مسائل ارضای محدودیت

دانشگاه صنعتی قوچان

محمدحسین سیگاری

مسئله ارضای محدودیت (CSP)

- CSP: Constraint Satisfaction Problem

- مجموعه متناهی از متغیرها: $X_1, X_2, \ldots, X_n$
- مجموعه متناهی از محدودیتها: $C_1, C_2, \ldots, C_m$
- دامنه های ناتهی برای هر یک از متغیرها: $D_{X1}, D_{X2}, \ldots, D_{Xn}$

- پارامترها:
  - n: تعداد متغیرها
  - d: تعداد اعضای مجموعه دامنه هر متغیر

مسئله ارضای محدودیت (CSP)

- حالت (state) در مسئله CSP:
  - انتساب (assignment) مقدار به یک یا چند متغیر
- انتساب سازگار (consistent assignment)
  - انتسابی که هیچ یک از محدودیت ها را نقض نکند
- انتساب کامل (complete assignment)
  - انتسابی که در آن به تمام متغیرها مقداردهی شود
- راه حل (solution):
  - زمانی که تمام متغیرها مقداردهی شوند و هیچ محدودیتی نقض نشود
  - یک راه حل، انتساب کامل و سازگار است.

انواع مسئله ارضای محدودیت (CSP)

انواع مسئله ارضای محدودیت (CSP)

محدودیت خطی

هر محدودیت به شکل یک رابطه خطی بین متغیرها

محدودیت غیرخطی

هر محدودیت به شکل یک رابطه غیرخطی بین متغیرها

- محدودیت خطی: $x+5y<8$ ،$x \neq y$
- محدودیت غیرخطی: $x^2+\sin(x)-y>0$

انواع مسئله ارضای محدودیت (CSP)

محدودیت های گسسته

۱- دامنه محدود
۲- دامنه نامحدود

محدودیت های پیوسته

دامنه ناشماراست

- مسائل با محدودیت های گسسته و دامنه محدود:
  - مرتبه زمانی حل مسئله: $O(d^n)$ (d اندازه دامنه و n تعداد متغیر)
- مسائل با محدودیت های خطی در حالت گسسته (دامنه نامحدود) و حالت پیوسته
  - با روش LP (Linear Prog.) حل مسائل $O(n)$ خواهد بود

انواع مسئله ارضای محدودیت (CSP)

محدودیت های یگانه (Unary)

- می توان آن محدودیت را در دامنه آن متغیر لحاظ کرد

محدودیت های دوگانه (Binary)

- محدودیت های به صورت دو به دو بین متغیرها تعریف می شود. مثل $x \neq y$

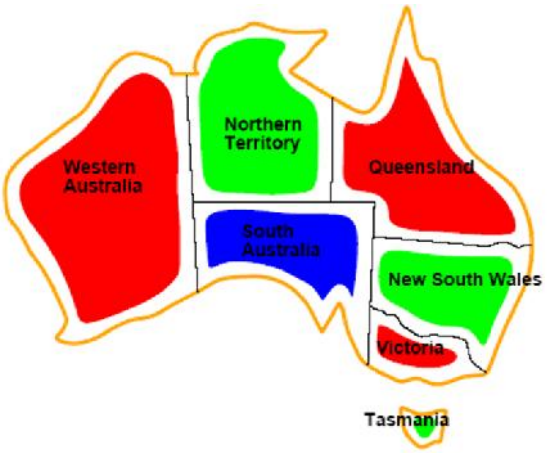
محدودیت های چندگانه (N-ary)

- معمولا محدودیت چندگانه به تعدادی محدودیت دوگانه قابل تبدیل است

رنگ آمیزی نقشه (Map Coloring)

- رنگ آمیزی نقشه با رنگ های آبی، قرمز و سبز
- شرط: هیچ دو شهر مجاور، رنگ یکسان نداشته باشند
- اجزای این CSP چیست؟
  - متغیرها؟
  - دامنه متغیرها؟
  - محدودیت ها؟

رنگ آمیزی نقشه (Map Coloring)



