



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

گزارش چهارم درس هوش مصنوعی

نگارش

آرمان صالحی

استاد راهنما

دکتر مهدی قطعی

بهار 1403

چکیده

این کد یک برنامه کامل برای حل مسئله رنگ آمیزی نقشه با استفاده از CSP است که قابلیت‌های مختلفی برای بهینه‌سازی و کاهش دامنه‌های متغیرها دارد. این برنامه به کاربر اجازه می‌دهد تنظیمات مختلفی را از طریق خط فرمان مشخص کند و سپس نتایج را به صورت گرافیکی نمایش می‌دهد.

واژه‌های کلیدی:

CSP – GeoDataFrame – AC-3 – LCV – LMV – BackTracking - MRV

صفحه

فهرست مطالب

چکیده أ

گزارش کار 3

گزارش کار

فایل CSP

کلاس CSP: این کلاس و متدهای آن به شما امکان می‌دهد تا مسئله‌های رضایت‌بخشی محدودیت‌ها را تعریف و حل کنید. شما می‌توانید متغیرها و محدودیت‌های مورد نظر خود را به مسئله اضافه کنید، مقداردهی متغیرها را انجام دهید و بررسی کنید که آیا تخصیص‌های شما با محدودیت‌ها سازگار هستند یا خیر.

متد __init__: این متد سازنده‌ی کلاس است که متغیرها و محدودیت‌های اولیه را مقداردهی می‌کند.

- **variables:** دیکشنری برای ذخیره متغیرها و دامنه‌های آن‌ها.
- **constraints:** لیستی از محدودیت‌ها.
- **var_unassigned:** لیستی از متغیرهایی که هنوز مقداردهی نشده‌اند.
- **var_constraints:** دیکشنری برای ذخیره محدودیت‌های مرتبط با هر متغیر.
- **assignments:** دیکشنری برای ذخیره تخصیص‌های انجام شده.
- **assignments_number:** شمارنده‌ی تعداد تخصیص‌ها.

متد add_constraint:

این متد یک محدودیت را به مسئله اضافه می‌کند.

متد add_variable:

این متد یک متغیر را با دامنه‌ی آن به مسئله اضافه می‌کند.

متد assign:

این متد مقداردهی به یک متغیر را انجام می‌دهد و بررسی می‌کند که آیا این تخصیص با محدودیت‌ها سازگار است یا خیر.

متد is_consistent:

این متد بررسی می‌کند که آیا تخصیص یک مقدار به یک متغیر، محدودیت‌ها را نقض می‌کند یا خیر.

متد is_complete:

این متد بررسی می‌کند که آیا همه متغیرها مقداردهی شده‌اند یا خیر.

متد is_assigned:

این متد بررسی می‌کند که آیا یک متغیر مقداردهی شده است یا خیر.

متد unassign:

این متد مقداردهی یک متغیر را لغو می‌کند و مقادیر دامنه‌ی آن را بازیابی می‌کند.

متد changing_domains:

این متد دامنه‌های متغیرها را به‌روزرسانی می‌کند و مقادیر حذف شده را بازیابی گرداند.

متد restore_domains:

این متد مقادیر حذف شده از دامنه‌ها را بازیابی می‌کند.

فایل graphics

این کد شامل دو تابع است که برای ترسیم نقشه‌ای رنگی از یک قاره خاص بر اساس یک راه‌حل مسئله رنگ‌آمیزی نقشه و داده‌های جغرافیایی استفاده می‌شود. در اینجا توضیحات دقیقی از هر بخش از کد آورده شده است:

1. تابع draw_colored_map

این تابع نقشه‌ای از یک قاره خاص را با استفاده از راه‌حل داده شده رنگ‌آمیزی می‌کند و تعداد تخصیص‌های انجام شده را نمایش می‌دهد.

آرگومان‌ها:

- solution (Dict[str, str]): یک دیکشنری که در آن کدهای ISO A3 کشورها به رنگ‌های تخصیص داده شده‌شان نگاشت شده‌اند.

- gdf (gpd.GeoDataFrame): یک GeoDataFrame شامل داده‌های جغرافیایی کشورها، شامل قاره‌ها و کدهای ISO A3 آن‌ها.

- continent (str): نام قاره‌ای که باید ترسیم شود.

- assignments_number (int): تعداد تخصیص‌های متغیر انجام شده در طول فرایند حل مسئله.

