



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

گزارش چهارم

مسئله ارضای محدودیت در رنگ آمیزی

علیرضا داودی _ 4013011

دکتر مهدی قطعی _ دکتر بهنام یوسفی مهر

بهار هزار و چهارصد و سه

چکیده

این مقاله به بررسی الگوریتم هایی برای رنگ آمیزی می پردازد. برای رنگ آمیزی نقشه از الگوریتم جست و جوی محدودیت^۱ استفاده شده که بهبود هایی برای الگوریتم پسگرد^۲ در آن وجود دارد. برای رنگ آمیزی نقشه قاره ها باید این را در نظر بگیریم که کشور های همسایه نمی توانند هم رنگ باشند. همچنین یک از دیگر چالش های این کد درجه همسایگی^۳ است که در ادامه به آن می پردازیم.

¹ Csp (constraint satisfaction problem)

² Backtracking solver

³ Neighborhood distance

صفحه	فهرست مطالب
أ.....	چکیده
1	فصل اول مقدمه
3	فصل دوم csp
3	csp 1-1
4	2-1 پیاده سازی csp
9	فصل دوم توضیح mr,lc,ac3
6	mr 1-2
7	lc 2-2
7	ac3 2-3
10.....	فصل سوم تحلیل الگوریتم ها
12	backtrack solving 3-1
13	lc 2-3
14	mr 3-3
14	ac3 3-3
10.....	فصل چهارم تعداد رنگ
15.....	فصل پنجم نتیجه
17.....	منابع
Abstract	21

در این پروژه باید کشور های یک قاره رنگ شود. رنگ امیزی این کشور ها باید به طوری باشد که رنگ کشور های همسایه یکی نباشد. البته اینکه درجه همسایگی نیز چه مقدار باشد نیز مهم است. برای حل این مشکل از الگوریتم CSP استفاده میکنیم. این الگوریتم با کمک الگوریتم پسگرد و تابع های بهبود برای ان استفاده میکند و این امکان را دارد با بررسی و برگشت های کمتر و زمان بسیار کم نقشه یک قاره را رنگ کند. در ادامه این مقاله به بررسی این الگوریتم می پردازیم.

: Csp

این کلاس برای نمایش مسئله CSP استفاده می‌شود و شامل ویژگی‌های زیر است:

variables : دیکشنری که متغیرهای مسئله و دامنه‌های ممکن برای هر متغیر را نگهداری می‌کند.

constraints : لیستی که محدودیت‌های بین متغیرها را ذخیره می‌کند.

unassigned_var : لیستی از متغیرهایی که هنوز ارزشی به آن‌ها اختصاص داده نشده است.

var_constraints : دیکشنری که محدودیت‌های مرتبط با هر متغیر را ذخیره می‌کند.

assignments : دیکشنری که ارزش‌های اختصاص داده شده به متغیرها را نگهداری می‌کند.

assignments_number : تعداد اختصاص‌های انجام شده.

متدها:

add_constraint : یک محدودیت به CSP اضافه می‌کند.

add_variable : یک متغیر و دامنه آن را به دیکشنری متغیرها اضافه می‌کند.

assign : یک مقدار به یک متغیر اختصاص می‌دهد و مطابقت انتساب را بررسی می‌کند.

is_consistent : بررسی می‌کند که انتساب کنونی یک مقدار به یک متغیر با محدودیت‌های CSP سازگار است یا خیر.

is_complete : بررسی می‌کند که آیا همه متغیرها به یک مقدار اختصاص یافته‌اند یا خیر.

is_assigned : بررسی می‌کند که آیا یک متغیر خاص به یک مقدار اختصاص یافته است یا خیر.

unassign : انتساب را بازگردانی می‌کند.

کلاس Solver

این کلاس برای حل مسئله CSP استفاده می‌شود و شامل ویژگی‌ها و متدهای زیر است:

domain_heuristic : در صورت فعال بودن، از هیوریستیک مقدار دامنه کمترین محدودیت (LCV) استفاده می‌کند.

variable_heuristic : در صورت فعال بودن، از هیوریستیک متغیر با کمترین مقدار باقی‌مانده (MRV) استفاده می‌کند.

AC_3 : در صورت فعال بودن، از الگوریتم AC-3 برای کاهش دامنه متغیرها استفاده می‌کند.

الگوریتم‌های به کار رفته:

Backtracking

این الگوریتم از روش بازگشت به عقب (Backtracking) برای حل مسئله CSP استفاده می‌کند. این الگوریتم به صورت رکورسیو تمام ترکیب‌های ممکن از مقادیر را برای متغیرها امتحان می‌کند و در صورتی که مقداری با محدودیت‌ها سازگار نباشد، یک گام به عقب برمی‌گردد و مقدار دیگری را امتحان می‌کند.