نمونه مسائل ارضای محدودیت

- رنگ آمیزی نقشه
- مسئله ۸ وزیر
- جدول کلمات متقاطع
- جدول Sudoku
- مسئله فروشنده دوره گرد (TSP)
- مسائل زمان بندی (Scheduling):
  - تنظیم برنامه کلاسی
  - تنظیم برنامه پروازهای فرودگاه

مطلق / اولویت دار
خطی / غیرخطی
گسسته (محدود/نامحدود) / پیوسته
یگانه / دوگانه / چندگانه

نمونه مسائل ارضای محدودیت

مطلق / اولویت دار
خطی / غیرخطی
گسسته (محدود/نامحدود) / پیوسته
یگانه / دوگانه / چندگانه

- رنگ آمیزی نقشه
  - محدودیت مطلق
  - محدودیت خطی
  - محدودیت گسسته با دامنه محدود
  - محدودیت چندگانه (هیچ دو شهر مجاور همرنگ نباشند)
- مسئله ۸ وزیر
  - محدودیت مطلق
  - محدودیت خطی
  - محدودیت گسسته با دامنه محدود
  - محدودیت چندگانه (در سطر و ستون هیچ وزیری، وزیر دیگری قرار نگیرد)
- جدول کلمات متقاطع
  - محدودیت مطلق
  - محدودیت غیرخطی (چرا که باید کلمات درج شده در جدول بامعنا باشد)
  - محدودیت های گسسته با دامنه محدود (اما با دامنه بسیاربزرگ)
  - محدودیت دوگانه و چندگانه (محدودیت یک حرف با حروف دیگر کلمه)

نمونه مسائل ارضای محدودیت

مطلق / اولویت دار
خطی / غیرخطی
گسسته (محدود/نامحدود) / پیوسته
یگانه / دوگانه / چندگانه

- جدول Sudoku
  - محدودیت مطلق
  - محدودیت خطی
  - محدودیت گسسته با دامنه محدود
  - محدودیت چندگانه (هیچ عددی نباید در یک سطر، ستون و یک مربع کوچک دوبار تکرار شود)
- مسئله فروشنده دوره گرد (TSP)
  - محدودیت مطلق (با توجه به NP-Hard بودن مسئله، در مسائل واقعی با محدودیت اولویت دار فرض می شود)
  - محدودیت خطی (هیچ شهری نباید دوبار در مسیر باشد) و محدودیت غیرخطی (کل هزینه مسیر کمینه شود)
  - محدودیت پیوسته
  - محدودیت چندگانه (در مسیر پیموده شده، هیچ شهری نباید دوبار مشاهده شود)
- مسائل زمان بندی (Scheduling): مثل تنظیم برنامه کلاسی
  - محدودیت اولویت دار
  - محدودیت خطی و غیرخطی (عدم تداخل کلاس‌ها محدودیت خطی، تنظیم بهینه آنها محدودیت غیرخطی)
  - محدودیت های گسسته و پیوسته
  - محدودیت یگانه، دوگانه و چندگانه

رمزنگاری ریاضی (Cryptarithmatic)

```
  TWO
+TWO
-----
 FOUR
```

- جایگذاری هر حرف با یکی از ارقام صفر تا ۹
- شرط: هیچ دو حرف غیریکسان، ارقام یکسانی نداشته باشند

- اجزای این CSP چیست؟
  - متغیرها؟
  - دامنه متغیرها؟
  - محدودیت ها؟
    - دو نوع محدودیت!

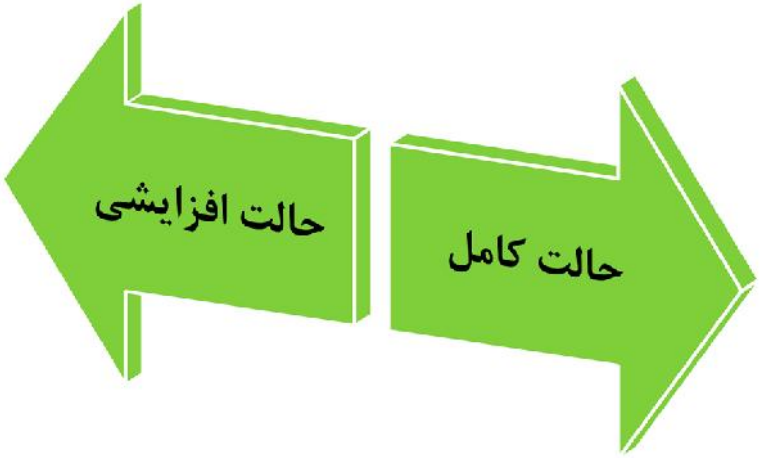
رمزنگاری ریاضی (Cryptarithmatic)

```
  TWO
+TWO
-----
 FOUR
```

