گزارش ۵: پیاده سازی حل یک مسئله ی تصمیم گیری با استفاده از ارضای محدودیت

نگارش

نیکا شهابی۹۷۱۳۰۲۳

استاد

دکتر مهدی قطعی

فروردین ۱۴۰۰

لینک پروژه در گیتاب:

https://github.com/nikashahabi/sudoku-CSP-solver

توضیح بازی سودوکو:

جدولی ۹*۹ که خانه های آن یا خالی هستند یا عددی از ۱ تا ۹ دارند به ما داده میشود. قصد این است که خانه های خالی را به گونه ای با اعداد ۱ تا ۹ پر کنیم که در هر سطر، ستون و مربع ۳*۳ عدد تکراری وجود نداشته باشد.

SUDOKU GAME WITH ANSWER

	9				6	1		
7	1			9			2	8
		8					4	9
	6		2	5				1
5				6	9		7	
1	2					8		
9	8			4			5	2
		5	6				1	

4	9	2	7	8	6	1	3	5
7	1	3	5	9	4	6	2	8
6	5	8	3	1	2	7	4	9
8	6	7	2	5	3	4	9	1
2	3	9	4	7	1	5	8	6
5	4	1	8	6	9	2	7	3
1	2	4	9	3	5	8	6	7
9	8	6	1	4	7	3	5	2
3	7	5	6	2	8	9	1	4

توضیح صورت مسئله:

با استفاده از الگوریتم های ارضای محدودیت، توسط یک هوش مصنوعی سودوکو بازی کنید و برای صفحه ی اولیه در صورت امکان یک جواب پیدا کنید. (در صورت وجود داشتن چند جواب، پیدا کردن یکی کافیست)

مدلسازی مسئله به شکل یک مسئله ی ارضای محدودیت یا CSP:

با کمک کتاب راسل مدلسازی را انجام میدهیم:

X: a set of variable: تمام خانه های خالی جدول اولیه ی داده شده

D: a set of domains = {1, 2, 3, …, 9} for each variable

C: a set of constraints that specify allowable combinations of values:
هر استیت به وسیله ی مقداردهی به یک یا همه ی متغیرها مشخص میشود

Constraints in sudoku : alldiiff for each row, column and 3*3 square

چرا این الگوریتم و برتری آن در این مسئله نسبت به الگوریتم های سرچ فصل۳:

ما به دنبال جوابی هستیم که قیدهای گفته شده را ارضا کند. دنبال تمامی جواب ها نیستیم و به علاوه جواب بهینه بین این جواب ها وجود ندارد. (و البته معنی ندارد) پس الگوریتم های ارضای محدودیت با توجه به مدلسازی انجام شده قابل اجرا روی مسئله ی ما هستند. به علاوه چون به دنبال تمامی جواب ها نیستیم روش هایی مانند

MRV , Degree heuristicسرعت پیدا کردن یک جواب را بالا میبرند.

از طرفی در الگوریتم های عادی برایsearch ما فقط نگاه میکنیم که آیا یک استیت Goal هست یا

خیر. ولی چون در الگوریتم های ارضای محدودیت ما همه ی شاخه ها را تا انتها ادامه نمیدهیم و یک

partial assignment

و ادامه ی شاخه ی آن رابعضی مواقع به طور کامل حذف میکنیم و ادامه نمیدهیم زمان حل مسئله به شدت پایین می آید. به علاوه چون ترتیب مقداردهی به متغیرها اهمیتی ندارد ما در هر مرحله فقط یک متغیر را مقداردهی میکنیم که این خود موجب کوچک شدن درخت جست و جو و بالا رفتن زمان نسبت به الگوریتم های سرچ عادی میشود.

توضیح الگوریتم:

در ابتدا تحت عنوان setDomains

```
setDomains(sudoku)
```

، دامنه ها در صورت لزوم محدود میشوند. در حقیقت خانه های خالی با توجه به سطر، ستون و مربعی که داخل آن هستند از ابتدا نمیتوانند مقادیری را داشته باشند. این مقادیر از دامنه ی آن ها حذف میشود. (اگر خانه هایی که از ابتدا عدد داشته اند را هم متغیر در نظر بگیریم این preprocess همان چک کردن arc consistency است.)