MRV (Minimum Remaining Values)

این هیوریستیک متغیری را انتخاب می‌کند که کمترین تعداد مقادیر باقی‌مانده در دامنه خود دارد. ایده پشت این هیوریستیک این است که اولویت را به متغیرهایی بدهیم که احتمال شکست در آن‌ها بیشتر است تا زودتر به شکست یا موفقیت برسیم.

LCV (Least Constraining Value)

این هیوریستیک برای انتخاب مقداری برای یک متغیر استفاده می‌شود که کمترین محدودیت را بر روی متغیرهای دیگر اعمال می‌کند.

AC-3 (Arc Consistency Algorithm)

این الگوریتم برای کاهش دامنه متغیرها استفاده می‌شود تا مقداری که مطمئناً به نتیجه شکست منجر می‌شوند را حذف کنیم. این کار باعث می‌شود عملیات حل مسئله سریع‌تر انجام شود.

arc_reduce

تلاش برای کاهش دامنه یک متغیر با حذف مقادیری که با دامنه متغیر دیگری تضاد دارند. این متد در متد `apply_AC3` استفاده می شود.

ایمپورت‌ها

argparse : برای پردازش آرگومان‌های خط فرمان.

Enum : برای تعریف مقادیر ثابت و قابل خواندن از سردر **enum** برای قاره‌ها.

CSP : کلاس تعریف شده مساله **CSP** که پیش‌تر توضیح داده شد.

Solver : کلاس حل‌کننده مساله **CSP**.

generate_borders_by_continent : تابعی که بر اساس قاره مشخص شده مرزها را تولید

می‌کند.

draw : تابعی که نتایج را به صورت گرافیکی نمایش می‌دهد.

random : ماژولی که برای تولید اعداد تصادفی استفاده می‌شود.

کلاس **Continent**

یک **enum** برای نمایش قاره‌های مختلف تعریف شده است. این به کاربر اجازه می‌دهد تا با استفاده از نام‌های قابل فهم قاره‌ها را انتخاب کند.

تابع **main**

این تابع اصلی برنامه است که فرآیند حل مساله **CSP** را آغاز می‌کند.

تجزیه و تحلیل آرگومان‌های خط فرمان

با استفاده از ماژول **argparse** ، برنامه می‌تواند تنظیمات ورودی را از کاربر دریافت کند:

--map (-m) : قاره‌ای که کاربر می‌تواند برای رنگ‌آمیزی انتخاب کند.

--lcv (-lcv) : اگر فعال باشد، از **heuristic** ارزش کمترین محدودیت (**LCV**) استفاده می‌کند.

--mrv (-mrv) : اگر فعال باشد، از **heuristic** کمترین مقادیر باقیمانده (**MRV**) استفاده می‌کند.

--arc-consistency (-ac3) : اگر فعال باشد، از الگوریتم **AC-3** برای حذف دامنه‌های

متغیرها استفاده می‌کند.

--Neighborhood-distance (-ND) : فاصله همسایگی که تعیین می‌کند رنگ مناطق

همسایه تا چه اندازه باید متفاوت باشد.

تولید مرزها برای قاره مشخص شده

با استفاده از تابع `generate_borders_by_continent` مرزهای قاره انتخاب شده تولید می‌شوند.

ایجاد نمونه‌ای از CSP:

یک نمونه از کلاس CSP ایجاد می‌شود و برای هر منطقه (متغیر)، دامنه‌های ممکن (رنگ‌ها) اضافه می‌شوند.

اضافه کردن محدودیت‌ها:

برای هر منطقه و همسایه‌هایش، محدودیت‌هایی اضافه می‌شوند که می‌گویند دو منطقه همسایه نمی‌توانند رنگ یکسان داشته باشند.

مقداردهی اولیه حل‌کننده و حل CSP:

یک نمونه از کلاس Solver با تنظیمات مناسب ایجاد شده و سپس متد `backtrack_solver` برای حل CSP فراخوانی می‌شود.

نمایش نتایج:

با استفاده از تابع `draw`، نتایج به صورت گرافیکی نمایش داده می‌شوند.

اجرای برنامه:

دستور: `if __name__ == '__main__':` اطمینان حاصل می‌کند که تابع `main` فقط زمانی اجرا می‌شود که اسکریپت به عنوان اسکریپت اصلی اجرا شود و نه هنگامی که به عنوان ماژول وارد می‌شود.

ایمپورت‌ها

ابتدا ماژول‌های ضروری برای اجرای برنامه را ایمپورت می‌کنیم:

`argparse`: برای پردازش آرگومان‌های خط فرمان.

`Enum`: برای تعریف مقادیر ثابت و قابل خواندن از سردر `enum` برای قاره‌ها.

`CSP`: کلاس تعریف شده مساله CSP که پیش‌تر توضیح داده شد.

`Solver`: کلاس حل‌کننده مساله CSP.

`: generate_borders_by_continent`

تابعی که بر اساس قاره مشخص شده مرزها را تولید می‌کند.

`draw`: تابعی که نتایج را به صورت گرافیکی نمایش می‌دهد.

`random`: ماژولی که برای تولید اعداد تصادفی استفاده می‌شود.