- **۷ جواب ممکن:**

- ۱۸۷۶=۹۳۸+۹۳۸
- ۱۸۵۶=۹۲۸+۹۲۸
- ۱۷۳۴=۸۶۷+۸۶۷
- ۱۶۹۲=۸۴۶+۸۴۶
- ۱۶۷۲=۸۳۶+۸۳۶
- ۱۵۳۰=۷۶۵+۷۶۵
- ۱۴۶۸=۷۳۴+۷۳۴

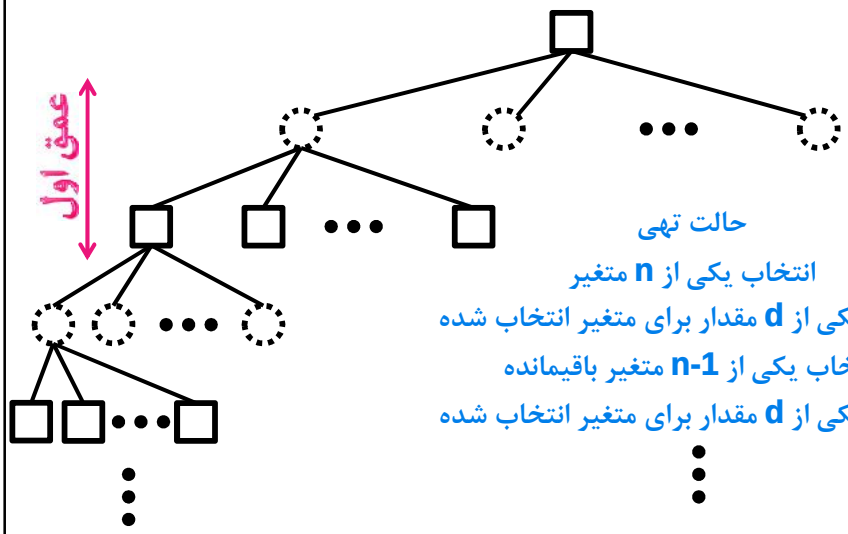
فرمول بندی مسائل CSP



فرمول بندی حالت افزایشی

- فرمول بندی حالت افزایشی (Incremental)
  - **حالت اولیه**: تهی {} (هیچ متغیری مقداری ندارد)
  - **تابع جانشین**: مقداردهی یک متغیر فاقد مقدار، به شرطی که این مقداردهی محدودیت ها را نقض نکند
  - **آزمون هدف**: آیا همه متغیرها مقداردهی شده؟ (آیا انتساب کامل است؟)
  - **هزینه مسیر**: هزینه مقداردهی هر متغیر برابر با یک!!!
- حل مسئله:
  - روش های جستجوی آگاهانه یا ناآگاهانه

فرمول بندی حالت افزایشی



حالت تهی

انتخاب یکی از **n** متغیر

انتخاب یکی از **d** مقدار برای متغیر انتخاب شده

انتخاب یکی از **n-1** متغیر باقیمانده

انتخاب یکی از **d** مقدار برای متغیر انتخاب شده

فرمول بندی مسائل CSP

- در فرمول بندی حالت افزایشی:
  - چون عمق این مسائل همواره ثابت و برابر **n** (تعداد متغیرها) است، بهترین روش از جستجوهای ناآگاهانه، روش DFS می باشد
  - **فاکتور انشعاب** مسائل CSP:
    - در عمق صفر: $b_1 = dn$
    - در عمق اول: $b_2 = d(n-1)$
    - ...
    - در عمق L: $b_L = d(n-L)$
  - تعداد برگ های درخت: d^n **n!**
  - تعداد جواب های غیریکسان: d^n
  - **خاصیت جابجایی** (Commutativity)
    - ترتیب مقداردهی متغیرها مهم نیست!

