

بسمه تعالی

هوش مصنوعی

مسائل ارضاء محدودیتها - ۲

نیمسال اول ۱۴۰۲-۱۴۰۱

دکتر مازیار پالهنک

آزمایشگاه هوش مصنوعی

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

یادآوری

توی این مسایل باید مقادیر مناسبی که به متغیرها نسبت داده میشود را پیدا کنیم

اگه مقادیری که متغیرها نسبت میدیم با توجه به محدودیت هاشون باشه و محدودیت هاروارضا کنن میگیریم انتساب سازگاره

اگه همه ی متغیرهای مسئله مقدار گرفته باشند انتساب کامل است ولی اگه تعدادی گرفته باشند و تعدادی نه میگیریم جزئی

حل مسئله یک انتساب کامل و سازگار است

در مثال رنگ امیزی نقشه استرالیا متغیرهای ما ایالت های ما بود و هدف ما این بود که به هریالتی یک رنگ اختصاص بدیم طوری که رنگ هیچ دوایالت مجاوری یکسان نشه

دامنه ی متغیرها سه رنگ قرمز و سبز و ابی

مسائل ارضای محدودیت را میشه بایک گراف ارضای محدودیت نمایش داد

در این گراف رئوس متناظر با متغیرهای ما هستند و یال ها متناظر با محدودیت هایی که بین دو یا چند متغیر وجود داره

برای پیداکردن حل میشه از مسئله ی جستجو استفاده کرد فقط باید مسئله را به درستی تدوین کنیم

حالت اولیه و اعمال و هدف ما و هزینه وتابع تالی چی هست

تنوع متغیرها: گسسته و پیوسته

محدودیت ها: یکتایی و دوتایی و محدودیت جهانی

یک جستجوی مناسب برای پیداکردن حل در مسائل ارضای محدودیت جستجوی عقبگرد است

این جستجو به صورت عمقی عمل میکنه

- متغیرها، دامنه ها، محدودیتها
- انتساب سازگار
- انتساب کامل، جزئی
- مثال رنگ آمیزی نقشه
- گراف محدودیت
- حل بصورت یک مسئله جستجو
- تنوع متغیرها
- تنوع محدودیتها
- جستجوی عقبگرد
- متغیر محدود شده ییشینه
- متغیر محدود کن ییشینه
- مقدار محدود کن کمینه
- چک جلو
- سازگاری کمان

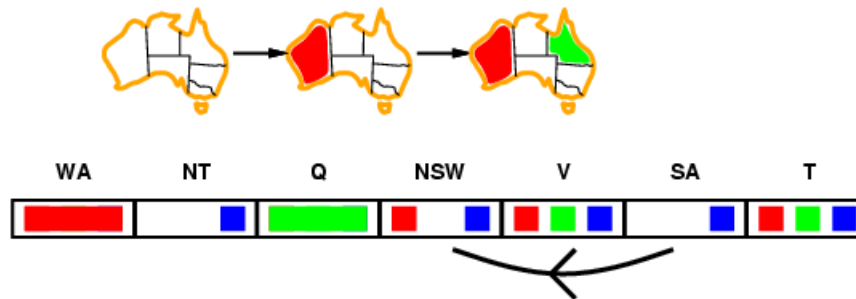
هربار یکی از تالی ها را بسط میدی ولی چون اینجا تعداد متغیرها مون ثابت است و در هر مرحله در جستجو فقط به یکی از متغیرها مقدار میدیم پس جستجوی ما تا عمق n

میره پس نگران نیسیم که توی حلقه ی بی نهایت قرار بگیره و تا عمق نامحدود بره و اینکه جستجوی عمق نخست از نظر حافظه مناسب است این روش جستجو میتونه برای ارضای محدودیت ها مناسب باشه

در حالت اولیه از راس ریشه شروع میکنیم که هیچ متغیری مقداری نداره بعدش متغیری که مقدار نگرفته بش مقادیر مناسب میدیم برای ساده شدن کار در هرمرحله فقط یک متغیری که مقدار نداره را انتخاب میکنیم هربار که به یک متغیرمیخواهیم مقدار بدیم چک میکنیم با متغیرهای قبلی که مقدار گرفتن سازگار است یا نه؟ اگر ناسازگار بود عقبگرد میکنیم و یه مقداردیگه ای را برای اون متغیر چک میکنیم

