به نام خدا

دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

درس تحلیل شبکه‌های پیچیده

استاد حقیرچهرقانی

تمرین اول

علیرضا مازوچی

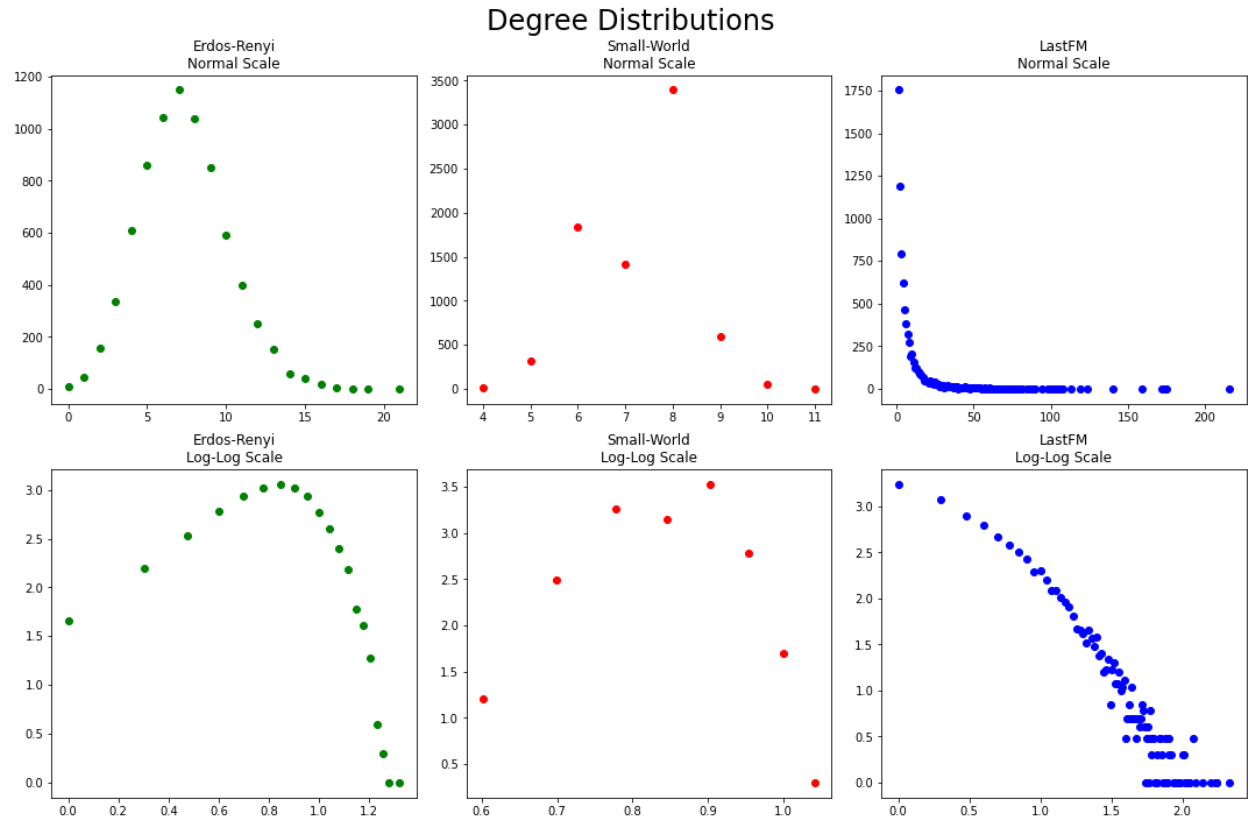
۴۰۰۱۳۱۰۷۵

سوال ۱

الف) برای تولید گراف تصادفی Erdos-Renyi یک راه آن است که احتمال p حساب گردد (تعداد یال‌های مدنظر تقسیم بر کل تعداد یال‌های ممکن) و متناسب با آن یال‌ها تولید شوند. اما در این رویکرد ممکن است تعداد یال‌های بدست آمده کمی متفاوت از تعداد یال‌های مدنظر باشد. راه دیگر که مورد استفاده من قرار گرفته است این است که ابتدا کل یال‌های ممکن را تولید کنیم و سپس به تعداد یال مدنظر از میان این مجموعه یال‌ها را انتخاب کنیم.