فرمول بندی حالت کامل

- فرمول بندی حالت کامل (Complete)
  - **فضای حالت**: هر انتساب کامل، یک حالت است
  - **تابع برازش (سودمندی)**: منفی مجموع تعداد محدودیت های نقض شده (در بهترین حالت تابع برازش صفر است)

- حل مسئله:
  - جستجوهای محلی (بهینه سازی)

فرمول بندی حالت کامل

- مقداردهی تصادفی تمام متغیرها
- انتخاب یکی از n متغیر و تغییر مقدار آن
- انتخاب یکی از n متغیر و تغییر مقدار آن
- انتخاب یکی از n متغیر و تغییر مقدار آن

گراف محدودیت (Constraint Graph)

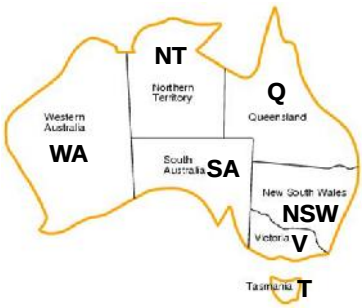
- با **گراف محدودیت** می توان هر مسئله CSP مدل سازی نمود!!!!!
  - مدل افزایشی (Incremental)
- در گراف محدودیت، هر گره **معادل** یک **متغیر** و هر **یال** معادل یک **محدودیت دوگانه** بین دو متغیر است.
- حل مسئله CSP یعنی جستجو در گراف محدودیت
- با گراف محدودیت چگونه می توان مسائل CSP با محدودیت یگانه و چندگانه را حل نمود؟

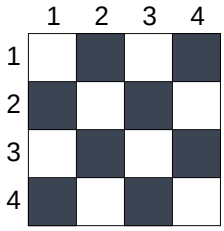
گراف محدودیت (Constraint Graph)

- **مزایای استفاده از گراف محدودیت**
- بازنمایی استاندارد (standard representation)
- هدف عمومی (generic goal)
- تابع جانشین عمومی (generic successor)
- تابع اکتشاف عمومی (generic heuristic)

کوییز

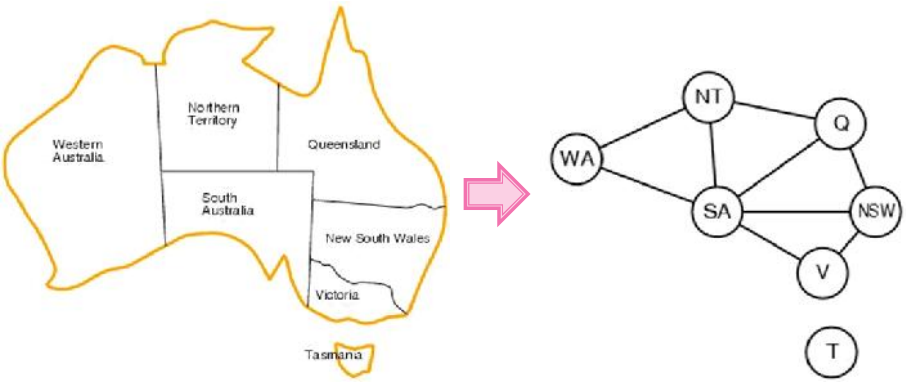
- گراف محدودیت مسائل زیر را رسم کنید:
  - رنگ آمیزی نقشه
  - رمزنگاری ریاضی (بدون رقم نقلی)
  - مسئله ۴ وزیر





