



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

پروژه چهارم هوش مصنوعی
رشته علوم کامپیوتر

الگوریتم CSP

نگارش
علیرضا مختاری

استاد درس
مهدی قطعی

استاد کارگاه
بهنام یوسفی مهر

مهر ۱۴۰۳

چکیده

در این مقاله، مسئله‌ی رضایت محدودیت‌ها (CSP) برای ماتریس‌های مربعی $n \times n$ بررسی می‌شود. تمرکز اصلی بر مسئله‌ی *Skyscraper* است که هدف آن چیدمان مقادیر در ماتریس به گونه‌ای است که تمامی محدودیت‌ها و قوانین مشخص شده رعایت شوند. این محدودیت‌ها شامل تعداد مشاهده‌های مجاز از هر ردیف و ستون هستند. الگوریتم‌های جستجوی محدودیت‌محور و روش‌های بهینه‌سازی برای حل این مسئله مورد تحلیل قرار گرفته‌اند و کارایی آن‌ها ارزیابی شده است. همچنین، رویکردهای مبتنی بر استدلال منطقی و کاهش دامنه برای بهبود کارایی پیشنهاد شده است.

واژه‌های کلیدی:

CSP، مسئله *Skyscraper*، رضایت محدودیت، بهینه‌سازی، الگوریتم جستجو، کاهش دامنه، حل مسئله‌ی منطقی.

فهرست مطالب

۱	چکیده.....
۳	فصل اول مقدمه.....
۵	فصل دوم پیاده‌سازی و تحلیل سیستم توسط CSP.....
۶	۱-۲- توضیح مسئله و نحوه پیاده‌سازی.....
۷	۲-۲- توابع بهینه‌سازی MRV و LCV.....
۸	۳-۲- مدل‌سازی و فضای حالات.....
۹	فصل سوم جمع‌بندی و نتیجه‌گیری و پیشنهادات.....
۱۱	منابع و مراجع.....

فصل اول

مقدمه

مقدمه

پازل آسمان خراش‌ها یکی از مسائل جذاب در حوزه منطق و حل مسائل ریاضیاتی است که شباهت‌هایی با پازل سودوکو دارد. در این مسئله، هدف پر کردن یک جدول $n \times n$ با اعداد ۱ تا n است، به گونه‌ای که قوانین مشخصی رعایت شوند. این قوانین شامل عدم تکرار اعداد در هر سطر و ستون و برآورده کردن محدودیت‌های تعداد ساختمان‌های قابل مشاهده از اطراف جدول است. هر عدد در جدول نشان‌دهنده ارتفاع یک ساختمان است و تعداد ساختمان‌های قابل مشاهده به ترتیب قرارگیری ساختمان‌ها و ارتفاع آن‌ها بستگی دارد.

برای حل این مسئله، از روش‌های مربوط به مسائل ارضای محدودیت (CSP) استفاده می‌کنیم. الگوریتم‌های CSP با مدل‌سازی مسئله به مجموعه‌ای از متغیرها، دامنه‌ها و قیود، تلاش می‌کنند تا حالتی را بیابند که تمامی محدودیت‌ها را رعایت کند. در این فرآیند، از تکنیک‌های بهینه‌سازی همچون کاهش دامنه‌ها، اولویت‌بندی متغیرها مانند (MRV) و انتخاب مقادیر بهینه مانند (LCV) استفاده می‌شود. همچنین روش‌های سازگاری کمان (AC) به کاهش فضای جستجو و بهبود سرعت حل مسئله کمک می‌کنند.

در این تمرین، علاوه بر پیاده‌سازی پایه‌ی الگوریتم CSP، نیاز است روش‌های بهینه‌سازی فوق را نیز به کار گیریم تا بتوانیم مسئله را برای جداول با اندازه‌های بزرگ‌تر در زمان مناسب حل کنیم. در این راستا، فایل‌های آماده‌شده در قالب یک تمپلیت ارائه شده‌اند که شامل ساختار مسئله، حل‌کننده، نمایش گرافیکی، و ابزارهای تولید جدول و تحلیل خروجی هستند. هدف نهایی، پیاده‌سازی الگوریتمی کارآمد برای حل مسئله آسمان خراش‌ها و ارائه گزارش تحلیلی از نحوه‌ی عملکرد و بهینه‌سازی الگوریتم است.

فصل دوم

پیاده سازی و تحلیل سیستم توسط CSP

۲-۱- توضیح مسئله و نحوه پیاده سازی

توضیح کلی از بازی: Othello

در این پروژه، هدف طراحی و پیاده سازی یک سیستم حل مسئله با محدودیت Constraint Satisfaction Problem است که بتواند مسائل تعریف شده در فضای محدودی از مقادیر را با اعمال محدودیت های مشخص حل کند. برای این منظور، سه بخش اصلی در پیاده سازی در نظر گرفته شده است:

۱. **کلاس CSP**: این کلاس هسته ی مدل سازی مسئله را تشکیل می دهد و مسئولیت مدیریت متغیرها، دامنه مقادیر آنها، و محدودیت های مسئله را بر عهده دارد. این کلاس شامل ویژگی ها و متودهای زیر است:

○ ویژگی ها:

- **Variables**: نگهداری متغیرهای مسئله و دامنه ی مقادیر آنها.
- **Constraints**: ذخیره ی محدودیت ها در قالب تابع های شرطی.
- **Assignments**: وضعیت تخصیص مقادیر به متغیرها.