ب) برای ایجاد مدل تصادفی Small-World من ابتدا به هر گره یک عدد آیدی از صفر تا تعداد گره مدنظر نسبت دادم. سپس حساب کردم که هر راس باید چه درجه‌ای داشته باشد. قسمت صحیح درجه مدنظر را می‌توان به سادگی برآورده کرد؛ برای این هدف هر گره را به گره‌های بعدی (بر اساس آیدی) متصل کردم. برای حل کردن قسمت غیرصحیح درجه آمدم و آن را در صد ضرب کردم. این عدد نشان می‌دهد که از هر صد گره چه تعداد آن نیاز به یک یال دیگر دارد تا در مجموع میانگین درجه هر گره برابر با درجه مورد نظر شود. به همین ترتیب به بخشی از گره‌ها یک یال دیگر به صورت منظم اضافه کردم. درجه مدنظر تعداد اعشار بیشتری از دو رقم دارد ولی تقریبا تا اینجای کار میانگین درجه بسیار نزدیک به چیزی است که انتظار داشتیم. در گام بعد با در نظر گرفتن $p=0.05$ بخشی از یال‌ها را حذف کردم و سپس به صورت تصادفی افزودم. تعداد یال‌های حذف‌شده و اضافه‌شده تا حدی متفاوت است تا مقدار اعشار باقی‌مانده که در مرحله قبل نادیده گرفته شده است در این گام برطرف گردد.

د) در تصویر ۱ نمودارهای توزیع درجه برای هر سه گراف هم به صورت عادی و هم به صورت log-log ترسیم شده است:



تصویر 1

با بررسی و مقایسه نمودارها می‌توان به نتایج زیر دست پیدا کرد. پیش از هر چیز باید توجه کرد که تعداد درجات و گره‌ها برابر است و تحلیل‌ها عادلانه خواهد بود:

- گراف واقعی گره‌هایی با درجه بسیار بزرگ دارد (قسمت چپ نمودار log-log) در حالی که در نمودارهای تصادفی چنین چیزی دیده نمی‌شود. همچنین گره‌ها با درجه بسیار پایین یعنی صفر و نزدیک به آن در گراف واقعی بسیار زیاد است درحالی که در گراف‌های تصادفی این چنین نیست.
- درجات گره‌های Small-World بسیار محدود و شامل چندین مقدار خاص است ولی Erdos-Renyi رنج درجات بیشتری را دارد. گراف واقعی حتی از گراف Erdos-Renyi همزمان هم رنج بیشتری دارد چراکه درجات بسیار بالا و بسیار پایین را به خوبی پوشش داده است.
- توزیع گراف‌های تصادفی تقریبا شبیه نمودارهای نرمال است ولی گراف واقعی اصلا نرمال نیست و از نوع log-log است.

۵) در جدول ۱ این مقدار برای هر سه گراف گزارش شده است:

جدول 1

گراف	Erdos-Renyi	Small-Wrold	LastFM
ضریب خوشه‌بندی	۰.۰۰۰۶	۰.۵۴۱۴	۰.۲۱۹۴

سوال ۲

برای اثبات گزاره‌های این سوال مقادیر N را برابر با ۱۰۰، ۱۰۰۰، ۵۰۰۰، ۱۰۰۰۰، ۲۵۰۰۰، ۵۰۰۰۰ و ۱۰۰۰۰۰ قرار دادم تا تاثیر بزرگ‌شدن گراف بر افزایش دقت گزاره‌ها را اثبات کنم. برای هر N مقادیر p را به ترتیب برابر با $\frac{1}{4N}$، $\frac{1}{2N}$، $\frac{3}{4N}$، $\frac{1}{N}$، $\frac{0.75+0.25\ln(N)}{N}$، $\frac{0.5+0.5\ln(N)}{N}$، $\frac{0.25+0.75\ln(N)}{N}$، $\frac{\ln(N)}{N}$، $\frac{2\ln(N)}{N}$ و $\frac{4\ln(N)}{N}$ قرار دادیم تا بررسی کنیم که در هر رژیم بزرگترین مؤلفه چه بخشی از کل گره‌ها را شامل می‌شود.