کلاس `Continent`:

یک `enum` برای نمایش قاره‌های مختلف تعریف شده است. این به کاربر اجازه می‌دهد تا با استفاده از نام‌های قابل فهم قاره‌ها را انتخاب کند.

تابع `main`:

این تابع اصلی برنامه است که فرآیند حل مساله `CSP` را آغاز می‌کند.

تجزیه و تحلیل آرگومان‌های خط فرمان

با استفاده از ماژول `argparse`، برنامه می‌تواند تنظیمات ورودی را از کاربر دریافت کند:

`(-m) --map``: قاره‌ای که کاربر می‌تواند برای رنگ‌آمیزی انتخاب کند.

`(-lcv) --lcv``: اگر فعال باشد، از `heuristic` ارزش کمترین محدودیت (`LCV`) استفاده می‌کند.

`(-mr) --mr``: اگر فعال باشد، از `heuristic` کمترین مقادیر باقیمانده (`MRV`) استفاده

می‌کند.

`(-ac3) --arc-consistency``: اگر فعال باشد، از الگوریتم `AC-3` برای حذف دامنه‌های

متغیرها استفاده می‌کند.

`(-ND) --Neighborhood-distance``: فاصله همسایگی که تعیین می‌کند رنگ مناطق

همسایه تا چه اندازه باید متفاوت باشد.

تولید مرزها برای قاره مشخص شده:

با استفاده از تابع `generate_borders_by_continent``، مرزهای قاره انتخاب شده تولید می‌شوند.

ایجاد نمونه‌ای از CSP:

یک نمونه از کلاس `CSP` ایجاد می‌شود و برای هر منطقه (متغیر)، دامنه‌های ممکن (رنگ‌ها) اضافه می‌شوند.

اضافه کردن محدودیت‌ها:

برای هر منطقه و همسایه‌هایش، محدودیت‌هایی اضافه می‌شوند که می‌گویند دو منطقه همسایه نمی‌توانند رنگ یکسان داشته باشند.

مقداردهی اولیه حل‌کننده و حل CSP:

یک نمونه از کلاس `Solver` با تنظیمات مناسب ایجاد شده و سپس متد `backtrack_solver` برای حل CSP فراخوانی می‌شود.

نمایش نتایج:

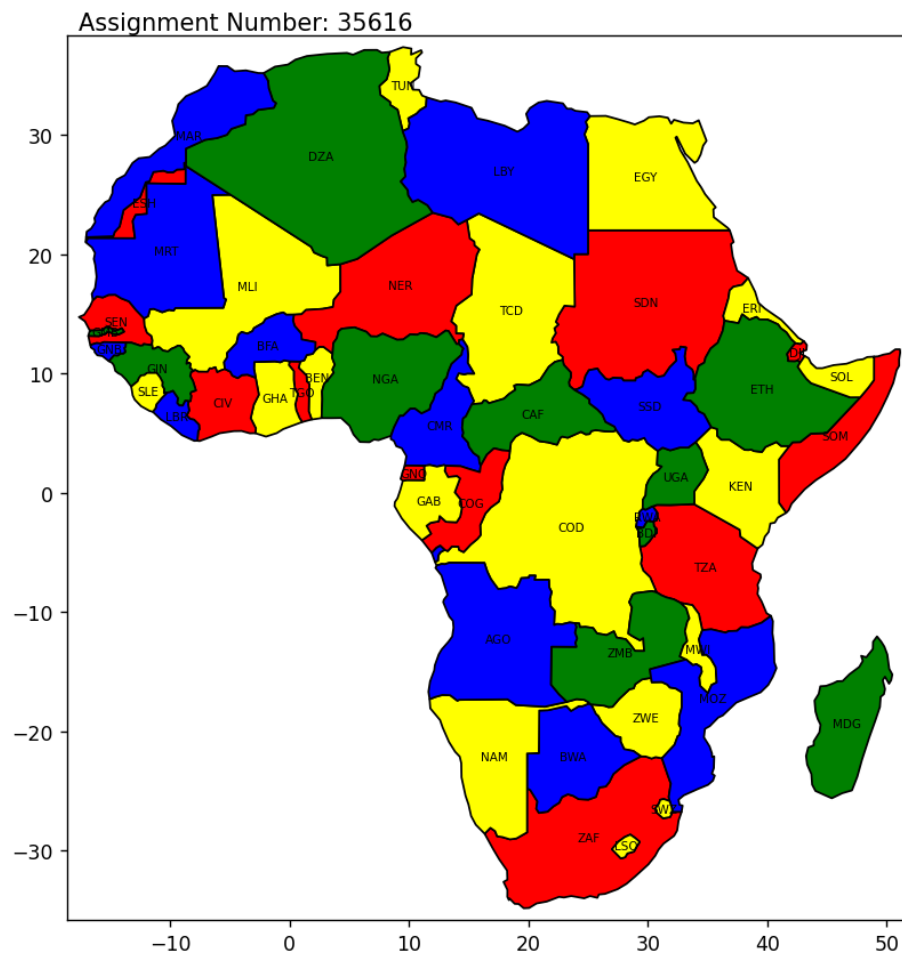
با استفاده از تابع `draw`، نتایج به صورت گرافیکی نمایش داده می‌شوند.