سازگاری کمان

- کمان منظور یالی است در گراف محدودیت
- کمان $X \rightarrow Y$ سازگار گفته می شود اگر برای هر مقدار X در دامنه X مقدار Y در دامنه Y وجود داشته باشد که با آن سازگار باشد.
- مثال کمان سازگار



جستجوی عقبگرد را میتوانیم بهینه تر پیاده سازی کنیم یعنی بگیم کدام یکی از متغیرها را آگه اول انتخاب کنیم بهتر است؟ در این حالت از یک مکاشفه استفاده کردیم ب اسم متغیر محدود شده ی بیشینه که میگه متغیری رازودتر انتخاب کن که دامنه ش مقادیری کمتری داره تا بتونیم زودتر متوجه شکست احتمالی بشیم تا آگه قراره شکست بخوریم زودتر متوجه شیم

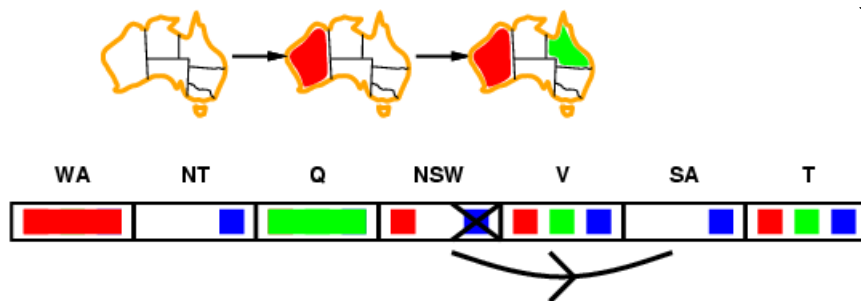
آگه چندتا متغیر محدود شده ی بیشینه داشته باشیم از بین اون متغیرها بهتر است متغیر محدودکن بیشینه را انتخاب کنیم یعنی اون متغیری را انتخاب کنیم که تعداد متغیرهایی که به واسطه ی مقداردهی به این متغیر نمیتونن مقدار بگیرن بیشتر باشه یعنی درجه یا تعدادیال هایی که بهش وصل هستنند بیشینه باشه

به این مکاشفه ی درجه هم میگیم که کمک میکنه فضای جستجو کاهش پیدا کنه و زودتر به حل برسیم

بعد از اینکه متغیر پیدا شد ما از مکاشفه ی محدودکن کمینه استفاده میکنیم یعنی از بین مقادیری که میتونیم انتخاب کنیم اونو را انتخاب کنیم که کمتر برای متغیرهای دیگه محدودیت ایجاد کنه یعنی به متغیرهای آینده اجازه بدیم مقادیر مناسب بگیرن

■ کمان $X \rightarrow Y$ سازگار گفته می شود اگر برای هر مقدار X در دامنه X مقدار Y در دامنه Y وجود داشته باشد که با آن سازگار باشد.

■ مثال کمان ناسازگار

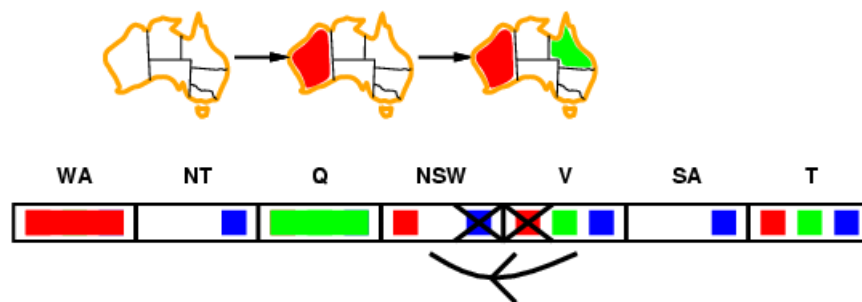


مکاشفه ی چک به جلو میگفت هر موقع به یک متغیر مقدار نسبت میدیم میرفتیم مقدار نسبت داده شده را از دامنه ی همسایه های اون متغیر حذف میکردیم که باعث میشد دامنه کوچک تر شه و در نهایت سریعتر جستجو به جواب برسه
مشکل چک به جلو: تشخیص ندادن یک سری ناسازگاری برای رفع این مشکل از سازگاری کمان استفاده میکنیم
سازگاری کمان یعنی بین هردو متغیری که در گراف محدودیت به هم مرتبط هستند نگاه کنیم که آیا برای هر مقداری که در دامنه ی متغیر اول هست مقداری پیدامیشه در دامنه ی متغیر دوم که به متغیر دوم نسبت بدیم و محدودیت ها هنوز ارضا شوند اگه برای یکی از مقادیر متغیر اول نتوانستیم مقداری در دامنه ی متغیر دوم پیدا کنیم که سازگار بشه اون مقدار را از دامنه ی متغیر اول حذف میکنیم به این صورت سازگاری کمان ایجاد میشه