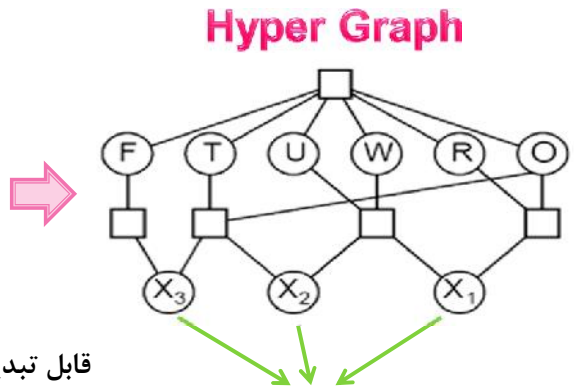

```
  ONE
+ONE
-----
  TWO
```


گراف محدودیت رنگ آمیزی نقشه



گراف محدودیت رمزنگاری ریاضی

TWO
+TWO
FOUR



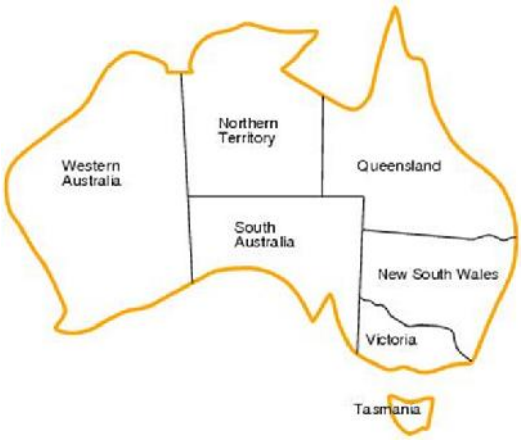
قابل تبدیل به یک گراف محدودیت
ساده با پیچیدگی بیشتر

رقم نقلی

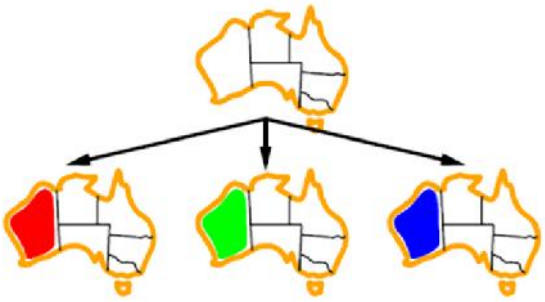
جستجوی عقبگرد (Backtracking)

- جستجوی عقبگرد (BT) همان جستجوی اول عمق (DFS) است.
- روش عملکرد:
  - در هر مرحله (عمق)، یکی از متغیرهای فاقد مقدار را مقداردهی می کند. اگر شرایطی برقرار شود که با توجه به محدودیت ها، نتوان متغیری را مقداردهی کرد، عقبگرد می کند
  - در هر عمق فقط یک متغیر مقداردهی می شود
    - عمق درخت: n
    - فاکتور انشعاب: d
    - تعداد گره های برگ: d^n
- یک جستجوی ناآگاهانه در مسائل بزرگ عملکرد خوبی ندارد

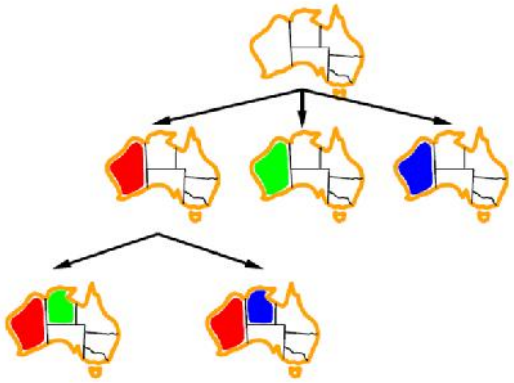
حل مسئله رنگ آمیزی نقشه با جستجوی عقبگرد



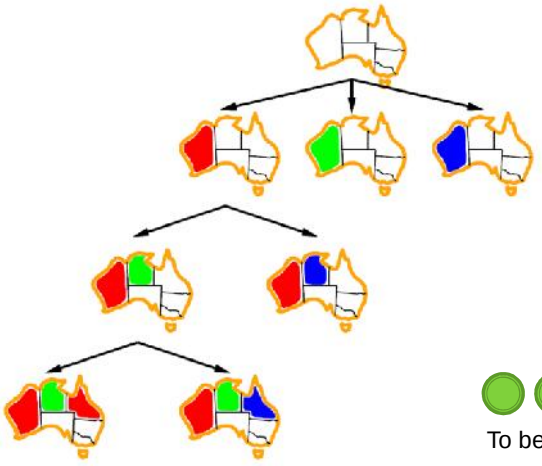
حل مسئله رنگ آمیزی نقشه با جستجوی عقبگرد



حل مسئله رنگ آمیزی نقشه با جستجوی عقبگرد



حل مسئله رنگ آمیزی نقشه با جستجوی عقبگرد



To be continued ...