اجرای برنامه:

دستور `if __name__ == '__main__':` اطمینان حاصل می‌کند که تابع `main` فقط زمانی اجرا می‌شود که اسکریپت به عنوان اسکریپت اصلی اجرا شود و نه هنگامی که به عنوان ماژول وارد می‌شود.

: Backtrack solver

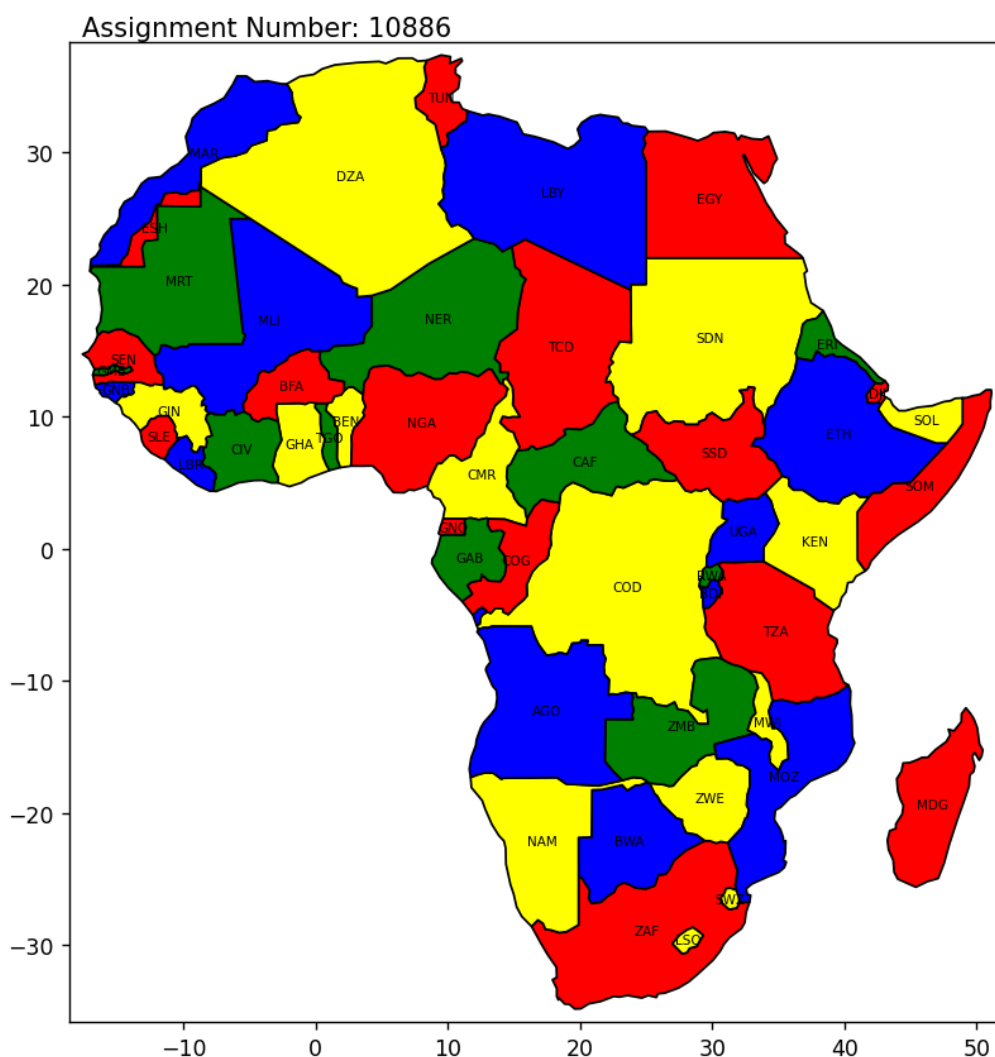
ابتدا از الگوریتم پسگرد استفاده میکنیم و از دیگر الگوریتم ها استفاده نمی کنیم.



طبق نتیجه بدست آمده ، این الگوریتم تعداد زیادی assignment دارد که به این دلیل است که الگوریتم انتخاب های اولیه خوبی برای کشور ها ندارد و همچنین رنگ هایی که انتخاب میکند باعث می شود که محدودیت های بیشتری ایجاد شود. برای اینکه این مشکلات رفع شود از هیوریستیک هایی مانند lcv و mrv استفاده میکنیم.