- سعی کن همه کمانها را سازگار کنی
- اگر سعی کنیم NT و SA را سازگار کنیم، به دامنه تهی می رسیم.
- سازگاری کمان می تواند قبل از جستجو به عنوان پیش پردازش اعمال شود.
- یا بعد از هر انتساب

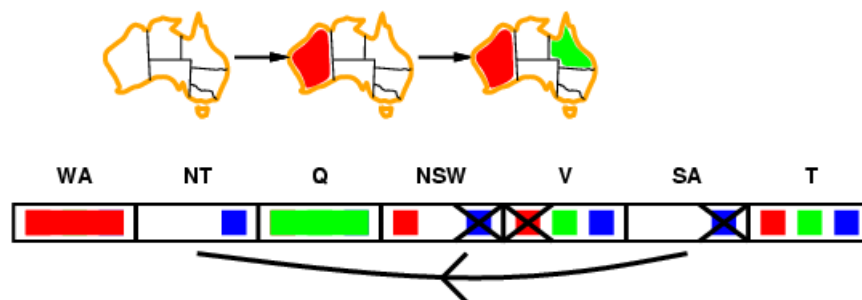
سازگاری کمان

- اگر X مقداری را از دست داد لازم است همسایگان آن چک شوند.
- چون حذف آن مقدار می تواند سازگاری کمان قبلی را از بین ببرد.



سازگاری کمان

■ سازگاری کمان شکست را زودتر از چک جلو تشخیص می دهد.



سازگاری کمان

function AC-3(*csp*) **returns** false if an inconsistency is found and true otherwise
queue \leftarrow a queue of arcs, initially all the arcs in *csp*

while *queue* is not empty **do**
 (*X_i*, *X_j*) \leftarrow POP(*queue*)
 if REVISE(*csp*, *X_i*, *X_j*) **then**
 if size of *D_i* = 0 **then return** false
 for each *X_k* **in** *X_i*.NEIGHBORS - {*X_j*} **do**
 add (*X_k*, *X_i*) to *queue*
return true

function REVISE(*csp*, *X_i*, *X_j*) **returns** true iff we revise the domain of *X_i*
revised \leftarrow false
for each *x* **in** *D_i* **do**
 if no value *y* in *D_j* allows (*x*, *y*) to satisfy the constraint between *X_i* and *X_j* **then**
 delete *x* from *D_i*
 revised \leftarrow true
return *revised*

دامنه ی
D_i
 تهی شد یعنی هیچ
 مقداری دیگه نمیتونه
 بگیره

اگه دامنه ی ایکس
 ای را اصلاح کنیم

بقیه ی همسایه های
x_i
 رو برو بررسی کن

سازگاریهای دیگر

سازگاری رأس:

- یک متغیر (متناظر با یک رأس در گراف م.ا.م.) سازگار-رأس است اگر همه مقادیر دامنه آن محدودیتهای یکتائی را ارضا کنند.
- مثلاً اگر مردم SA رنگ سبز را دوست ندارند باید از دامنه آن حذف شود.
- یک گراف سازگار-رأس نامیده می شود اگر همه رئوس آن سازگار-رأس باشند.

وقتی حذف کنیم رنگ
سبز را بقیه ی
مقادیری که توی
دامنه باقی ماندند
سازگاری رأس دارند

سازگاریهای دیگر

- **سازگاری مسیر:**
- سازگاری کمان امکان دارد دامنه متغیرها را به یک عضو کاهش داده و پاسخ مسئله را بیابد،
- یا دامنه برخی متغیرها بدون عضو بماند و مشخص شود که مسئله بدون پاسخ است.
- اما همواره برخی مشکلات را متوجه نمی شود.
- مثلاً اگر در مسئله رنگ آمیزی نقشه دامنه هرمتغیر دارای دو رنگ {آبی، سبز} باشد.
- گراف سازگاری کمان دارد ولی مسئله حل شدنی نیست.