سپس با استفاده از backtracking به حل مسئله میپردازیم. الگوریتم backtracking در بین خود از

```
selectUnassignedVar(sudoku, varChoosing, values, neighbors)
selectValue(var, values, valueChoosing, neighbors)
```

استفاده میکند.
valueChoosing و varChoosing در واقع استرینگ هایی هستند که به وسیله ی آن ها مشخص میکنیم که انتخاب متغیر و مقدار آن چگونه انجام شود.
برای انتخاب مقدار متغیر دو حالت داریم:

```
value choosing from 1 to 9
least constraining value
```

در اولی اعداد ۱ تا ۹ به ترتیب امتحان میشوند و در دومی اعدادی که در دامنه های همسایه ی متغیر مورد نظر بیشتر حضور دارند دیرتر امتحان و انتخاب میشوند.
برای انتخاب متغیر بعدی برای مقداردهی هم سه حالت داریم:

```
MRV
first unassigned var
degree heuristic
```

در اولی اولویت با متغیری است که مقدارهای legal موجود کمتری دارد که میتواند از بین آنها مقدار بپذیرد.
در دومی اولین خانه ی جدول که مقداردهی نشده انتخاب میشود.
و در سومی متغیری انتخاب میشود که در کمترین تعداد constraint برای متغیرهای مقداردهی نشده ی دیگر حضور دارد.

(در آخر به زمان اجرای ترکیب این ۶ حالت میپردازیم.)

الگوریتم علاوه بر بهره بردن از برخی هیوریستیک ها از forwardChecking نیز همیشه کمک میگیرد. به این صورت که بعد از انتخاب مقدار یک متغیر، متغیرهای در سطر، ستون و مربع آن که مقداردهی نشده اند مقدار انتخاب شده از دامنه ی آنها حذف میشود. البته اگر backtrack اتفاق بیوفتد این مقدار به دامنه ی متغیرهای گفته شده بازمیگردد.

Forwardchecking

با استفاده از سه تابع زیر انجام میشود.

```
getForwardCheck(var, value, neighbors, values)

forwardCheck(inferences, values, value)

reverseForwardCheck(inferences, values, value)

example: values[0][0] = [1, 2, 3]. 1, 2, 3 are legal values
for (0,0)

example: neighbors[0][0] = [(0,1)]. (0,1) is a unassigned
variable which has a binary constraint with (0,0)
```

توضیح توابع مسئله:

به صورت کامنت در بین کد

تحلیل زمانی و مکانی الگوریتم:

زمانی:

Arc consistency: $9 * 9 * O(9) = O(9^3)$

اگر با الگوریتم ac-3 هم انجام میشد زمانش همین بود

Backtracking in general: $O(b^d)$ where b is the branching factor (size of domain and d is the number of variables)

—-> O(9^81)

But with different heuristics and forward checking we see drastic change in the running time

مکانی: برای نگه داشتن values , neighbors
$O(9^3)$
صفحه ی اصلی یک بار نگه داری میشود و در هر call پاس داده میشود.

مقایسه ی زمان اجرای یک سودوکوی رندوم با هر یک از ۶ حالت زیر:

```
initsudoku = [[7,8,0,4,0,0,1,2,0],
    [6,0,0,0,7,5,0,0,9],
    [0,0,0,6,0,1,0,7,8],
    [0,0,7,0,4,0,2,6,0],
    [0,0,1,0,5,0,9,3,0],
    [9,0,4,0,6,0,0,0,5],
    [0,7,0,3,0,0,0,1,2],
    [1,2,0,0,0,7,4,0,0],
    [0,4,9,2,0,6,0,0,7]]
```

First unassigned var	MRV	Degree heuristic	
3.3005690574646	1.30630588531494	4.44880485534668	**Least constraining variable**
5.52742981910706	1.54769206047058	3.85449600219727	**Value choosing from 1 to 9**

همان طور که مشاهده میشود استفاده از least constraining variable بهتر از استفاده ی از اعداد ۱ تا ۹ به ترتیب است. همینطور مشاهده میشود که MRV بهتر از Degree heuristic عمل کرده.

ترکیب MRV و least constraining variable هم بهترین.

