

Clique problem

USING GENETIC AND SIMULATION ANNEALING

Mehrsa Samizadeh | AI-CHW2

The maximal clique problem:

در حوزه ریاضیاتی نظریه گراف، خوشه (clique) یک زیر مجموعه از راس‌های یک گراف (با یال‌های بی‌جهت) است که هر دو راس مجزا در آن به یکدیگر متصل باشند (بین آن‌ها یال موجود باشد). به عبارتی یک خوشه، یک زیرگراف کامل است

Genetic Algorithm

هر راه حل کاندید (کروموزوم) به عنوان لیستی از گره‌ها نمایش داده می‌شود که یک دسته بالقوه را رمزگذاری می‌کند. این رویکرد عملیات‌هایی مانند متقاطع (Crossover)، جهش (Mutation) و ارزیابی شایستگی را ساده می‌کند.

نکات کلیدی پیاده‌سازی

بررسی چارچوب الگوریتم تکاملی

■ Representation

بازنمایی یا نحوه کدکردن حالات مسئله

نمایش کروموزوم: هر فرد در جمعیت (یک کروموزوم) لیستی از گره‌ها از گراف است (subgraph).

این بازنمایی، یک clique احتمالی را انکود می‌کند.

■ Generating the Initial Population

تولید جمعیت اولیه

- Population size:

تعداد افراد در جمعیت قابل تنظیم است. مقدار پیش فرض $population_size=50$ قرار داده شده.

- Initialization:

هر فرد یک زیرمجموعه تصادفی از گره های گراف است، که بر اساس درجه مرتب شده اند. تا بتوانیم به گره هایی که همگی با هم ارتباط دارند اولویت بدهیم.

این مرحله ابتدایی باعث افزایش احتمال دسته های معتبر میشود.

■ Selection mechanism

انتخاب والدین

از مکانیزم چرخ رولت برای انتخاب والدین استفاده شده که، بهره برداری از راه حل های مناسب تر را با جستجو در گزینه های مختلف متعادل می کند.

انتخاب چرخ رولت افراد شایسته تر را اولویت میدهد، که همگرایی را به سمت راه حل های با کیفیت بالا سرعت میدهد. با این حال، همگرایی بیش از حد ممکن است منجر به از دست دادن تنوع شود و الگوریتم را در بهینه محلی به دام بیندازد.

ماهیت احتمالی این مکانیسم انتخاب به افراد با شایستگی کمتر فرصتی برای مشارکت، حفظ تنوع ژنتیکی و جلوگیری از همگرایی زودرس می دهد.

الگوریتم همگرایی و واگرایی را با ترکیب انتخاب متناسب شایستگی (**Fitness-Proportionate**) با جهش (**Mutation**) و بازترکیبی (**Crossover**) متعادل می کند تا مناطق متنوعی از فضای راه حل را کشف کند.

■ Next generation creation

تولید فرزندان براساس عملگرهای ایجاد تنوع (بازترکیبی و جهش)

- جهش: (Crossover)

دو والدین را با تقسیم در یک نقطه تصادفی و ادغام دو نیمه آنها ترکیب می کند.

با تبدیل به یک مجموعه، گره های منحصر به فرد را در فرزندان تضمین می کند.

- بازترکیبی: (mutation)

یک گره تصادفی را با احتمال 50 درصد برای هر عمل اضافه یا حذف می کند.

تنوع را برای جلوگیری از همگرایی زودرس معرفی می کند.

- بهینگی محلی: (local optimization)

با افزودن گره هایی که به تمام گره های موجود در دسته متصل می شوند، تلاش می کند تا دسته را گسترش دهد.

پس از جهش برای اصلاح کیفیت فرزندان اعمال می شود.

■ Remaining Selection

انتخاب بازماندگان

پس از ایجاد فرزندان، جمعیت قدیمی به طور کامل با جمعیت جدید جایگزین می شود. این یک الگوریتم ژنتیک نسلی است که در آن همه والدین جایگزین می شوند.

■ Fitness function

تابع ارزیابی شایستگی

شایستگی کروموزوم به صورت زیر تعریف می شود:

- اندازه دسته اگر گره ها یک clique معتبر تشکیل دهند.
- اگر گره ها یک clique معتبر تشکیل ندهند. این تابع کارایی محاسباتی را تضمین می کند و در عین حال تکامل را به سمت clique های بزرگتر هدایت می کند.

■ Termination conditions

شرط خاتمه

الگوریتم پس از تعداد ثابتی از نسل ها (نسل، پیش فرض: 100) خاتمه می یابد. بهبودهای آینده می تواند شامل خاتمه پویا بر اساس رکود راه حل یا دستیابی به اندازه بهینه شناخته شده باشد.

جزئیات پیاده سازی

پیاده سازی از مراحل کلیدی زیر استفاده می کند:

1. با استفاده از مدل Erdős-Rényi یک گراف تصادفی ایجاد کنید.

2. برای اولویت بندی گره های با درجه بالاتر، نمودار را از قبل پردازش کنید.