سازگاریهای دیگر

- دو متغیر $\{X_i, X_j\}$ نسبت به متغیر X_m سازگار-مسیر هستند اگر:
- برای هر انتساب $\{X_i=a, X_j=b\}$ که با محدودیتهای (X_i, X_j) سازگار است، انتسابی برای X_m وجود داشته باشد که محدودیتهای (X_i, X_m) و (X_m, X_j) را ارضا کند.
- بطور مثال تلاش برای سازگاری مسیر (WA, SA) نسبت به NT در حالتی که دامنه متغیرها دارای دو رنگ $\{\text{آبی و قرمز}\}$ باشد.
- دو انتساب سازگار $\{WA=\text{red}, SA=\text{blue}\}$ یا $\{WA=\text{blue}, SA=\text{red}\}$
- هیچ رنگی برای NT باقی نمی ماند.

سازگاریهای دیگر

- **سازگاری - k :**
- یک م.ا.م. سازگار- k است اگر برای هر انتساب سازگار به $k-1$ متغیر، یک انتساب سازگار برای هر متغیر k ام وجود داشته باشد.
- سازگاری-۱ همانند سازگاری رأس
- سازگاری-۲ همانند سازگاری کمان
- سازگاری-۳ همانند سازگاری مسیر
- یک م.ا.م. **سازگار- k -قوی** است اگر سازگار- k ، سازگار- $k-1$ ، سازگار- $k-2$ ،، تا سازگار-۱ باشد.

سازگاریهای دیگر

■ محدودیتهای جهانی:

■ همانگونه که قبلاً اشاره شد یک محدودیت جهانی شامل محدودیتی است که شامل چند متغیر (نه الزاماً همه) گردد.

■ بطور مثال **Alldiff**

■ روشی ساده برای تشخیص ناسازگاری در Alldiff:

■ اگر m متغیر در محدودیت شرکت دارند و اگر آنها روی هم دارای n مقدار متفاوت هستند و $m > n$ ، در این صورت محدودیت قابل ارضاء نیست.

سازگاریهای دیگر

- منجر به یک راهکار ساده:
- هر متغیر تک مقدار را از محدودیت حذف نمائید،
- مقدار آن متغیرها را از دامنه دیگر متغیرها حذف نمائید،
- مادامی که متغیر تک مقداری باقی مانده این کار را تکرار کنید،
- هر زمان یک دامنه تهی تشخیص داده شد، یا تعداد متغیرها بیش از تعداد مقادیر باقی مانده بود یک ناسازگاری تشخیص داده شده است.
- بطور مثال: در مسئله رنگ آمیزی WA،SA،NT با دو رنگ ناسازگاری تشخیص داده می شود (سه متغیر و دو رنگ).

انتخاب یک متغیر
انتساب نشده

انتساب اولیه تهی
است

مسئله ی ارضای
محدودیت

جستجوی عقبگرد

function BACKTRACKING-SEARCH(*csp*) **returns** a solution or *failure*
return BACKTRACK(*csp*, {})

function BACKTRACK(*csp*, *assignment*) **returns** a solution or *failure*
if *assignment* is complete **then return** *assignment*
var ← SELECT-UNASSIGNED-VARIABLE(*csp*, *assignment*)
for each *value* in ORDER-DOMAIN-VALUES(*csp*, *var*, *assignment*) **do**
 if *value* is consistent with *assignment* **then**
 add {*var* = *value*} to *assignment*
 inferences ← INFERENCE(*csp*, *var*, *assignment*)
 if *inferences* ≠ *failure* **then**
 add *inferences* to *csp*
 result ← BACKTRACK(*csp*, *assignment*)
 if *result* ≠ *failure* **then return** *result*
 remove *inferences* from *csp*
 remove {*var* = *value*} from *assignment*
return failure

استنتاج
به کاربردن سازگاری
راس و کمان و مسیر
و...

ایا این مقدار سازگار
هست با انتسابی که
تالان داشتیم؟

حذف استنتاج هایی
که تالان انجام دادیم

مازیار پالهنک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

15

به صورت بازگشتی
و عمقی میریم جلو

اگه همه ی مقادیر چک شد
و سازگاری پیدا نشد ما
شکست برمیگردانیم

ایا میتونیم از دامنه
ی متغیرها مقادیری
را حذف کنیم؟

جستجوی محلی برای ارضاء محدودیتها

- جستجوهای محلی از حالت کامل استفاده می کنند.
- در این حالت وضعیتی که همه متغیرها مقدار دارند.
- برای اعمال به م.ا.م.
- اجازه داشتن حالاتی که محدودیتها ارضا نشده اند.
- تغییر مقدار متغیرها (در جهت ارضاء کردن محدودیتها)
- انتخاب متغیر: بصورت تصادفی هر متغیر ناسازگار انتخاب شود.
- انتخاب مقدار با مکاشفه کمترین برخورد:
- انتخاب مقداری که کمترین محدودیتها را می شکند.

