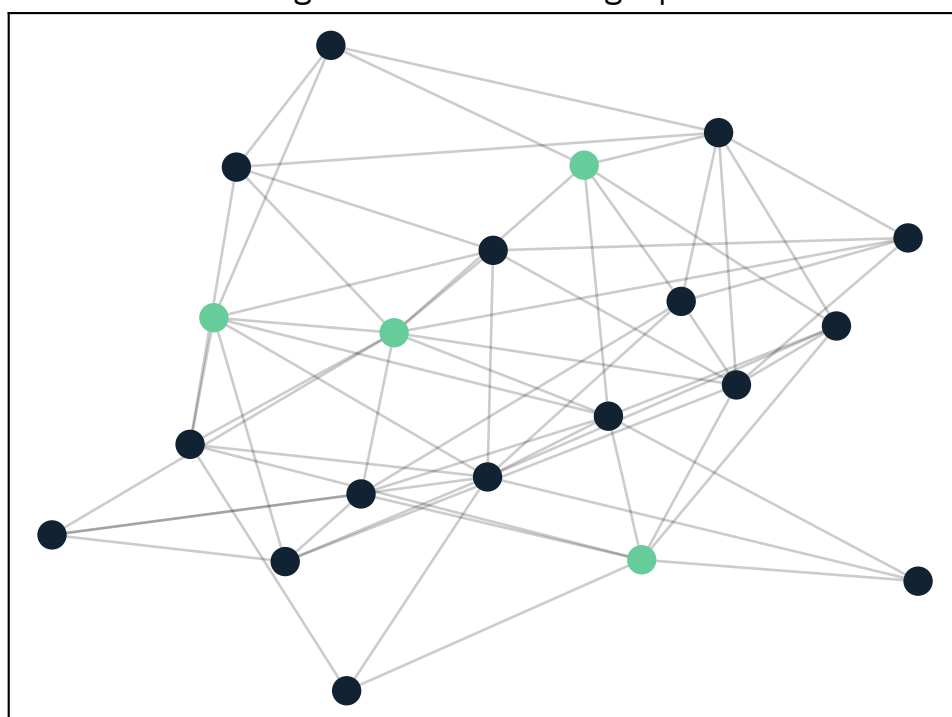


Annealing Search Results graph 2.0-20



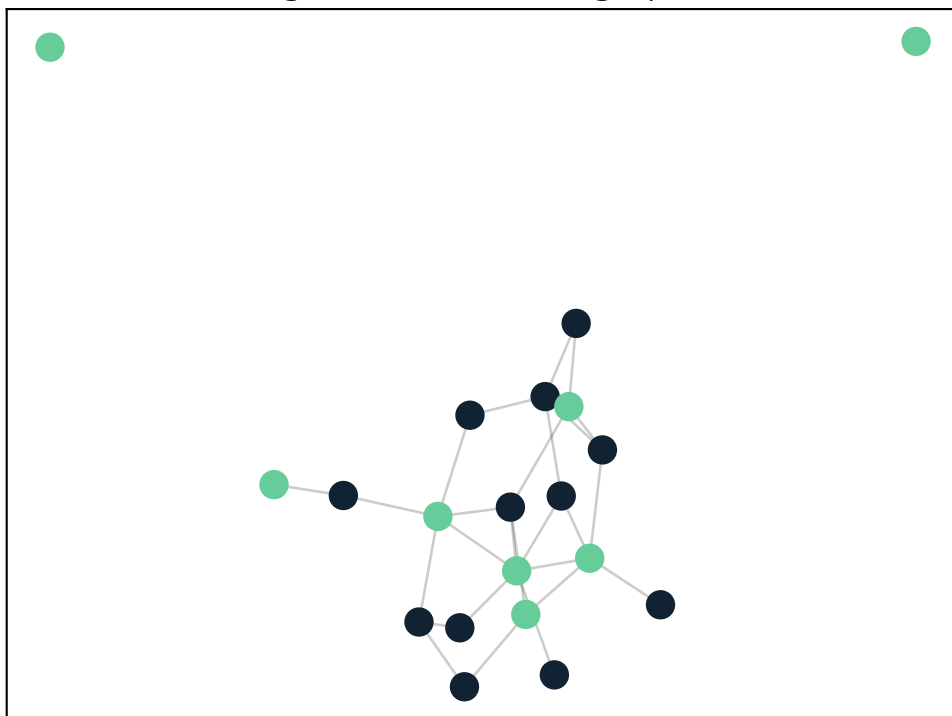
شکل ۳-۱:  $P = 0.2$

Annealing Search Results graph 3.0-20



شکل ۳-۱۱:  $P = 0.3$

Annealing Search Results graph 12.0-20