بهبود جستجوی عقبگرد

- جستجوی عقبگرد یک جستجوی ناآگاهانه است.
- با تعریف توابع اکتشاف (هیوریستیک)، می توان عملکرد جستجوی عقبگرد را در حل مسائل CSP بهبود بخشید
- به دلیل خاص بودن مسائل CSP، جستجوهای آگاهانه برای این مسائل تاحدودی با روش های جستجوی آگاهانه معمولی فرق دارد و عملکرد بهتری دارد
  - توابع اکتشاف و روش های جستجو در مسائل CSP یکسان است!

توابع اکتشافی برای مسائل CSP

اکتشاف حداقل مقادیر باقیمانده

- حداقل مقادیر باقیمانده
  - MRV (Minimum Remaining Value)
- نام های دیگر:
  - محدود ترین متغیر (The Most Constrained Variable)
  - اولین شکست (First Failure)
- عملکرد:
  - در هر مرحله، از میان متغیرهای فاقد مقدار، متغیری انتخاب شود که دامنه مقادیر مجاز آن کمترین تعداد اعضا را داشته باشد
  - کاهش فاکتور انشعاب
  - در این حالت، مسیرهایی که با شکست مواجه می شوند، زودتر آشکار می شوند

اکتشاف حداقل مقادیر باقیمانده

اکتشاف درجه ای

- اکتشاف درجه ای
  - Degree Heuristic
- نام دیگر:
  - محدود کننده ترین متغیر (The Most Constraining Variable)
- عملکرد:
  - متغیری را انتخاب می کند که بیشتری محدودیت را برای سایر متغیرها ایجاد می کند
  - کاهش فاکتور انشعاب
  - در این حالت، مسیرهایی که با شکست مواجه می شوند، زودتر آشکار می شوند

اکتشاف درجه ای

ترکیب اکتشاف حداقل مقادیر باقیمانده و اکتشاف درجه ای

- اولویت کدام اکتشاف بیشتر است؟

اکتشاف حداقل مقادیر باقیمانده > اکتشاف درجه ای

- ابتدا بر اساس اکتشاف MVR، متغیری را انتخاب می کنیم که تعداد مقادیر مجاز کمتری داشته باشد. اگر دو یا چند متغیر، از این نظر در شرایط یکسان بودند، از اکتشاف درجه ای استفاده می کنیم

ترکیب اکتشاف حداقل مقادیر باقیمانده و اکتشاف درجه ای

اکتشاف مقدار با کمترین محدودکنندگی

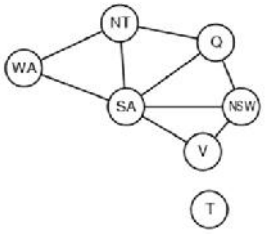
- مقدار با کمترین محدودکنندگی:
  - The Least Constraining Value
- عملکرد:
  - مقداری را برای انتساب به متغیر انتخاب می کند که کمترین محدودیت را برای سایر متغیرها ایجاد کند
  - ایجاد شرایط قابل انعطاف برای انتساب های بعدی

اکتشاف مقدار با کمترین محدودکنندگی

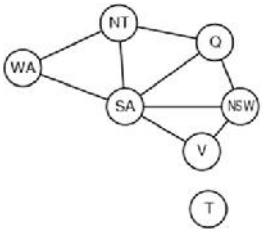
بررسی پیشرو (Forward Checking)

- **نحوه عملکرد روش بررسی پیشرو (FC):**
- وقتی یک انتساب به متغیری مانند X صورت می گیرد، روش بررسی پیشرو، تمام متغیرهای فاقد مقدار مثل Y که در گراف محدودیت به متغیر X متصل هستند (یعنی محدودیتی بین X و Y برقرار است)، را مورد بررسی قرار می دهد و مقداری از دامنه Y که با مقدار انتخاب شده برای X سازگاری ندارد، از دامنه Y حذف می کند.
- اگر متغیری باشد که دامنه آن تهی شود، این انتساب ناسازگار است.