پیش از هر چیز باید نکته‌ای در مورد ناحیه ۳ بیان شود و آن فرمول اندازه نسبی بزرگترین مولفه گراف در شکل ۲ تمرین است. مطابق این فرمول تنها $p-p_c$ برابر گره‌ها باید در بزرگترین مولفه باشد. این عدد برابر است با $p-\frac{1}{N}$. مقدار p برای آنکه در ناحیه سوم قرار گرفته باشد باید به شکل $\frac{x}{N}$ باشد به گونه‌ای که $1 \le x \le \ln(N)$. پس نسبت مذکور حداکثر $\frac{\ln(N)}{N}$ است. این مقدار عدد بسیار کمی است. در بی‌نهایت و برای Nهای بزرگ به صفر میل می‌کند و جدای از آن با دو ناحیه مجاور سازگار نیست. در ناحیه ۲ باید نسبت اندازه بزرگترین مولفه $N^{-\frac{1}{3}}$ باشد. این عدد از ناحیه ۳ بیشتر است در حالی که باید همواره کمتر باشد. از طرفی در مرز نواحی ۳ و ۴ هم باید روابط ناحیه ۳ و هم ۴ برآورده شود. در ناحیه ۴ ادعا می‌شود که یک مولفه شامل تمام گره‌ها وجود دارد و در ناحیه ۳ گفته می‌شود که تعداد محدودی در بزرگترین مولفه است که همچنان تناقض است. حتی شکل ۱ تمرین با این فرمول سازگار نیست. نتایج عملی هم ابدا آن را تایید نمی‌کند.

در جای دیگر من خوانده‌ام که در ناحیه ۳ حداکثر $\ln(N)$ تای گره‌ها در بزرگترین مولفه وجود نخواهد داشت. به نظر می‌رسد که این رابطه بدون مشکل باشد. با ترکیب

این موارد به نظر می‌رسد که فرمولی مانند $1-\frac{\ln(N)}{N}+p$ برای نسبت بزرگترین مولفه به مراتب فرمول بهتری باشد. بر این اساس در ادامه گزارش مبنا این رابطه جدید خواهد بود. نه رابطه‌ای که حتی از نظر تئوری برقراری آن امکان‌پذیر نیست.