شبیه به الگوریتم تپه نوردی

حداکثر تا این تعداد
مرحله جلو میریم

Figure 6.9

function MIN-CONFLICTS(*csp*, *max_steps*) **returns** a solution or *failure*

inputs: *csp*, a constraint satisfaction problem

max_steps, the number of steps allowed before giving up

current ← an initial complete assignment for *csp*

for *i* = 1 to *max_steps* **do**

if *current* is a solution for *csp* **then return** *current*

var ← a randomly chosen conflicted variable from *csp*.

value ← the value *v* for *var* that minimizes CONFLICTS(*csp*, *var*, *v*, *current*)

set *var* = *value* in *current*

return *failure*

حالت اولیه یک
انتساب کامل است که
لزوما هم سازگار
نیست

اگر حالت فعلی یک
حل است پس جواب
پیدا شده

The MIN-CONFLICTS local search algorithm for CSPs. The initial state may be chosen randomly or by a greedy assignment process that chooses a minimal-conflict value for each variable in turn. The CONFLICTS function counts the number of constraints violated by a particular value, given the rest of the current assignment.

مازیار پالهنک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

17

اگر ماکس استپ تمام
شد شکست
برمیگردانیم

مقداری را انتخاب
میکنیم که کمترین
تداخل را ایجاد کنه

مثال: ۸ وزیر

- حالات: ۸ وزیر در ۸ ستون (8^8 حالت)
- انتخاب تصادفی یک متغیر
- انتخاب مقداری برای آن متغیر با کمترین تضاد
- امکان استفاده تا چند میلیون وزیر!

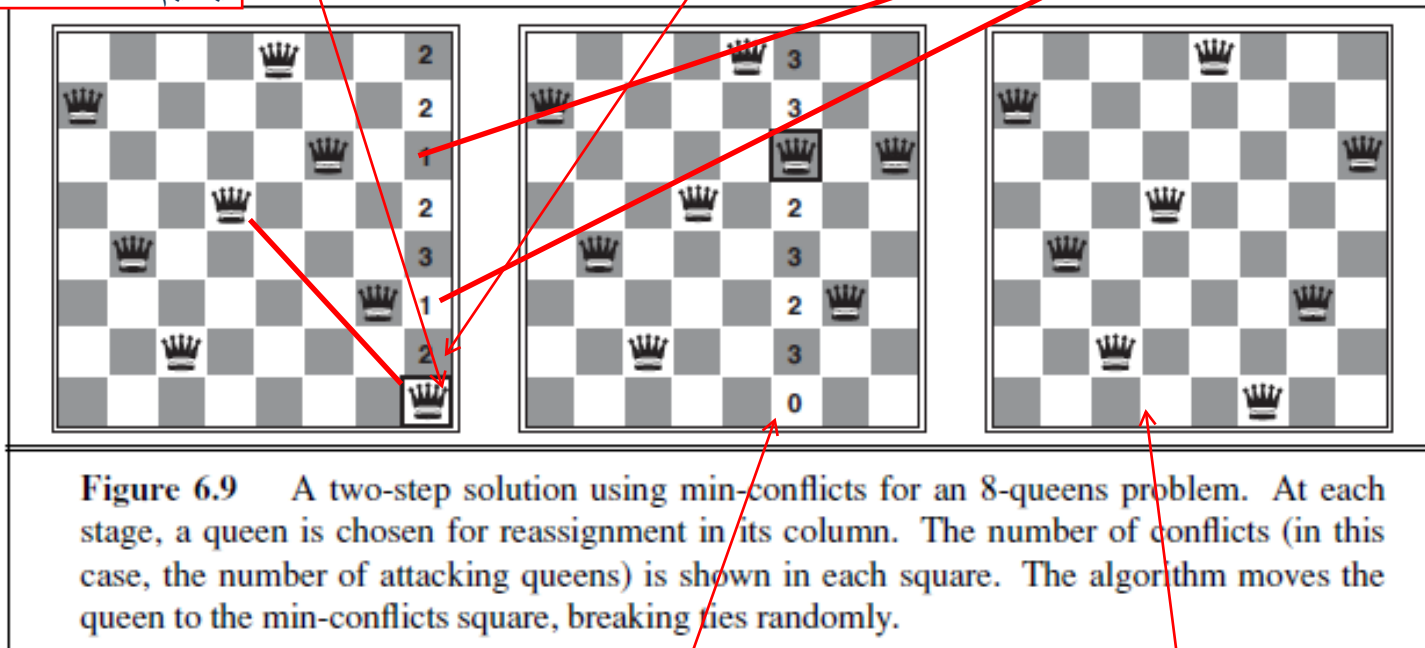
اگر از این مکاشفه
استفاده کنیم تا
چند میلیون وزیر را هم
میتوانیم با این روش
حل کنیم