نمونه ی خروجی و ورودی

```
initsudoku = [[7,8,0,4,0,0,1,2,0],
    [6,0,0,0,7,5,0,0,9],
    [0,0,0,6,0,1,0,7,8],
    [0,0,7,0,4,0,2,6,0],
    [0,0,1,0,5,0,9,3,0],
    [9,0,4,0,6,0,0,0,5],
    [0,7,0,3,0,0,0,1,2],
    [1,2,0,0,0,7,4,0,0],
    [0,4,9,2,0,6,0,0,7]]
sudoku = copy.deepcopy(initsudoku)
time3 = time.time()
printSudoku(backtrackingSearch(sudoku, "MRV", "value choosing
from 1 to 9"))
time4 = time.time()
print("running time of MRV + value choosing from 1 to 9 = ",
time4 - time3)
```

Users/nayvaka/Documents/University/Term6/AI/AIProjects/Sudoku/venv//
bin/python /Users/nayvaka/Documents/University/Term6/AI/AIProjects/
Sudoku/main.py
performing backtracking + MRV + value choosing from 1 to 9
chosen variable = (1, 3)chosen value = 1backtracked. choosing a new value
for (1, 3) if available
chosen value = 2backtracked. choosing a new value for (1, 3) if available
chosen value = 3backtracked. choosing a new value for (1, 3) if available
chosen value = 4backtracked. choosing a new value for (1, 3) if available
chosen value = 5backtracked. choosing a new value for (1, 3) if available
chosen value = 6backtracked. choosing a new value for (1, 3) if available
chosen value = 7backtracked. choosing a new value for (1, 3) if available
chosen value = 8chosen value is consistent

chosen variable = (1, 6)chosen value = 1backtracked. choosing a new value for (1, 6) if available
chosen value = 2backtracked. choosing a new value for (1, 6) if available
chosen value = 3chosen value is consistent
chosen variable = (0, 8)chosen value = 1backtracked. choosing a new value for (0, 8) if available
chosen value = 2backtracked. choosing a new value for (0, 8) if available
chosen value = 3backtracked. choosing a new value for (0, 8) if available
chosen value = 4backtracked. choosing a new value for (0, 8) if available
chosen value = 5backtracked. choosing a new value for (0, 8) if available
chosen value = 6chosen value is consistent
chosen variable = (1, 1)chosen value = 1chosen value is consistent
chosen variable = (1, 2)chosen value = 1backtracked. choosing a new value for (1, 2) if available
chosen value = 2chosen value is consistent
chosen variable = (1, 7)chosen value = 1backtracked. choosing a new value for (1, 7) if available
chosen value = 2backtracked. choosing a new value for (1, 7) if available
chosen value = 3backtracked. choosing a new value for (1, 7) if available
chosen value = 4chosen value is consistent
chosen variable = (2, 6)chosen value = 1backtracked. choosing a new value for (2, 6) if available
chosen value = 2backtracked. choosing a new value for (2, 6) if available
chosen value = 3backtracked. choosing a new value for (2, 6) if available
chosen value = 4backtracked. choosing a new value for (2, 6) if available
chosen value = 5chosen value is consistent
chosen variable = (2, 2)chosen value = 1backtracked. choosing a new value for (2, 2) if available
chosen value = 2backtracked. choosing a new value for (2, 2) if available
chosen value = 3chosen value is consistent
chosen variable = (0, 2)chosen value = 1backtracked. choosing a new value for (0, 2) if available
chosen value = 2backtracked. choosing a new value for (0, 2) if available
chosen value = 3backtracked. choosing a new value for (0, 2) if available
chosen value = 4backtracked. choosing a new value for (0, 2) if available
chosen value = 5chosen value is consistent
chosen variable = (2, 0)chosen value = 1backtracked. choosing a new value for (2, 0) if available
chosen value = 2backtracked. choosing a new value for (2, 0) if available
chosen value = 3backtracked. choosing a new value for (2, 0) if available
chosen value = 4chosen value is consistent
chosen variable = (2, 1)chosen value = 1backtracked. choosing a new value for (2, 1) if available
chosen value = 2backtracked. choosing a new value for (2, 1) if available
chosen value = 3backtracked. choosing a new value for (2, 1) if available