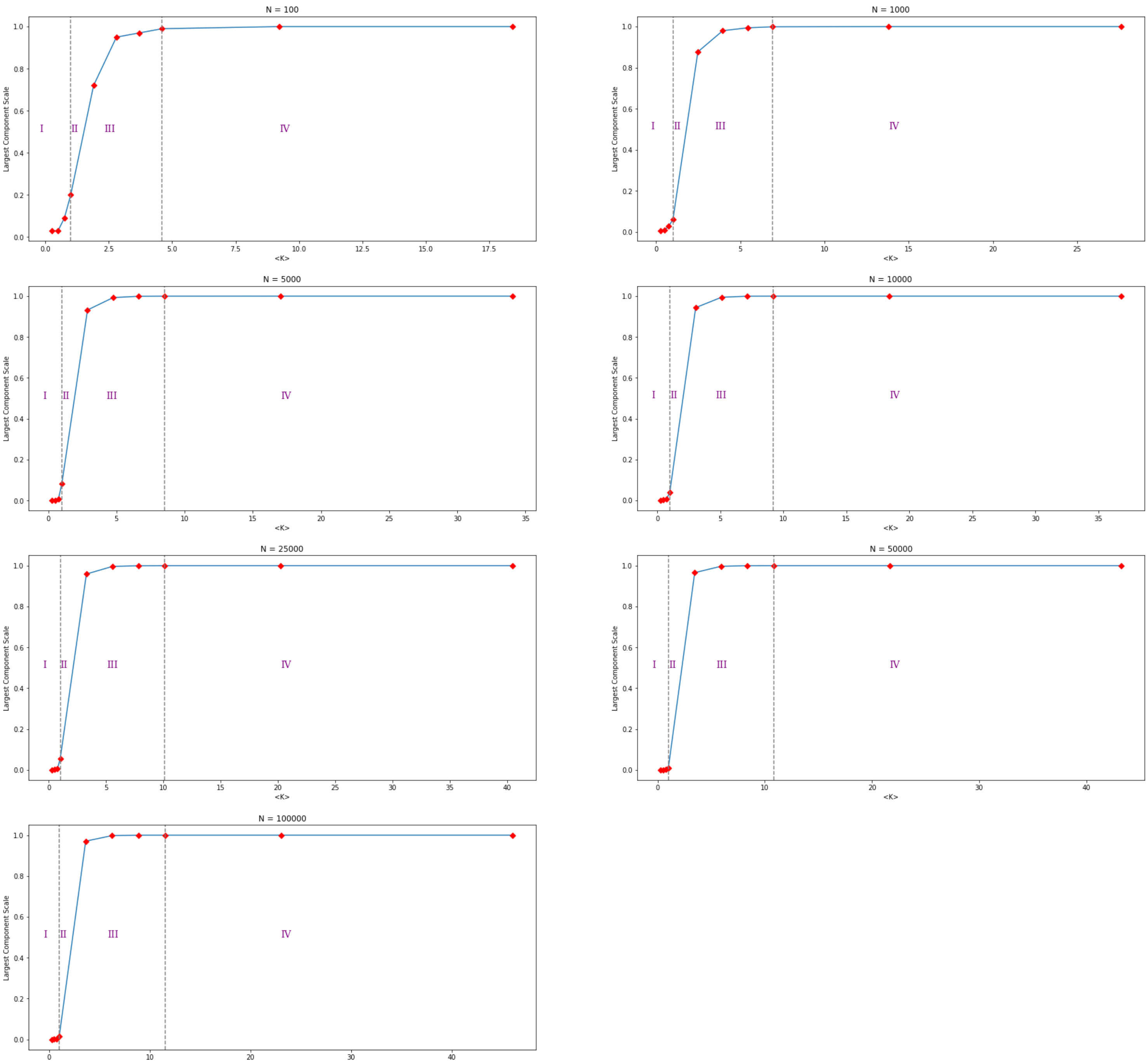
در تصویر ۲ تمام نتایج آورده شده است. در ناحیه ۱ در تمام شکل‌ها عملا هیچ مؤلفه بزرگی که بتواند شامل بخش قابل توجهی از گره‌ها شود وجود ندارد. در ناحیه ۲ که تنها می‌تواند شامل یک نقطه باشد در برخی از حالت‌ها یک مؤلفه نه چندان بزرگ شکل گرفته است. مثلا برای N=100 بزرگترین مولفه شامل ۰.۲ گره‌هاست که بسیار نزدیک به $N^{-\frac{1}{3}}$ است. در ناحیه ۳ به صورت بصری می‌توان دید که بزرگترین مولفه شامل بخش قابل توجهی از گره‌هاست و تنها بخش کمی از گره‌ها را شامل نمی‌شود. برای Nهای بزرگ تقریبا تمام گره‌ها در بزرگترین مولفه موجود است که این مشاهده با با رابطه موجود در صورت سوال اصلا نمی‌خواند ولی با رابطه جدید تشریح‌شده در پاراگراف قبل سازگار است. در ناحیه چهارم تقریبا تمام گره‌ها را می‌توانیم در بزرگترین مولفه داشته باشیم که منطقی است.