این متغیر مشکل داره
پس باید مقدارشو
تغییر بدیم
پس توی همون
ستونی که هست
سطرش را عوض
میکنیم

مکاشفه می‌گه مقداری را
برای متغیر انتخاب کن که
کمترین تضاد و کانفلیکت
را ایجاد کنه

اگه وزیر را ببریم به
سطر بالایی باعث
ایجاد دوتا برخورد با
وزیرهای دیگه میشه

بین این دوتا یکی
را به تصادف
انتخاب میکنیم



کمترین تداخل
اینجاست

مازیار پالهنک

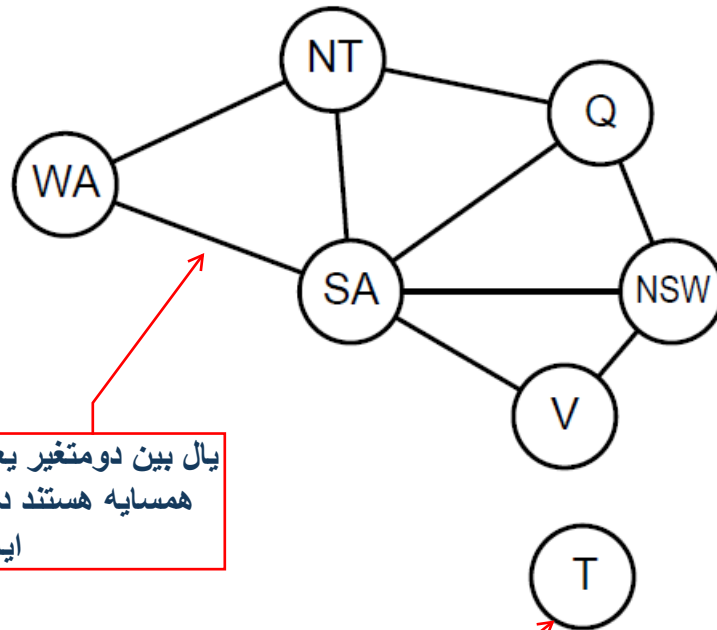
هوش مصنوعی - نیمسال اول ۱۳۰۱-۰۲

19

وزیر ۸
در صفحه قرار گرفتند
و هیچکدام دیگری را
تهدید نمیکنند

شکستن مسئله به تعدادی زیرمسئله و حل اون زیرمسئله ها و درنهایت حل هارا ترکیب کنیم

استفاده از ساختار مسئله



یال بین دو متغیر یعنی همسایه هستند دوتا ایالت

این چون هیچ همسایه ای ندار ما میتونیم رنگ امیزیش را جداگانه انجام بدیم

- تقسیم مسئله به زیر مسائل
- به عنوان مثال رنگ آمیزی T
- یافتن اجزاء متصل در گراف محدودیت مسئله
- هر جزء یک زیر مسئله ارضاء محدودیتها

توی این گراف دوتا جز متصل داریم

کل تعداد متغیرها مون
در کل مساله

■ اگر هر زیر مسئله دارای C متغیر از n متغیر مسئله باشد.

■ تعداد زیر مسائل n/C

■ اگر d اندازه هر دامنه

■ هزینه بدترین حل $O(d^c \cdot n/c)$

■ در مقابل $O(d^n)$

■ d ضریب انشعاب

■ مسئله ای که ممکن است چند میلیون سال طول بکشد در چند ثانیه (یا کسری از ثانیه) حل خواهد شد.