chosen value = 4backtracked. choosing a new value for (2, 1) if available
chosen value = 5backtracked. choosing a new value for (2, 1) if available
chosen value = 6backtracked. choosing a new value for (2, 1) if available
chosen value = 7backtracked. choosing a new value for (2, 1) if available
chosen value = 8backtracked. choosing a new value for (2, 1) if available
chosen value = 9chosen value is consistent
chosen variable = (2, 4)chosen value = 1backtracked. choosing a new value for (2, 4) if available
chosen value = 2chosen value is consistent
chosen variable = (3, 8)chosen value = 1chosen value is consistent
chosen variable = (3, 3)chosen value = 1backtracked. choosing a new value for (3, 3) if available
chosen value = 2backtracked. choosing a new value for (3, 3) if available
chosen value = 3backtracked. choosing a new value for (3, 3) if available
chosen value = 4backtracked. choosing a new value for (3, 3) if available
chosen value = 5backtracked. choosing a new value for (3, 3) if available
chosen value = 6backtracked. choosing a new value for (3, 3) if available
chosen value = 7backtracked. choosing a new value for (3, 3) if available
chosen value = 8backtracked. choosing a new value for (3, 3) if available
chosen value = 9chosen value is consistent
chosen variable = (4, 1)chosen value = 1backtracked. choosing a new value for (4, 1) if available
chosen value = 2backtracked. choosing a new value for (4, 1) if available
chosen value = 3backtracked. choosing a new value for (4, 1) if available
chosen value = 4backtracked. choosing a new value for (4, 1) if available
chosen value = 5backtracked. choosing a new value for (4, 1) if available
chosen value = 6chosen value is consistent
chosen variable = (4, 3)chosen value = 1backtracked. choosing a new value for (4, 3) if available
chosen value = 2backtracked. choosing a new value for (4, 3) if available
chosen value = 3backtracked. choosing a new value for (4, 3) if available
chosen value = 4backtracked. choosing a new value for (4, 3) if available
chosen value = 5backtracked. choosing a new value for (4, 3) if available
chosen value = 6backtracked. choosing a new value for (4, 3) if available
chosen value = 7chosen value is consistent
chosen variable = (4, 8)chosen value = 1backtracked. choosing a new value for (4, 8) if available
chosen value = 2backtracked. choosing a new value for (4, 8) if available
chosen value = 3backtracked. choosing a new value for (4, 8) if available
chosen value = 4chosen value is consistent
chosen variable = (5, 1)chosen value = 1backtracked. choosing a new value for (5, 1) if available
chosen value = 2backtracked. choosing a new value for (5, 1) if available
chosen value = 3chosen value is consistent

chosen variable = (3, 1)chosen value = 1backtracked. choosing a new value for (3, 1) if available
chosen value = 2backtracked. choosing a new value for (3, 1) if available
chosen value = 3backtracked. choosing a new value for (3, 1) if available
chosen value = 4backtracked. choosing a new value for (3, 1) if available
chosen value = 5chosen value is consistent
chosen variable = (3, 0)chosen value = 1backtracked. choosing a new value for (3, 0) if available
chosen value = 2backtracked. choosing a new value for (3, 0) if available
chosen value = 3backtracked. choosing a new value for (3, 0) if available
chosen value = 4backtracked. choosing a new value for (3, 0) if available
chosen value = 5backtracked. choosing a new value for (3, 0) if available
chosen value = 6backtracked. choosing a new value for (3, 0) if available
chosen value = 7backtracked. choosing a new value for (3, 0) if available
chosen value = 8chosen value is consistent
chosen variable = (3, 5)chosen value = 1backtracked. choosing a new value for (3, 5) if available
chosen value = 2backtracked. choosing a new value for (3, 5) if available
chosen value = 3chosen value is consistent
chosen variable = (0, 5)chosen value = 1backtracked. choosing a new value for (0, 5) if available
chosen value = 2backtracked. choosing a new value for (0, 5) if available
chosen value = 3backtracked. choosing a new value for (0, 5) if available
chosen value = 4backtracked. choosing a new value for (0, 5) if available
chosen value = 5backtracked. choosing a new value for (0, 5) if available
chosen value = 6backtracked. choosing a new value for (0, 5) if available
chosen value = 7backtracked. choosing a new value for (0, 5) if available
chosen value = 8backtracked. choosing a new value for (0, 5) if available
chosen value = 9chosen value is consistent
chosen variable = (0, 4)chosen value = 1backtracked. choosing a new value for (0, 4) if available
chosen value = 2backtracked. choosing a new value for (0, 4) if available
chosen value = 3chosen value is consistent
chosen variable = (4, 0)chosen value = 1backtracked. choosing a new value for (4, 0) if available
chosen value = 2chosen value is consistent
chosen variable = (4, 5)chosen value = 1backtracked. choosing a new value for (4, 5) if available
chosen value = 2backtracked. choosing a new value for (4, 5) if available
chosen value = 3backtracked. choosing a new value for (4, 5) if available
chosen value = 4backtracked. choosing a new value for (4, 5) if available
chosen value = 5backtracked. choosing a new value for (4, 5) if available
chosen value = 6backtracked. choosing a new value for (4, 5) if available
chosen value = 7backtracked. choosing a new value for (4, 5) if available
chosen value = 8chosen value is consistent