نهایتا سعی دارم تا نشان دهم که با افزایش Nدقت گزاره‌های بیان‌شده بیشتر می‌شود و در عین حال به صورت کمی و نه از روی تصاویر دقت خروجی را بدست آورم. برای هر N ده p وجود دارد که برای هر کدام یک نسبت بزرگترین مولفه در عمل و یک نسبت بزرگترین مولفه از نظر تئوری وجود دارد. لذا می‌توان یک خطایی را برای هر p گزارش کرد و مجموع ده خطا را به عنوان خطای نهایی هر N پیشنهاد کرد. در جدول ۲ این میزان خطا برای هر N مشخص شده است.

جدول 2

N	۱۰۰	۱۰۰۰	۵۰۰۰	۱۰۰۰۰	۲۵۰۰۰	۵۰۰۰۰	۱۰۰۰۰۰
خطا	۰.۴۰۷	۰.۲۰۴	۰.۱۰۷	۰.۰۷۵	۰.۰۷۰	۰.۰۵۶	۰.۰۴۰

همانطور که مشخص است به خوبی با افزایش N خطا کاهش پیدا کرده است و خطا برای بزرگترین N تنها ۰.۰۴ است که عدد مناسبی است.


N = 100
N = 1000
N = 5000
N = 10000
N = 25000
N = 50000
N = 100000
Largest Component Scale
<K>
I
II
III
IV

تصویر 2

سوال ۳

الف) با فرض آنکه بین هر دو گره حداکثر یک یال می‌توانیم داشته باشیم، هر گره در بخش اول (N_1 گره) حداکثر می‌تواند به تمام گره‌ها در بخش دوم (N_2 گره) متصل شود. پس بیشینه تعداد یال این مجموعه برابر است با $N_1 * N_2$

ب) دو بخشی بودن گراف تنها محدودیتی که برای اتصال یال‌ها ایجاد می‌کند آن است که گره‌های در یک بخش نمی‌توانند بهم متصل باشند. در بخش اول و در حالت کامل‌بودن حداکثر $\binom{N_1}{2}$ یال و در بخش دوم به طور مشابه $\binom{N_2}{2}$ یال می‌توانست وجود داشته باشد که در حالت دو بخشی ممنوع شده است. پس جواب این قسمت برابر است با $\binom{N_1}{2} + \binom{N_2}{2}$

ج) اگر در این قسمت فرض شده است که گراف دوبخشی فعلی دارای تمام یال‌ها است و قصد داریم تعداد یال‌ها در این حالت را نسبت به حالتی که یک گراف غیر دوبخشی کامل با همین تعداد گره داریم حساب کنیم می‌توان نوشت:

$$\frac{N_1 * N_2}{\binom{N_1 + N_2}{2}} = \frac{2 * N_1 * N_2}{(N_1 + N_2)(N_1 + N_2 - 1)} \approx \frac{2 * N_1 * N_2}{N_2 * N_2} = \frac{2N_1}{N_2}$$

د) هر یالی که در گراف دوبخشی وجود داشته، یک سرش در بخش اول و سر دیگرش در بخش دوم گراف است. پس مجموع درجات گره‌های هر دو بخش برابر است. مجموع درجات یک بخش برابر است با $N_i * k_i$. بنابراین می‌توان نوشت:

$$N_1 * k_1 = N_2 * k_2$$

سوال ۴

برای آنکه اجرای الگوریتم در زمان معقول امکان‌پذیر باشد، هم در این سوال و هم در سوال بعد هر یال را به صورت احتمالاتی با میزان احتمال ۰.۰۱ در نظر گرفتم. برای آنکه نتایج پایدارتر باشد مقدار realization را برابر با ۵ لحاظ کردم. با توجه به آنکه احتمال بسیار پایینی برای اجرا در نظر گرفته شده است، خروجی الگوریتم‌ها از نظر دقت چندان جالب نخواهند بود و در اجراهای محتلف می‌تواند کاملا متفاوت باشد ولی انتظار می‌رود از حالت تصادفی بهتر باشد. برای اثبات این مورد صد مرتبه مجموعه ده‌تایی تصادفی ایجاد کردم و میانگین نتایج را برای آن‌ها جمع‌آوری کردم.