توی هر زیرمساله ما
 C
تا متغیر داریم پس تا عمق سی باید
بریم پایین
اندازه ی دامنه هم دی است یعنی
هرمتغیر
 d
مقدار میتونه به خودش بگیره که
مثل ضریب انشعاب است

حل مساله بدون کمک
گرفتن از زیرمسائل

م.ا.م. با ساختار درختی

- یک گراف محدودیت یک درخت است هر گاه هر دو رأس فقط با یک مسیر به هم وصل باشند.
- هر م.ا.م. با ساختار درختی در زمانی خطی نسبت به تعداد متغیرها قابل حل است.
- یک م.ا.م. سازگار کمان جهتدار تحت یک ترتیب دهی متغیرهای X_1, X_2, \dots, X_n است اگر و تنها اگر X_i با X_j سازگار کمان باشد برای $j > i$

X_j
در ترتیب دهی بعد از
 X_i
قرار گرفته باشد

بین متغیرهای مسئله
یک ترتیب گذاشته

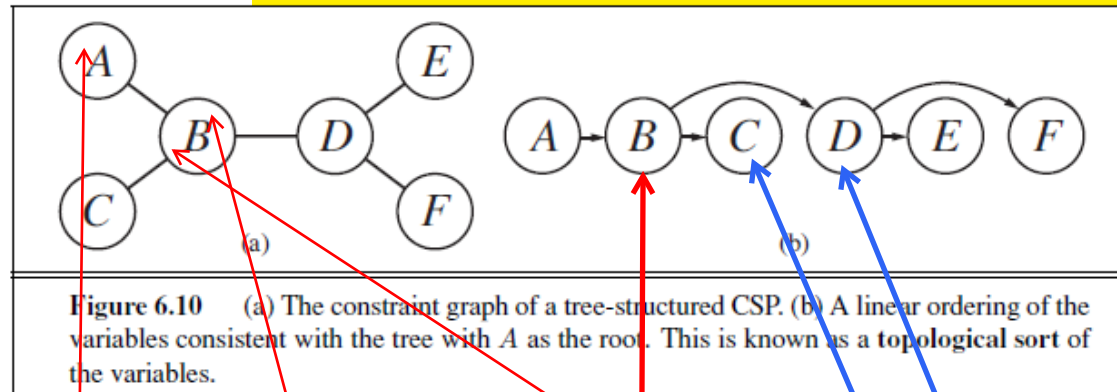
سازگاری جهت دار
است یعنی از متغیری
که اندیش کمتر
است به متغیری که
اندیش بیشتر است

مصنوعی - نیمسال اول ۱۴۰۱-۰۲

با ساختار درختی

با این روش ترتیب متغیرها معلوم شد
سازگاری کمان هم جهت دار است پس
باید چک کنیم آیا
A با B
سازگاری کمان داره؟

- یک متغیر را به عنوان ریشه انتخاب کنید،
- متغیرها را از ریشه به برگ به گونه ای ترتیب دهید که ولی هر رأس قبل از آن در این ترتیب دهی قرار گیرد.
- ترتیب دهی همبندی (topological order)



مازیار پالهنک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

23

این رأس را به
عنوان ریشه انتخاب
کردیم

این رأس فرزند
A
است

رأس بی دوتا فرزند
داره
C, D
پس باید بعد از بی
قرار بگیرن

Figure 6.11

function TREE-CSP-SOLVER(*csp*) **returns** a solution, or *failure*

inputs: *csp*, a CSP with components *X*, *D*, *C*

n ← number of variables in *X*

انتساب تهی

assignment ← an empty assignment

root ← any variable in *X*

X ← TOPOLOGICAL-SORT(*X*, *root*)

ترتیب دهی به راس ها

for *j* = *n* **down to** 2 **do**

MAKE-ARC-CONSISTENT(PARENT(*X_j*), *X_j*)