: Lcv

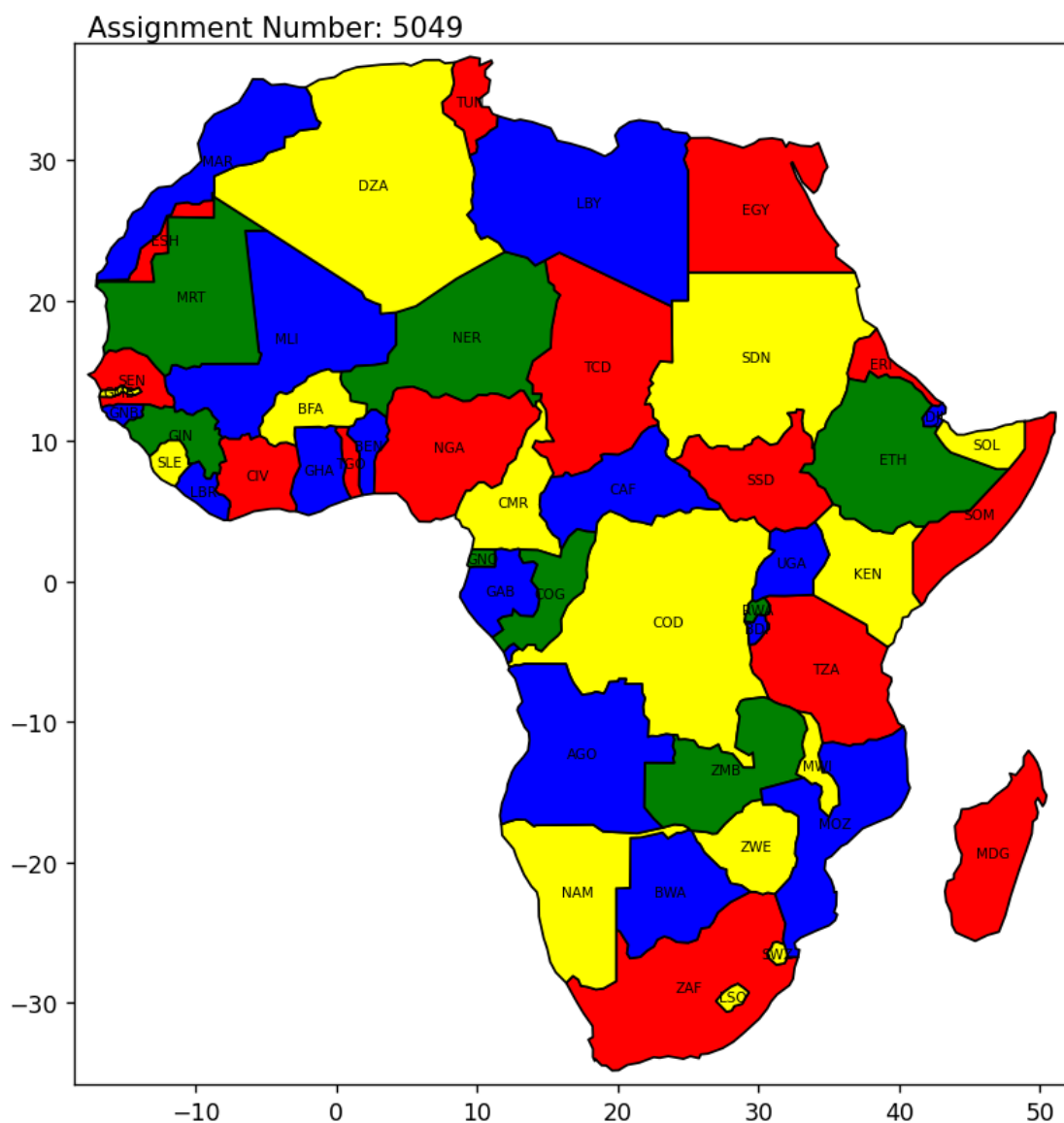
حال از هیوریتیک Icy برای دسته بندی رنگ ها برای یک کشور به طوری که کمترین محدودیت را ایجاد کند استفاده میکنیم.



طبق نتیجه بدست آمده با این کار تعداد assignment ها کاهش پیدا کرده زیرا الگوریتم رنگ های بهتری را برای هر کشور پیدا میکند اما هنوز الگوریتم به اصلاح بیشتری نیاز دارد. برای این کار باید از الگوریتم AC3 استفاده کنیم که دامنه متغیر ها را کم میکند.