در این سوال و سوال ۵ فرض شده است که الگوریتم حریصانه، الگوریتم Hill Climbing بدون Lazy Evaluation و با نادیده گرفتن هزینه‌ها باشد؛ چه این هزینه واقعا یکسان باشد چه نباشد. اما در CELF از Lazy Evaluation در هر دو حالت Hill Climbing بهره گرفته شده است.

در سوال ۴ هزینه انتخاب تمام گره‌ها با هم یکسان است. پس از منظر تئوری در این حالت و در الگوریتم CELF هر دو اجرا نباید تفاوت چندانی با هم داشته باشند. لذا انتظار می‌رود خروجی الگوریتم CELF و Hill Climbing کلاسیک مشابه باشد. اما از منظر سرعت CELF از Lazy Evaluation بهره می‌برد که خیلی موثر است. مطابق اسلاید درس انتظار می‌رود تسریع ۷۰۰ برابر مشاهده شود.

در جدول ۳ نتایج برای اجرای دو الگوریتم و حالت تصادفی آورده شده است.

جدول 3

	Random	Hill Climbing	CELF
زمان اجرا (ثانیه)	-	۴۵۱.۰۶	۴.۹۳
دقت خروجی	۱۹.۷۷	۳۳.۴	۱۴۴.۶

مطابق جدول و از نظر زمان مشاهده می‌شود که با رفتن از Hill Climbing به CELF نزدیک به صد برابر تسریع داشتیم که مطابق انتظار بود. از نظر دقت خروجی الگوریتم حریصانه توانسته است دقت را بهتر کند و به طرز غیر قابل انتظاری CELF

دقت بهتری داشته است. به نظر می‌رسد که این نتیجه به واسطه تصادف به این خوبی در آمده باشد و شاید قابل تکرار نباشد.

سوال ۵

برای این مسئله تمام شرایط مانند سوال ۴ است، فقط نیاز به بیان چندین ملاحظه است؛ در این سوال بودجه را برابر ۱۰ در نظر گرفته‌ام. همچنین در هر realization ده گره به تصادف به عنوان گره‌های آلوده در نظر گرفته شده است که باید توسط گره‌های پیشنهادی الگوریتم‌ها کشف شود تا جلوی شیوع آن گرفته شود.

در این سوال چون هزینه انتخاب گره‌ها باهم متفاوت است انتظار می‌رود که CELF علاوه بر بهبود سرعت (به واسطه Lazy Evaluation) بتواند بهبود در خروجی را هم به واسطه داشتن Hill Climbing نرمال‌شده بدست آورد. اما از طرفی باید این نکته را هم در نظر گرفت که گره‌های آلوده اولیه با احتمال کمی می‌توانند آلودگی را منتشر کنند. مطابق سوال ۴ احتمالا تنها ده گره جدید را بتوانند آلوده کنند (با توجه به عدد ۱۹.۷۷ در قسمت تصادفی) و گره‌های سنسور معمولا نمی‌توانند جلوی این تعداد آلودگی کم را بگیرند؛ به بیان دیگر شیوع بیشتر به واسطه احتمال پایین متوقف خواهد شد تا به واسطه سنسورهای انتخابی و در این شرایط احتمالا تفاوت چندانی بین الگوریتم‌ها نباشد. علاوه بر این‌ها بودجه ۱۰ برای این سوال خیلی کم است و تعداد گره نهایی انتخاب‌شده زیاد نخواهد بود. قاعدتا اگر بودجه خیلی بیشتری داشتیم باز جای بهبود دقت وجود داشت.

در جدول ۴ نتایج برای اجرای دو الگوریتم و حالت تصادفی آورده شده است.