if it cannot be made consistent **then return** *failure*

for *i* = 1 **to** *n* **do**

assignment[*X_i*] ← any consistent value from *D_i*

if there is no consistent value **then return** *failure*

return *assignment*

ایجاد سازگاری کمان
بین ولی راس و
خودش

وقتی
J=2
میشه ولی راسمون
میشه راس یک که
ریشه است

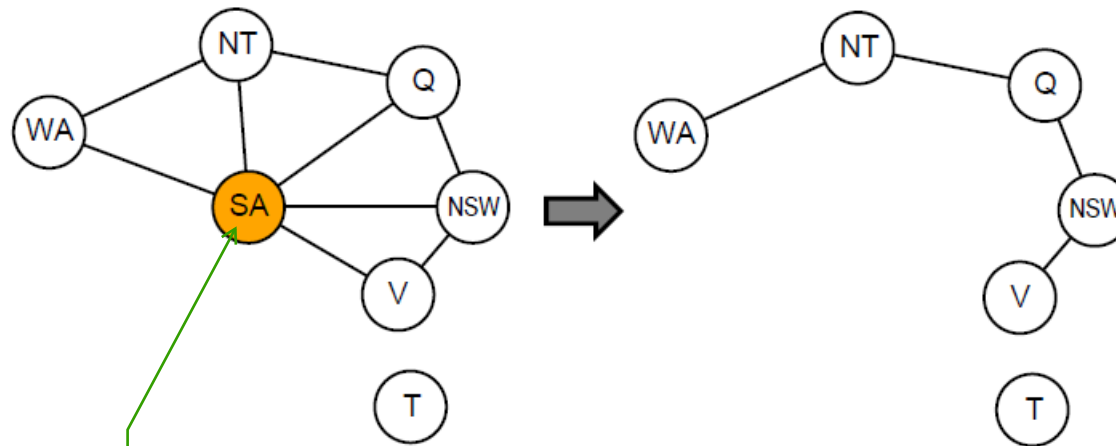
The TREE-CSP-SOLVER algorithm for solving tree-structured CSPs. If the CSP has a solution in linear time; if not, we will detect a contradiction.

اگه سازگاری کلی
ایجاد نشد شکست
برگردان

همه ی گراف ها ساختار درختی ندارند و
 برخی ساختار تقریبا درختی دارند
 مثلا اگه یه نود را از گراف زیر حذف کنیم
 تبدیل به درخت میشه و میتونیم از الگوریتم
 صفحه قبل استفاده کنیم

م.ا.م. با ساختار تقریبا درختی

■ مقدار دهی یک متغیر، حذف مقدار آن از دیگر متغیرها، اجرای
 الگوریتم درختی



این مقدار را ک
 انتخاب کردیم از
 دامنه ی بقیه حذف
 میکنیم

پالهنک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

25

م.ا.م. با ساختار تقریباً درختی

- ممکن است مقدار SA مناسب نباشد نیاز به تکرار
- بطور کلی:

■ انتخاب مجموعه S از کل متغیرهای م.ا.م. به گونه ای که پس از حذف S گراف محدودیت درخت شود.

- برای هر انتساب سازگار متغیرهای S

■ حذف مقادیر ناسازگار با S از دیگر متغیرها،

- اگر مابقی م.ا.م. دارای حل است آنرا به همراه مقادیر S بازگردان

اول یک انتساب سازگار برای متغیرهای توی

S

پیدامیکنیم بعد مقادیر ناسازگار با متغیرهای اس را از متغیرهای دیگه حذف میکنیم بعد میریم برای بقیه ی متغیرها به روش درختی دنبال حل میگردیم اگه یه حل پیدا کردیم اون حل را به همراه مقادیری که برای اس پیدا کردیم برمیگردانیم اگه هم پیدا نکردیم یه انتساب سازگار دیگه را برای متغیرهای مجموعه ی اس پیدامیکنیم و اونها را تست میکنیم

بار پالهنک

26



مازیار پالهنک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

27

خلاصه

اضافه کردن قسمت استنتاج
که سازگاری هایی که داشتیم را
چک میکردیم
اگر اون سازگاری ها باعث میشد
مقادیری از متغیرها حذف شه
حذف میکردیم
این باعث میشد جستجو بهتر
انجام شه

- سازگاری کمان
- معرفی سازگاریهای دیگر،
- الگوریتم جستجوی عقبگرد بصورت کاملتر
- جستجوی محلی
- استفاده از ساختار مسئله

اون گراف های محدودیتی که
ساختار درختی دارند را اگر یک
ترتیب دهی برای متغیرهاشون
در نظر بگیریم و سازگاری کمائی
براشون اعمال کنیم

- ساختار درختی
- ساختار تقریباً درختی

مازیار پالهنک

هوش مصنوعی - نیمسال اول ۱۴۰۱-۰۲

28

پیدا کردن حل با
زمان خطی نسبت
به تعداد متغیرها

حذف چندین راس
از گراف محدودیت
برای تبدیل به
ساختاری درختی



- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوه درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- در تهیه اسلایدها از سایت کتاب استفاده شده است.