: AC3

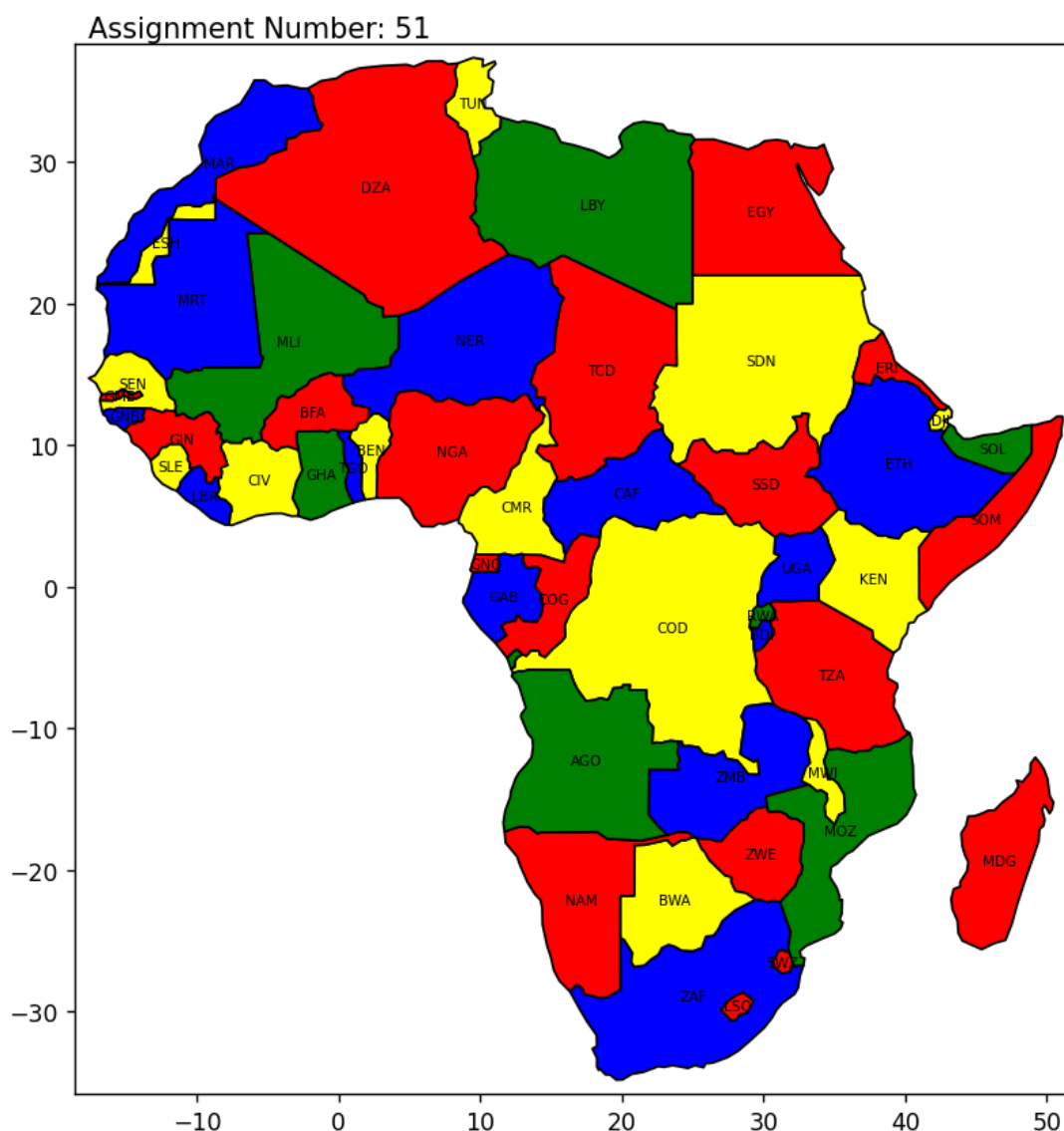
این الگوریتم برای کاهش دامنه متغیرها استفاده می‌شود تا مقادیری که مطمئناً به نتیجه شکست منجر می‌شوند را حذف کنیم. این کار باعث می‌شود عملیات حل مسئله سریع‌تر انجام شود.



طبق نتیجه بدست آمده تعداد assignment ها کمتر شده ولی همچنان میشه این عدد را کم کرد. برای این کار باید از الگوریتم mrv برای دسته بندی متغیر ها استفاده کنیم.