جدول 4

	Random	Hill Climbing	CELF
زمان اجرا (ثانیه)	-	۹۵.۱۸	۱۳۳.۳۲
دقت خروجی	۴۶۸۷.۶۰	۴۶۸۹.۵۰	۴۶۹۰.۵۰

بهبود دقت به میزان ناچیز از حالت تصادفی به Hill Climbing و از Hill Climbing به CELF مشاهده شده است که مطابق انتظار بود. اما زمان اجرای بدتر CELF نسبت به Hill Climbing عجیب ولی قابل توجیه است. Lazy Evaluation زمانی تاثیر خود را نشان می‌دهد که تعداد گره انتخابی زیاد باشد، این در حالی است که با توجه به بودجه تعداد گره انتخابی خیلی کم است. از طرفی در CELF دو مرتبه Hill Climbing اجرا می‌شود که خود یک ضریب دو در زمان اجراست.

نهایتا نوبت به تشریح الگوریتم‌ها و بهبود سرعت آن می‌رسد. در الگوریتم Hill Climbing ابتدا میزان سود هر گره بدست می‌آید و بهترین آن انتخاب می‌شود. سپس در برای انتخاب گره بعد سود حاشیه‌ای تمام گره‌ها حساب می‌شود و گره‌ای برگزیده می‌شود که بیشترین سود حاشیه‌ای را به مجموعه انتخابی اضافه کند تا زمانی که بودجه به پایان برسد.

الگوریتم CELF با دو ایده این روش حریصانه را بهبود داده است. یک ایده برای بهبود دقت و یک ایده برای بهبود سرعت. در الگوریتم Hill Climbing کلاسیک اگر هزینه انتخاب گره‌ها متفاوت باشد الگوریتم دچار مشکل می‌شود و حد تئوری دقت خود یعنی $(1-\frac{1}{e})$ حالت بهینه را از دست می‌دهد. می‌توان یک نسخه دیگر از Hill Climbing ارائه داد که در آن همه چیز مشابه است به غیر از آن که سود حاشیه‌ای تقسیم بر هزینه انتخاب گره شود. در این حالت هم حد تئوری وجود نخواهد داشت ولی اگر الگوریتم Hill Climbing دو بار اجرا شود؛ یک بار با نرمال‌سازی وزن و یک بار بدون آن و بهترین این دو حالت برگردانده شود حد تئوری $\frac{1}{2}\left(1-\frac{1}{e}\right)$ اثبات می‌شود.

ایده دوم الگوریتم CELF استفاده از Lazy Evaluation که سرعت اجرا را می‌تواند خیلی بهتر کند. این ایده بر اساس ویژگی sub modularity کار می‌کند. مطابق این ویژگی و به طور خلاصه می‌توان دید که سود حاشیه‌ای انتخاب هر گره در طول اجرای الگوریتم کمتر می‌شود. بنابراین اگر در یک لحظه انتخاب گره‌ای سود حاشیه‌ای بیشتری از سود حاشیه‌ای قدیمی یک گره دیگر داشته باشد قطعا گره اول بر گره دوم ارجحیت دارد. با این منطق در CELF ابتدا یک لیست از سود هر گره به صورت مرتب ایجاد می‌شود و اولین گره آن برگزیده. سپس در گام‌های بعد اولین گره لیست با بیشترین سود حاشیه‌ای برداشته می‌شود و مقدار سود حاشیه‌ای آن بروز

می‌شود؛ اگر همچنان این سود حاشیه‌ای بیشینه باشد نیاز به محاسبه سود حاشیه‌ای مابقی نیست ولی اگر مقدار سود حاشیه‌ای جدید از سود حاشیه‌ای قبلی چندین گره کمتر باشد لازم است آن چند گره (و نه همه) نیز بررسی شود. بدین ترتیب در هر گام سود حاشیه‌ای تعداد کمی از گره‌ها و نه همه آن‌ها بررسی می‌گردد.