مراحل:

1. فیلتر کردن قاره: ابتدا GeoDataFrame برای قاره‌ی مورد نظر فیلتر می‌شود و یک کپی از داده‌های آن قاره ساخته می‌شود.
2. تخصیص رنگ‌ها: برای هر کشور در قاره، رنگی که در دیکشنری solution تخصیص داده شده است، انتخاب می‌شود. اگر کشوری در solution نباشد، رنگ خاکستری روشن (lightgrey) به آن تخصیص داده می‌شود.
3. تنظیمات نقشه: نقشه‌ی قاره با استفاده از رنگ‌های تخصیص داده شده ترسیم می‌شود و مرز کشورها با رنگ سیاه مشخص می‌شود.
4. تنظیم محدوده‌های نقشه: محدوده‌ی نمایش نقشه برای قاره‌ی اروپا و سایر قاره‌ها تنظیم می‌شود.
5. اضافه کردن متن‌ها: نام کشورها (کدهای ISO A3) در مرکز جغرافیایی آن‌ها نوشته می‌شود و تعداد تخصیص‌ها در بالا و چپ نقشه نمایش داده می‌شود.
6. نمایش نقشه: نقشه با استفاده از plt.show() نمایش داده می‌شود.

2. تابع draw

این تابع داده‌های جغرافیایی را از یک فایل CSV بارگیری می‌کند، آن‌ها را به یک GeoDataFrame تبدیل می‌کند و سپس تابع draw_colored_map را برای ترسیم نقشه فراخوانی می‌کند.

آرگومان‌ها:

- continent (str): نام قاره‌ای که باید ترسیم شود.
- solution (Dict[str, str]): دیکشنری که در آن کدهای ISO A3 کشورها به رنگ‌های تخصیص داده شده‌شان نگاشت شده‌اند.
- assignments_number (int): تعداد تخصیص‌های متغیر انجام شده در طول فرایند حل مسئله.

مراحل:

1. بارگیری داده‌ها: داده‌های جغرافیایی از یک فایل CSV به یک DataFrame پانداس بارگیری می‌شود.
2. تبدیل داده‌ها: داده‌های جغرافیایی از فرمت WKT به شیء‌های هندسی Shapely تبدیل می‌شوند.
3. ساخت GeoDataFrame: DataFrame به یک GeoDataFrame تبدیل می‌شود.

4. فراخوانی تابع `draw_colored_map`: این تابع با استفاده از پارامترهای ورودی فراخوانی می‌شود تا نقشه‌ی رنگی ترسیم شود.

جمع‌بندی

این کد به طور کلی برای حل و نمایش یک مسئله رنگ‌آمیزی نقشه طراحی شده است. ابتدا داده‌های جغرافیایی بارگیری و به فرم مناسب تبدیل می‌شوند، سپس نقشه‌ای از قاره مورد نظر ترسیم می‌شود که در آن کشورها با رنگ‌های تخصیص داده شده نمایش داده می‌شوند و تعداد تخصیص‌های انجام شده نیز نمایش داده می‌شود.

فایل `map_generator`

این کد تابعی را تعریف می‌کند که یک دیکشنری ایجاد می‌کند که هر کشور در یک قاره خاص را به یک لیست از کدهای ISO A3 کشورهای همسایه آن کشور نگاشت می‌کند. این دیکشنری با استفاده از داده‌های جغرافیایی و اطلاعات همسایگی از یک فایل CSV ایجاد می‌شود.

تابع `generate_borders_by_continent`

این تابع داده‌های جغرافیایی و همسایگی را از یک فایل CSV بارگیری می‌کند، آن‌ها را بر اساس قاره مورد نظر فیلتر می‌کند و سپس دیکشنری‌ای ایجاد می‌کند که هر کشور را به لیستی از کدهای ISO A3 کشورهای همسایه‌اش در همان قاره نگاشت می‌کند.

آرگومان‌ها:

- `continent (str)`: نام قاره‌ای که می‌خواهیم مرزها و همسایگی‌های آن را تولید کنیم.

خروجی:

- `Dict[str, List[str]]`: یک دیکشنری که در آن کلیدها کدهای ISO A3 کشورها هستند و مقادیر لیستی از کدهای ISO A3 کشورهای همسایه در همان قاره است.

مراحل:

1. بارگیری داده‌ها: داده‌ها از فایل `countries_dataset.csv` بارگیری می‌شوند و ستون `geometry` به نوع هندسی تبدیل می‌شود.

2. فیلتر کردن بر اساس قاره: داده‌ها بر اساس قاره مورد نظر فیلتر می‌شوند.
3. ایجاد دیکشنری همسایگی: دیکشنری‌ای ایجاد می‌شود که در آن هر کشور به لیستی از کشورهای همسایه‌اش نگاشت می‌شود.