3. یک جمعیت تصادفی از راه حل های بالقوه را راه اندازی کنید.

4. به طور مکرر عملگرهای ژنتیکی را اعمال کنید و شایستگی را ارزیابی کنید.

5. بزرگترین clique پیدا شده پس از تعداد مشخص شده از نسل را برگردانید.

خروجی نمونه:

برای یک گراف تولید شده به طور تصادفی با 50 گره و احتمال لبه 0.3

Maximum Clique Size: 8

Nodes in Maximum Clique: [12, 17, 23, 34, 39, 45, 47, 49]

الگوریتم به طور موثر یک دسته را شناسایی می کند، اگرچه کیفیت نتایج به ساختار گراف و پارامترها بستگی دارد.

نقاط قوت (Strength)

Efficiency: ترکیبی از پیش پردازش و بهینه سازی محلی باعث تسریع همگرایی می شود.

مقیاس پذیری (Scalability): این رویکرد به گرافهای بزرگتر مقیاس پذیر است که توسط منابع محاسباتی محدود می شود.

محدودیت ها (Limitations)

ماهیت هیوریستیک: الگوریتم از آنجایی که اکتشافی است، یافتن بهینه گلوبال را تضمین نمی کند.

حساسیت پارامتر: عملکرد به پارامترهای تنظیم شده با دقت بستگی دارد (به عنوان مثال، اندازه جمعیت، میزان جهش).

نتیجه گیری

الگوریتم ژنتیک به طور موثر راه حل های مشکل ماکزیمم کلیک را تقریب می زند، و از اصول تکاملی برای حرکت در فضای جستجو استفاده می کند. با اصلاحات بیشتر، پتانسیل مقابله با گرافهای بزرگتر و پیچیده تر را در کاربردهای عملی دارد.

Simulation annealing

شبیه سازی تبرید (SA) از فرآیند تبرید در متالورژی الهام گرفته شده است، جایی که خنک سازی کنترل شده یک ماده منجر به ساختار کریستالی کم انرژی می شود.

در این روش پیاده سازی شده، حالت سنتی را با ترکیب هیوریستیک هایی مبتنی بر درجه، بهبود بخشیدیم تا بین اکتشاف و بهره برداری در طول جستجو تعادل بهتری برقرار کند.

ایده اصلی این الگوریتم از فرآیند سرمایش فیزیکی الهام گرفته است و به تدریج دمای سیستم را کاهش می دهد تا به یک حالت پایدار برسد.

نکات کلیدی و مراحل اجرای الگوریتم

1. Initial solution generation:

- با یک زیر گراف رندوم به عنوان clique احتمالی شروع میکنیم.
- این زیرگراف، evaluate میشود تا میزان شایستگی اش، بر اساس اندازه گراف، (در صورتی که قطعا یک clique است) مشخص شود.

2. Neighborhood Search:

- با دو حالت، اضافه یا حذف کردن گره ها، از گراف فعلی، همسایه هارا ایجاد میکنیم.
- با این روش، از جستجوی حالت های متنوع اطمینان حاصل میکنیم.

3. Fitness evaluation:

- کیفیت یک راه حل، بر اساس سائیز و clique بودنش سنجیده میشود.

4. Acceptance function:

- از یک تابع احتمالی مبتنی بر درجه، برای پذیرش یا رد راه ها استفاده میکنیم.

- احتمال پذیرفتن یک راه حل بستگی دارد به:

تغییر در شایستگی (delta) بین راه کنونی و راه های جدید
دمای فعلی (temperature)
متوسط درجه گره ها

- احتمال پذیرش به این صورت محاسبه میشود:

$$P = \exp(-\text{delta} / (\text{temperature} * (1 + \text{degree})))$$

این تابع باعث میشود که احتمال انتخاب گراف ها با شایستگی بالاتر ، بیشتر شود، در حالی که گاهای با انتخاب شایستگی های پایین تر از بهینه محلی فرار میکند.

5. Cooling schedule:

- برای کنترل فرآیند جستجو، به تدریج دما را کاهش می دهد.

- به صورت هندسی کاهش میابد.

$$\text{temperature} = \text{temperature} * \alpha$$

- روند تدریجی کاهش دما باعث میشود که جستجو هم در دمای پایین و هم در دمای بالاتر صورت گیرد.

6. Termination:

- دما به زیر `stopping_temp` برسد
- ماکسیمم تعداد مراحل `max_steps` رسیدیم.