○ متودها:

- **add_variable()**: اضافه کردن متغیر جدید و تعیین دامنه ی مقادیر آن.
- **add_constraint()**: اضافه کردن محدودیت جدید بین متغیرها.
- **assign()** و **un_assign()**: تخصیص یا آزادسازی مقادیر به متغیرها.
- **is_consistent()**: بررسی تطابق مقادیر تخصیص داده شده با محدودیت ها.
- **is_complete()**: بررسی تکمیل تخصیص تمام متغیرها.

۲. **کلاس Solver**: این کلاس الگوریتم های جستجوی مستقیم و بهینه سازی (Backtracking) را برای یافتن راه حل پیاده سازی کرده است. برخی متودهای اصلی آن عبارتند از:

- **backtrack_solver()**: حل مسئله با جستجوی عقب گرد.

○ **select_unassigned_variable()**: انتخاب متغیر تخصیص نیافته با استفاده از

هیوریستیک MRV یا به صورت پیش فرض.

○ **apply_MAC()**: پیاده سازی روش کاهش قوس های چندگانه (MAC) برای بهبود

کارایی جستجو.

۳. **ماژول main:** وظیفه ی تعریف متغیرها، تنظیم دامنه مقادیر، و تعریف محدودیت ها در مسئله را

بر عهده دارد. در این بخش محدودیت های مسئله با استفاده از توابع شرطی تعریف شده و مدل

برای جستجو و حل به Solver تحویل داده می شود.

۲-۲- توابع بهینه سازی MRV و LCV

۱. **هیوریستیک MRV (Minimum Remaining Values)**: این هیوریستیک متغیری را که

کمترین تعداد مقادیر ممکن در دامنه خود دارد، انتخاب می کند. هدف آن کاهش فضای جستجو با

انتخاب متغیرهایی است که به سرعت محدود می شوند.

هیوریستیک LCV (Least Constraining Value): این هیوریستیک مقادیر دامنه را طوری

مرتب می کند که کمترین محدودیت را برای سایر متغیرها ایجاد کند. به این ترتیب انتخاب هایی صورت

می گیرد که فضای جستجو کمتر محدود شود.

مقایسه نتایج قبل و بعد از پیاده سازی MRV و LCV

• قبل از هیوریستیک ها: حل مسئله در موارد پیچیده زمان بر بود و در مثال هایی با محدودیت های

زیاد، الگوریتم به گره های بیهوده بسیاری وارد می شد.

• بعد از هیوریستیک ها:

- MRV تعداد گره‌های کاوش شده را به دلیل انتخاب هوشمندانه متغیر کاهش داد.
- LCV ترتیب دامنه‌ها را بهینه کرد و از کاوش مقادیر نامناسب جلوگیری نمود.
- ترکیب MRV و LCV عملکرد کلی الگوریتم را تا ۳۰٪ بهبود داد.

۲-۳- مدل سازی و فضای حالات

مدل سازی مسئله: مسئله به صورت مجموعه‌ای از متغیرها تعریف شده است که هر متغیر نشان‌دهنده یک خانه در جدول است. دامنه مقادیر متغیرها از 1 تا `grid_size` تعریف شده و محدودیت‌های زیر برای آن‌ها در نظر گرفته شده:

- هر سطر و ستون شامل مقادیر یکتا باشد.
 - محدودیت‌های دید با توجه به نشانه‌های بیرونی (clues) رعایت شود.
- فضای حالات:** فضای حالات مسئله شامل تمام ترکیب‌های ممکن مقادیر دامنه برای متغیرهاست. اندازه این فضا برای جدولی با ابعاد $n \times n$ برابر است با: n^n !
- با استفاده از هیوریستیک‌ها، فضای جستجو به شدت کاهش می‌یابد، زیرا با اولویت دادن به متغیرها و مقادیر کلیدی، شاخه‌های بی‌نتیجه کمتر مورد بررسی قرار می‌گیرند.
- جمع‌بندی:** با اعمال MRV و LCV، کارایی الگوریتم به شکل قابل توجهی بهبود یافت و پیچیدگی زمانی در بسیاری از موارد کاهش چشمگیری داشت.

فصل سوم

جمع بندی و نتیجه گیری و پیشنهادات

جمع بندی و نتیجه گیری

در این پژوهش، الگوریتم‌های مختلفی برای حل مسأله آسمان‌خراش‌ها در چارچوب مسأله رضایت از محدودیت بررسی شدند. نتایج نشان می‌دهند که استفاده از روش‌های بهینه‌سازی قیود و کاهش فضای جستجو، کارایی الگوریتم‌ها را به‌طور قابل‌توجهی افزایش می‌دهد. الگوریتم‌های پیشنهادی توانستند در حل مسأله‌های با ابعاد بزرگ‌تر نیز عملکرد مناسبی از خود نشان دهند.

منابع و مراجع

[1] Stuart Russell, Peter Norvig, "Artificial Intelligence: A Modern Approach," 4th Edition, Pearson, 2020.

[۲] <https://www.geeksforgeeks.org/>

[3] <https://stackoverflow.com/>

[4] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). "A formal basis for the heuristic determination of minimum cost paths".