توضیحات جزئی:

1. بارگیری و تبدیل داده‌ها:
- فایل CSV شامل اطلاعات کشورها بارگیری می‌شود.
- ستون geometry که شامل داده‌های جغرافیایی به صورت WKT (Well-Known Text) است، به اشیاء هندسی تبدیل می‌شود.
2. فیلتر کردن بر اساس قاره:
- داده‌ها برای قاره‌ی مشخص شده فیلتر می‌شوند.
3. ایجاد دیکشنری همسایگی:
- برای هر کشور در قاره‌ی فیلتر شده، لیستی از کشورهای همسایه استخراج می‌شود.
- اگر ستون neighbors شامل رشته‌ای از کدهای ISO A3 کشورهای همسایه باشد، آن‌ها به صورت لیست جدا می‌شوند.
- کشور و لیست همسایه‌های آن به دیکشنری borders اضافه می‌شوند.
این تابع در نهایت دیکشنری‌ای ایجاد می‌کند که می‌تواند در حل مسائل مختلف از جمله مسائل رنگ‌آمیزی نقشه استفاده شود.

فایل solver

این کد یک کلاس Solver را پیاده‌سازی می‌کند که برای حل مسائل مربوط به "مسئله رضایت محدودیت‌ها" (Constraint Satisfaction Problem یا CSP) طراحی شده است. کلاس شامل روش‌های مختلفی برای انتخاب متغیرها و مقادیر، استفاده از الگوریتم AC-3 برای کاهش دامنه‌ها، و پیاده‌سازی الگوریتم برگشت‌پذیری (Backtracking) برای پیدا کردن راه‌حل‌های ممکن است. بیایید بخش‌های مختلف این کد را به طور کامل توضیح دهیم:

1. تعریف کلاس Solver

- این متد سازنده کلاس است که پارامترهای مختلفی را دریافت می کند:
- csp: شیئی از کلاس CSP که شامل مسئله رضایت محدودیت هاست.
- domain_heuristics: فلگ برای استفاده از هیوریستیک های دامنه.
- variable_heuristics: فلگ برای استفاده از هیوریستیک های متغیر.
- AC_3: فلگ برای استفاده از الگوریتم AC-3.

2. الگوریتم برگشت پذیری (Backtracking)

- این متد الگوریتم برگشت پذیری را پیاده سازی می کند:
- اگر AC-3 فعال باشد، دامنه های متغیرها را کاهش می دهد.
- بررسی می کند که آیا مسئله کامل شده است یا نه.
- یک متغیر تخصیص نیافته را انتخاب می کند.
- برای هر مقدار ممکن از دامنه، بررسی می کند که آیا تخصیص آن مقدار با محدودیت ها سازگار است یا نه.
- اگر سازگار باشد، آن را تخصیص می دهد و به صورت بازگشتی الگوریتم را فراخوانی می کند.
- در صورت عدم موفقیت، تخصیص ها را بازمی گرداند و دوباره تلاش می کند.

3. انتخاب متغیر تخصیص نیافته

- این متد یک متغیر تخصیص نیافته را انتخاب می کند:
- اگر هیوریستیک متغیر فعال باشد، از هیوریستیک "کمترین مقدار باقی مانده" (MRV) استفاده می کند.
- در غیر این صورت، اولین متغیر تخصیص نیافته را انتخاب می کند.

4. ترتیب دهی مقادیر دامنه

- این متد مقادیر دامنه یک متغیر را به ترتیب خاصی برمی گرداند:
- اگر هیوریستیک دامنه فعال باشد، از هیوریستیک "کمترین مقدار محدود کننده" (LCV) استفاده می کند.
- در غیر این صورت، مقادیر دامنه را به همان ترتیب اصلی بازی گرداند.

5. کاهش دامنه ها با الگوریتم AC-3

- این متد دامنه متغیر 'x' را بر اساس محدودیت های بین 'x' و 'y' کاهش می دهد:
- اگر هیچ مقدار در دامنه 'x' با هیچ مقدار در دامنه 'y' سازگار نباشد، مقدار از دامنه 'x' حذف می شود.
- مقادیر حذف شده را برمی گرداند.
- این متد الگوریتم AC-3 را برای کاهش دامنه های متغیرها پیاده سازی می کند:
- محدودیت ها را در صف قرار می دهد و به تدریج بررسی می کند.
- دامنه متغیرها را بر اساس محدودیت ها کاهش می دهد.
- مقادیر حذف شده را برمی گرداند.

6. هیوریستیک "کمترین مقدار باقی مانده" (MRV)

- این متد متغیری را انتخاب میکند که کمترین مقادیر باقیمانده را در دامنه خود دارد.