chosen variable = (5, 3)chosen value = 1chosen value is consistent
chosen variable = (5, 5)chosen value = 1backtracked. choosing a new value for (5, 5) if available
chosen value = 2chosen value is consistent
chosen variable = (5, 7)chosen value = 1backtracked. choosing a new value for (5, 7) if available
chosen value = 2backtracked. choosing a new value for (5, 7) if available
chosen value = 3backtracked. choosing a new value for (5, 7) if available
chosen value = 4backtracked. choosing a new value for (5, 7) if available
chosen value = 5backtracked. choosing a new value for (5, 7) if available
chosen value = 6backtracked. choosing a new value for (5, 7) if available
chosen value = 7backtracked. choosing a new value for (5, 7) if available
chosen value = 8chosen value is consistent
chosen variable = (5, 6)chosen value = 1backtracked. choosing a new value for (5, 6) if available
chosen value = 2backtracked. choosing a new value for (5, 6) if available
chosen value = 3backtracked. choosing a new value for (5, 6) if available
chosen value = 4backtracked. choosing a new value for (5, 6) if available
chosen value = 5backtracked. choosing a new value for (5, 6) if available
chosen value = 6backtracked. choosing a new value for (5, 6) if available
chosen value = 7chosen value is consistent
chosen variable = (6, 0)chosen value = 1backtracked. choosing a new value for (6, 0) if available
chosen value = 2backtracked. choosing a new value for (6, 0) if available
chosen value = 3backtracked. choosing a new value for (6, 0) if available
chosen value = 4backtracked. choosing a new value for (6, 0) if available
chosen value = 5chosen value is consistent
chosen variable = (6, 5)chosen value = 1backtracked. choosing a new value for (6, 5) if available
chosen value = 2backtracked. choosing a new value for (6, 5) if available
chosen value = 3backtracked. choosing a new value for (6, 5) if available
chosen value = 4chosen value is consistent
chosen variable = (7, 3)chosen value = 1backtracked. choosing a new value for (7, 3) if available
chosen value = 2backtracked. choosing a new value for (7, 3) if available
chosen value = 3backtracked. choosing a new value for (7, 3) if available
chosen value = 4backtracked. choosing a new value for (7, 3) if available
chosen value = 5chosen value is consistent
chosen variable = (7, 7)chosen value = 1backtracked. choosing a new value for (7, 7) if available
chosen value = 2backtracked. choosing a new value for (7, 7) if available
chosen value = 3backtracked. choosing a new value for (7, 7) if available
chosen value = 4backtracked. choosing a new value for (7, 7) if available
chosen value = 5backtracked. choosing a new value for (7, 7) if available
chosen value = 6backtracked. choosing a new value for (7, 7) if available