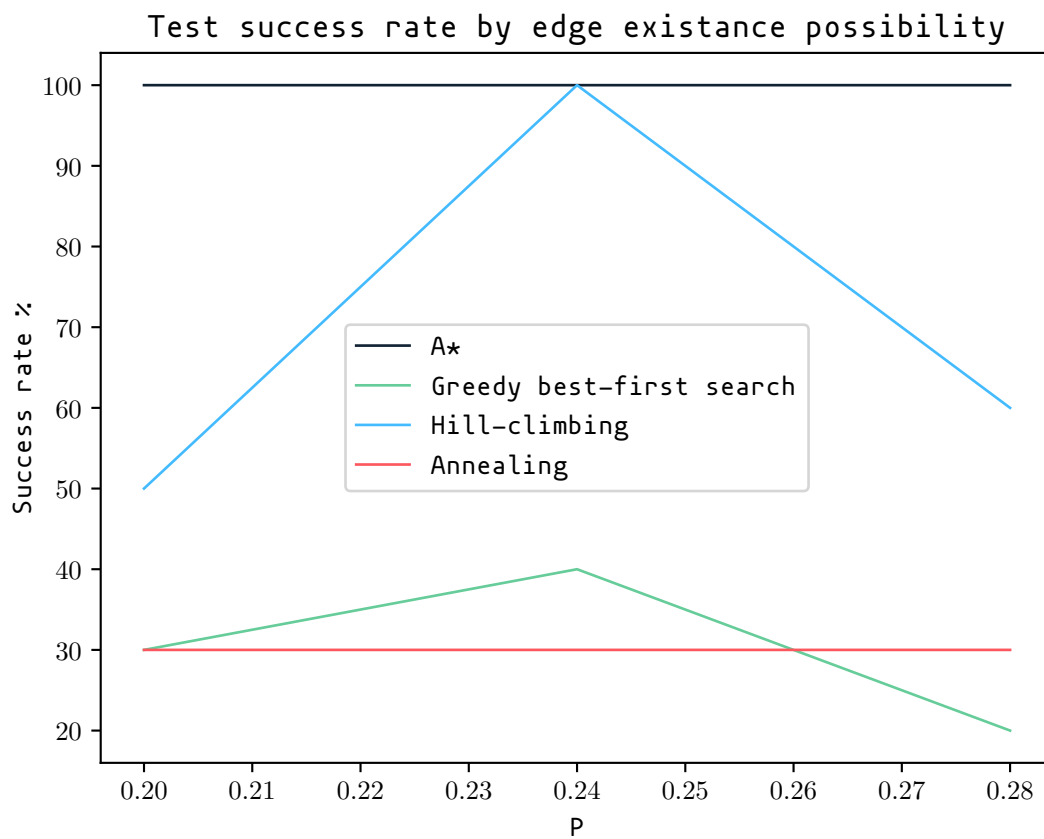


شکل ۳-۱۲:  $P = 0.12$

## فصل چهارم

### مقایسه روش‌های متفاوت

شکل ۴-۱: تعداد تکرارهای برنامه به نسبت احتمال وجود هر یال



شکل ۴-۲: درصد یافتن جواب درست با افزایش احتمال وجود هر یال