:MRV

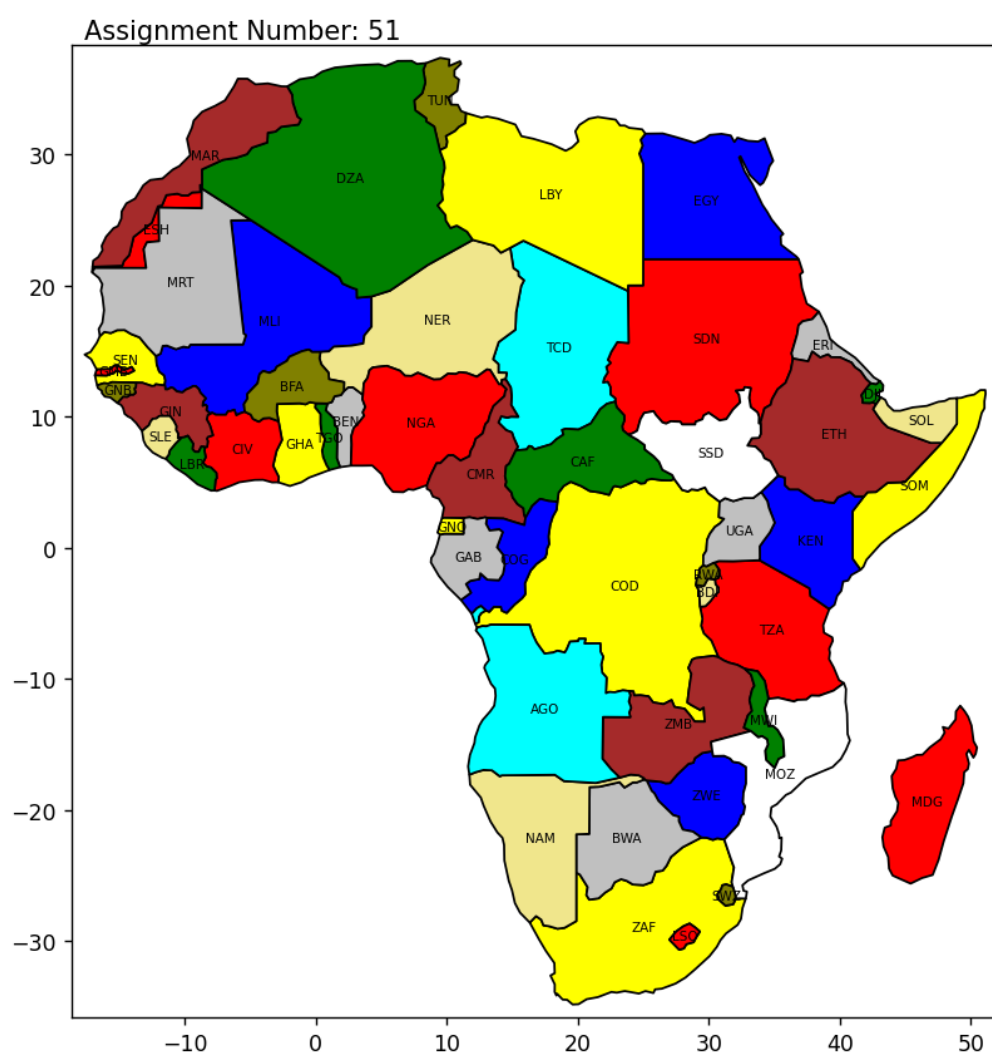
این هیورستیک متغیری را انتخاب می‌کند که کمترین تعداد مقادیر باقی‌مانده در دامنه خود دارد. ایده پشت این هیورستیک این است که اولویت را به متغیرهایی بدهیم که احتمال شکست در آن‌ها بیشتر است تا زودتر به شکست یا موفقیت برسیم.



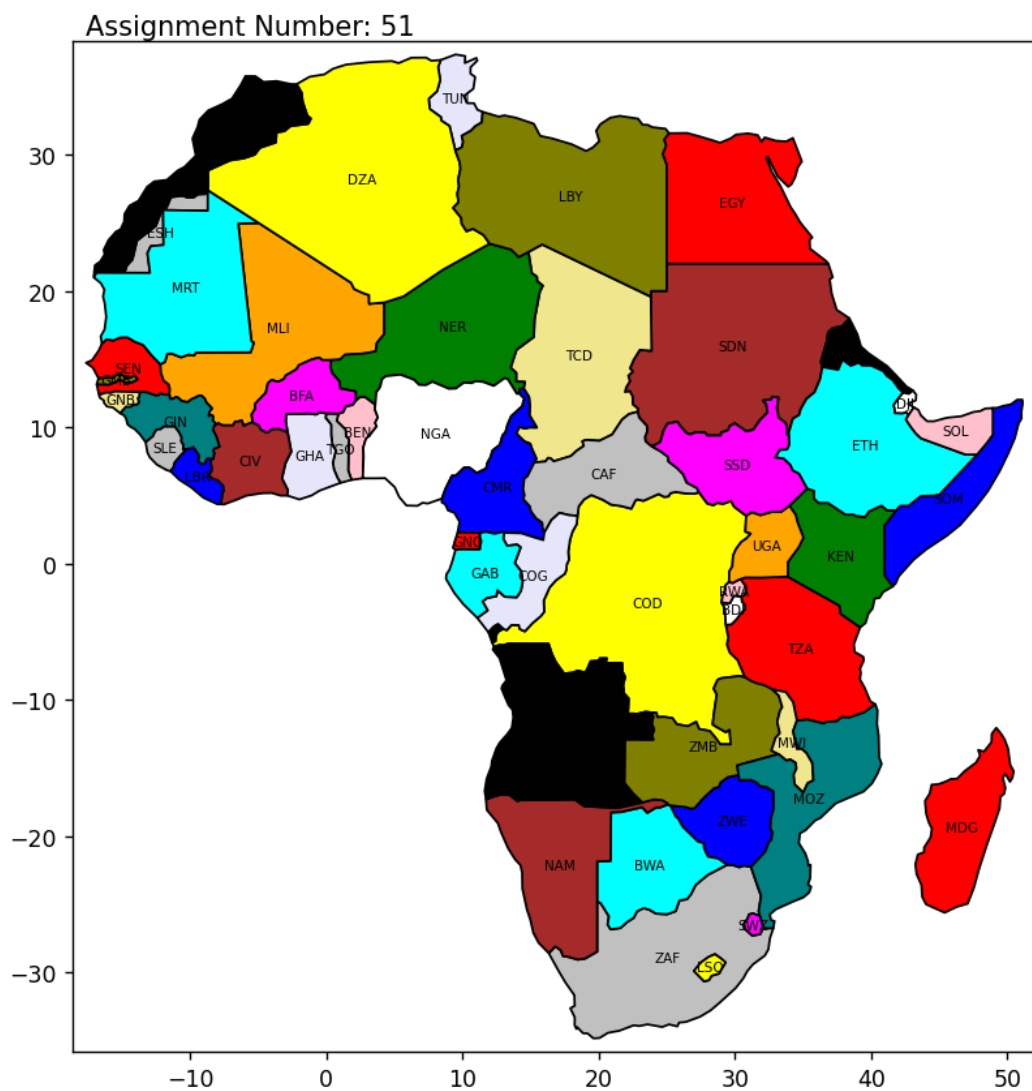
طبق نتیجه بدست آمده تعداد assignment ها کاهش پیدا کرده به طوری که تعداد ان با تعداد کشور های قاره افریقا برابر شده که این یعنی به بهینه ترین حالت خود رسیده است.