chosen value = 7backtracked. choosing a new value for (7, 7) if available
chosen value = 8backtracked. choosing a new value for (7, 7) if available
chosen value = 9chosen value is consistent
chosen variable = (7, 4)chosen value = 1backtracked. choosing a new value for (7, 4) if available
chosen value = 2backtracked. choosing a new value for (7, 4) if available
chosen value = 3backtracked. choosing a new value for (7, 4) if available
chosen value = 4backtracked. choosing a new value for (7, 4) if available
chosen value = 5backtracked. choosing a new value for (7, 4) if available
chosen value = 6backtracked. choosing a new value for (7, 4) if available
chosen value = 7backtracked. choosing a new value for (7, 4) if available
chosen value = 8chosen value is consistent
chosen variable = (6, 4)chosen value = 1backtracked. choosing a new value for (6, 4) if available
chosen value = 2backtracked. choosing a new value for (6, 4) if available
chosen value = 3backtracked. choosing a new value for (6, 4) if available
chosen value = 4backtracked. choosing a new value for (6, 4) if available
chosen value = 5backtracked. choosing a new value for (6, 4) if available
chosen value = 6backtracked. choosing a new value for (6, 4) if available
chosen value = 7backtracked. choosing a new value for (6, 4) if available
chosen value = 8backtracked. choosing a new value for (6, 4) if available
chosen value = 9chosen value is consistent
chosen variable = (7, 2)chosen value = 1backtracked. choosing a new value for (7, 2) if available
chosen value = 2backtracked. choosing a new value for (7, 2) if available
chosen value = 3backtracked. choosing a new value for (7, 2) if available
chosen value = 4backtracked. choosing a new value for (7, 2) if available
chosen value = 5backtracked. choosing a new value for (7, 2) if available
chosen value = 6chosen value is consistent
chosen variable = (6, 2)chosen value = 1backtracked. choosing a new value for (6, 2) if available
chosen value = 2backtracked. choosing a new value for (6, 2) if available
chosen value = 3backtracked. choosing a new value for (6, 2) if available
chosen value = 4backtracked. choosing a new value for (6, 2) if available
chosen value = 5backtracked. choosing a new value for (6, 2) if available
chosen value = 6backtracked. choosing a new value for (6, 2) if available
chosen value = 7backtracked. choosing a new value for (6, 2) if available
chosen value = 8chosen value is consistent
chosen variable = (6, 6)chosen value = 1backtracked. choosing a new value for (6, 6) if available
chosen value = 2backtracked. choosing a new value for (6, 6) if available
chosen value = 3backtracked. choosing a new value for (6, 6) if available
chosen value = 4backtracked. choosing a new value for (6, 6) if available
chosen value = 5backtracked. choosing a new value for (6, 6) if available
chosen value = 6chosen value is consistent

chosen variable = (7, 8)chosen value = 1backtracked. choosing a new value for (7, 8) if available
chosen value = 2backtracked. choosing a new value for (7, 8) if available
chosen value = 3chosen value is consistent
chosen variable = (8, 0)chosen value = 1backtracked. choosing a new value for (8, 0) if available
chosen value = 2backtracked. choosing a new value for (8, 0) if available
chosen value = 3chosen value is consistent
chosen variable = (8, 4)chosen value = 1chosen value is consistent
chosen variable = (8, 6)chosen value = 1backtracked. choosing a new value for (8, 6) if available
chosen value = 2backtracked. choosing a new value for (8, 6) if available
chosen value = 3backtracked. choosing a new value for (8, 6) if available
chosen value = 4backtracked. choosing a new value for (8, 6) if available
chosen value = 5backtracked. choosing a new value for (8, 6) if available
chosen value = 6backtracked. choosing a new value for (8, 6) if available
chosen value = 7backtracked. choosing a new value for (8, 6) if available
chosen value = 8chosen value is consistent
chosen variable = (8, 7)chosen value = 1backtracked. choosing a new value for (8, 7) if available
chosen value = 2backtracked. choosing a new value for (8, 7) if available
chosen value = 3backtracked. choosing a new value for (8, 7) if available
chosen value = 4backtracked. choosing a new value for (8, 7) if available
chosen value = 5chosen value is consistent
126|439|785
349|875|612
578|621|493

261|943|857
934|758|261
785|162|934

612|394|578
493|587|126
857|216|349
running time of MRV + value choosing from 1 to 9 = 1.5898540019989014

Process finished with exit code 0

ایده برای بهتر کردن پروژه:

می توانیم از الگوریتم های Local searchهم استفاده کنیم و زمان آن ها را با الگوریتم های زده شده مقایسه کنیم.

References:

به جز استفاده از شبه کد های Backtracking وایده های موجود در کتاب از منبع دیگری استفاده نشد و از کدی کمک گرفته نشد. فقط صفحه سودوکوی ورودی اولیه را از یک سایت برداشتم!!