7. هیوریستیک "کمترین مقدار محدود کننده" (LCV)

- این متد مقادیر دامنه یک متغیر را بر اساس هیوریستیک LCV مرتب می کند:

- مقادیری را انتخاب می‌کند که کمترین محدودیت‌ها را برای متغیرهای دیگر ایجاد می‌کنند.

فایل main

این برنامه به کاربر اجازه می‌دهد که یک قاره را انتخاب کند و تنظیمات مختلفی مانند استفاده از هیوریستیک‌ها و الگوریتم‌های بهینه‌سازی را مشخص کند. سپس، برنامه نقشه را رنگ‌آمیزی می‌کند و نتایج را نمایش می‌دهد.

1. ایمپورت‌ها و تعریف کلاس‌ها

- این خطوط کتابخانه‌های مورد نیاز و ماژول‌های مختلف را وارد می‌کنند:
- ``argparse`` برای تجزیه و تحلیل آرگومان‌های خط فرمان.
- ``Enum`` برای تعریف مقادیر ثابت برای قاره‌ها.
- ``CSP`` و ``Solver`` برای مدیریت و حل مسئله CSP.
- ``generate_borders_by_continent`` برای تولید مرزهای قاره‌ها.
- ``draw`` برای نمایش نتایج.
- ``random`` برای تولید اعداد تصادفی (در صورت نیاز).

2. تعریف کلاس Continent

- این کلاس مقادیر ثابت برای قاره‌ها را تعریف می‌کند. این مقادیر به صورت Enum تعریف شده‌اند که استفاده از آن‌ها را در آرگومان‌های خط فرمان آسان می‌کند.

3. تابع main

- این خطوط یک parser را برای تجزیه و تحلیل آرگومان‌های خط فرمان ایجاد می‌کنند و اطلاعات مربوط به برنامه و توضیحات آن را تنظیم می‌کنند.

4. تعریف آرگومان‌های خط فرمان

- این خطوط آرگومان‌های مختلف خط فرمان را تعریف می‌کنند:
- `--map`: انتخاب قاره‌ای که نقشه آن رنگ‌آمیزی خواهد شد.
- `--lcv`: فعال کردن هیوریستیک LCV.
- `--mrv`: فعال کردن هیوریستیک MRV.
- `--arc-consistency`: فعال کردن الگوریتم AC-3 برای کاهش دامنه‌های متغیرها.
- `--Neighborhood-distance`: تنظیم فاصله همسایگی برای تعیین تشابه رنگ‌ها بین مناطق.

5. پردازش آرگومان‌های خط فرمان

- این خطوط آرگومان‌های خط فرمان را تجزیه و تحلیل می‌کنند و مقدار قاره انتخاب شده را به رشته تبدیل می‌کنند.

6. تولید مرزهای قاره انتخاب شده

- این خط مرزهای قاره انتخاب شده را با استفاده از تابع `generate_borders_by_continent` تولید می‌کند.

7. ایجاد یک نمونه از CSP و افزودن متغیرها و دامنه‌ها

- این خطوط یک نمونه از کلاس CSP ایجاد می‌کنند و متغیرها (مناطق) و دامنه‌های آن‌ها (رنگ‌ها) را به آن اضافه می‌کنند. دامنه‌های رنگ‌ها بر اساس مقدار `Neighborhood-distance` تعیین می‌شوند.

8. افزودن محدودیت‌ها (مرزهای بین مناطق)

- این بخش محدودیت‌ها را به مسئله اضافه می‌کند:
- اگر منطقه‌ای هیچ همسایه‌ای نداشته باشد، محدودیت خاصی برای آن اضافه می‌شود.
- برای هر منطقه و همسایه‌های آن، محدودیت‌های عدم تشابه رنگ‌ها اعمال می‌شود.
- این محدودیت‌ها بر اساس مقدار `Neighborhood-distance` تنظیم می‌شوند تا تأثیر همسایگان همسایگان نیز در نظر گرفته شود.

9. ایجاد و استفاده از Solver

- این خطوط یک نمونه از کلاس Solver ایجاد می‌کنند و تنظیمات مختلفی مانند استفاده از هیوریستیک‌ها و الگوریتم AC-3 را به آن اعمال می‌کنند.
- سپس الگوریتم برگشت‌پذیری برای حل مسئله اجرا می‌شود.

10. نمایش نتایج

- این بخش نتایج حل مسئله را نمایش می‌دهد:
- تعداد رنگ‌های استفاده شده را چاپ می‌کند.
- نقشه رنگ‌آمیزی شده را نمایش می‌دهد.

11. اجرای تابع main

- این خط اطمینان حاصل می‌کند که تابع `main` فقط زمانی اجرا می‌شود که این فایل به عنوان برنامه اصلی اجرا شود، نه وقتی که به عنوان ماژول وارد شود.