یکی دیگر از اهداف این پروژه فاصله همسایگی است. ورودی به این صورت است که یک عدد به عنوان ND^4 داده میشود. این مشخص میکند چه کشور هایی نباید هم رنگ باشند. برای مثال اگر درجه همسایگی دو باشد، در این صورت ایران با افغانستان همسایه است و افغانستان نیز با چین همسایه است در نتیجه چین همسایه درجه دو ایران محسوب میشود و نباید هم رنگ باشند.

برای مشخص کردن این که چه کشور هایی باهم نباید هم رنگ باشند در قسمت main هنگام ادد کردن constraint ها یک حلقه به تعداد فاصله همسایگی درست میکند و هربار همسایه های همسایه ها را نیز برای region ادد میکند. این باعث می شود که درجه همسایگی حفظ شود.



تصویر بالا مثالی از فاصله همسایگی دو است که میتوان این را از تصویر دریافت که همزنگ نیستند.



تصویر بالا مثالی از فاصله همسایگی با مقدار سه باشد.

تعداد رنگ:

الگوریتم پسگرد با هیوریستیک mr_v از کمترین رنگ برای رنگ آمیزی استفاده می کند. پس میتوان از mr_v استفاده کرد. میتوان تعداد رنگ های به کار رفته در mr_v را شمرد. به این صورت میشود :

برای رنگ آمیزی افریقا با درجه همسایگی یک به چهار رنگ نیاز داریم.

برای رنگ آمیزی افریقا با درجه همسایگی دو به ده رنگ نیاز داریم.

برای رنگ آمیزی افریقا با درجه همسایگی سه به شانزده رنگ نیاز داریم.

برای رنگ آمیزی یک نقشه میتوان از الگوریتم پسگرد استفاده کرد که این الگوریتم به تنهایی زمان زیادی می برد. برای بهبود آن میتوان از هیوریستیک lc_v برای کم کردن دامنه متغیر ها ، هیوریستیک mr_v برای مرتب کردن متغیر ها برای بررسی ، ac_3 برای کم حذف بعضی رنگ ها.

این ها باعث می شود که الگوریتم سرعت بالایی داشته باشد. از طرفی اینکه چه تعداد رنگی برای رنگ آمیزی نیاز است نیز مهم است. اگر فاصله همسایگی یک باشد با تعداد چهار رنگ حداکثر رنگ خواهد شد.

اگر فاصله همسایگی بیشتر شود تعداد رنگ ها نیز بیشتر میشود.

https://en.wikipedia.org/wiki/Four_color_theorem

<https://medium.com/@co.2020.asbaig/solving-map-coloring-problems-using-constraint-satisfaction-problems-csps-5359ac57d4c2>

<https://www.geeksforgeeks.org/m-coloring-problem>

<https://codereview.stackexchange.com/questions/289067/map-coloring-algorithm-functional-implementation-in-python>

To color a map, you can use the backtracking algorithm, which alone takes a lot of time. To improve it, you can use the lcv heuristic to reduce the range of variables, the mrv heuristic to sort the variables for checking, ac3 to remove some colors

These make the algorithm have a high speed. On the other hand, how many colors are needed for painting is also important. If the neighbor distance is one, it will be colored with the maximum number of four colors

.If the neighbor distance increases, the number of colors also increases



Amirkabir University of Technology

(Tehran Polytechnic)

Faculty MCS

Fourth Project

Csp in Map-coloring

4013011 _ Alieza Davoudi

Dr.Yousefi Mehr _ Dr.Ghatee

Spring, 2024