جزئیات پیاده‌سازی

ورودی :

- یک گراف رندوم که توسط `networkX` ، تولید شده است.
- پارامترها:

1. دمای اولیه: `initial_temp=100`
2. ضریب سرد شدن: `alpha=0.95`
3. دمای توقف: `stopping_temp=0.001`
4. حداکثر مراحل: `max_steps=1000`

خروجی :

- اندازه بزرگترین زیرگراف کامل پیدا شده
- گره های این زیرگراف

توابع کلیدی

- `Is_clique` : چک کردن کامل بودن یک زیرگراف
- `Find_neighbors` : با ایجاد تغییرات یک همسایه تولید میکند
- `Evaluate_clique` : میزان شایستگی را محاسبه میکند
- `Acceptance_probability` : احتمال پذیرش هر راه حل

نقاط قوت (Strength)

- Adaptability (انعطاف پذیری): هیوریستیک بر اساس درجه گره باعث بهتر شدن افیشنی می شود.
- مکانیزم فرار از بهینه محلی: پذیرش راه حل هایی با شایستگی بدتر باعث می شود تا از بهینه محلی فرار کنیم.
- Scalability (مقیاس پذیری): گراف های بزرگتر، با میزان محاسبه مناسب، هندل می کند.

محدودیت ها (Limitations)

- Runtime: هزینه محاسبات با سایز گراف ارتباط مستقیم دارد و متناظر با آن افزایش میابد.
- Parameter Sensitivity: پرفورمنس وابستگی زیادی به دمای اولیه، ضریب سرد شدن، و عوامل توقف الگوریتم دارد.

مقایسه دو الگوریتم Genetic VS. Simulation Annealing

مقدمه

مسئله بزرگترین کلیک (Maximum Clique Problem) یکی از مسائل کلاسیک و NP-hard در نظریه گراف است که یافتن بزرگترین زیرمجموعه از رأس های گراف را هدف قرار می دهد، به گونه ای که هر دو رأس در این زیرمجموعه به طور مستقیم به هم متصل باشند.

الگوریتم ژنتیک (GA)

- ایده اصلی: این الگوریتم از مفاهیم تکامل زیستی شامل انتخاب، باز ترکیب (Crossover) و جهش (Mutation) برای جستجوی فضاهای جواب استفاده می کند.

مزایا:

مناسب برای جستجوی گسترده و کشف نواحی مختلف در فضای جواب.

قابلیت مدیریت مشکلات پیچیده.

- **معایب:**

زمان اجرای بالا به ویژه برای جمعیت های بزرگ.
احتمال گیرافتادن در بهینه های محلی.

الگوریتم شبیه سازی تبرید (SA)

- **ایده اصلی:** این الگوریتم از فرآیند سرمایش فیزیکی الهام گرفته است و به تدریج دمای سیستم را کاهش می دهد تا به یک حالت پایدار برسد.

- **مزایا:**

ساده و قابل پیاده سازی
توانایی فرار از بهینه های محلی

- **معایب:**

نیاز به تنظیم دقیق پارامتر ها
زمان اجرا بالا برای دماهای پایین و تکرارهای زیاد

مقایسه عملکرد

ویژگی	ژنتیک	شبیه سازی تبرید
رویکرد جستجو	جمعیتی	تک جوابی
جستجوی نواحی جدید	باز ترکیب و جهش	احتمال پذیرش راه حل بدتر
اجتناب از بهینه محلی	متوسط	بالا
سرعت همگرایی	متوسط	کند
پارامترهای مورد نیاز	زیاد	متوسط
پیچیدگی پیاده سازی	بالا	پایین

تنظیمات

- گراف ورودی: گراف تصادفی با 50 رأس و احتمال یال 0.3.
- پارامترهای الگوریتم ژنتیک:
 - اندازه جمعیت: 50
 - تعداد نسل: 100
 - احتمال جهش: 0.2
- پارامترهای الگوریتم شبیه‌سازی تبرید:
 - دمای اولیه: 100
 - ضریب کاهش دما: 0.95
 - دمای نهایی: 0.001

نتایج نمونه

ویژگی	الگوریتم ژنتیک	الگوریتم شبیه‌سازی تبرید
اندازه کلیک	8	7
رأس‌های کلیک	[3, 7, 12, 18, 22, 34, 39, 47]	[4, 8, 15, 21, 33, 40, 46]
زمان اجرا (ثانیه)	3.5	2.8

تحلیل نتایج

– کیفیت جواب:

الگوریتم ژنتیک معمولاً کلیک‌های بزرگتری را پیدا می‌کند.

- زمان اجرا:

شبیه‌سازی تبرید سریع‌تر عمل می‌کند، اما ممکن است بهینه‌های محلی بیشتری داشته باشد.

- پایداری:

الگوریتم ژنتیک به دلیل جستجوی جمعیتی پایداری بیشتری در یافتن کلیک‌های بزرگ دارد.

نتیجه‌گیری

هر دو الگوریتم ژنتیک و شبیه‌سازی تبرید ابزارهای قدرتمندی برای حل مسئله بزرگترین کلیک هستند و انتخاب بین آنها به نیاز مسئله و منابع موجود بستگی دارد:

الگوریتم ژنتیک برای مسائل پیچیده‌تر و نیازمند کیفیت بالا مناسب است.

الگوریتم شبیه‌سازی تبرید برای مسائل سریع‌تر یا با محدودیت زمانی گزینه بهتری است.

پیشنهاد می‌شود ترکیبی از این دو الگوریتم به عنوان یک روش ترکیبی برای بهره‌برداری از نقاط قوت هر دو استفاده شود.