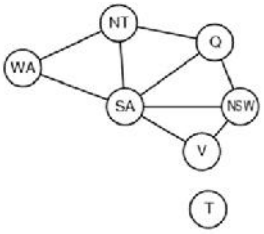
بررسی پیشرو در رنگ آمیزی نقشه



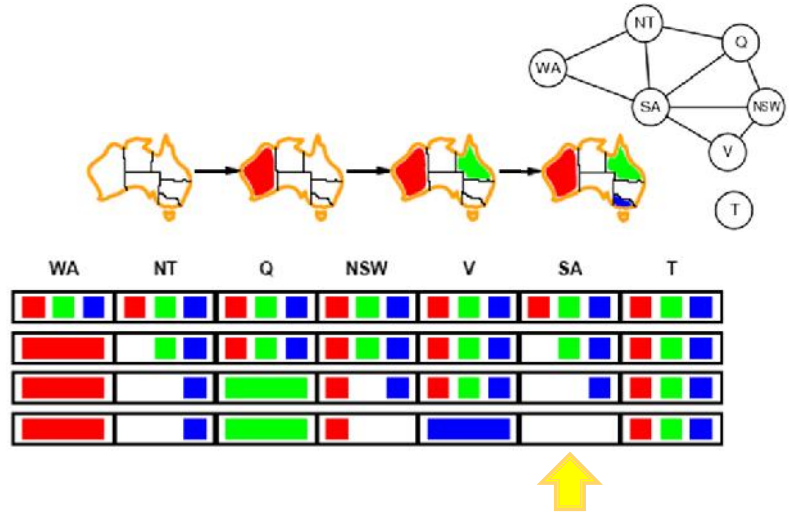
بررسی پیشرو در رنگ آمیزی نقشه



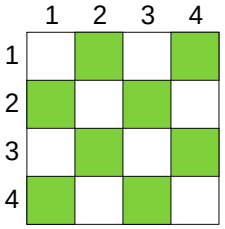
بررسی پیشرو در رنگ آمیزی نقشه

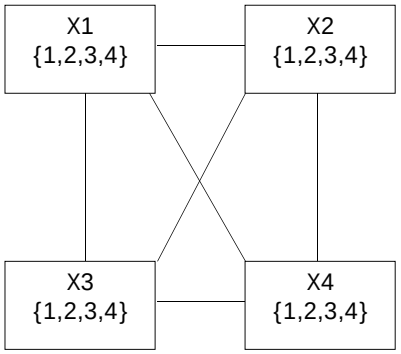


بررسی پیشرو در رنگ آمیزی نقشه

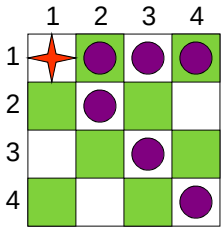


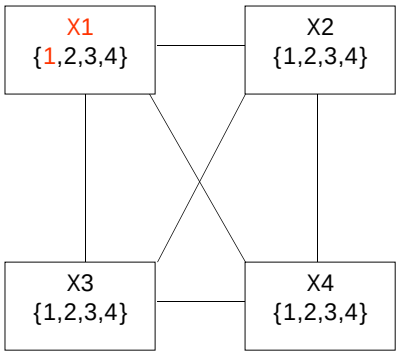
بررسی پیشرو در مسئله ۴ وزیر



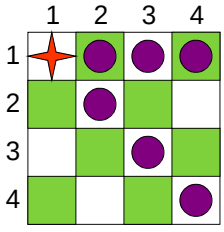


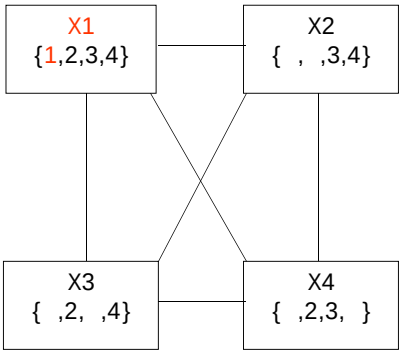
بررسی پیشرو در مسئله ۴ وزیر



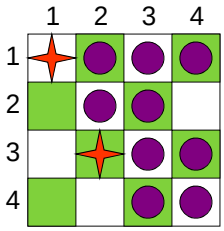


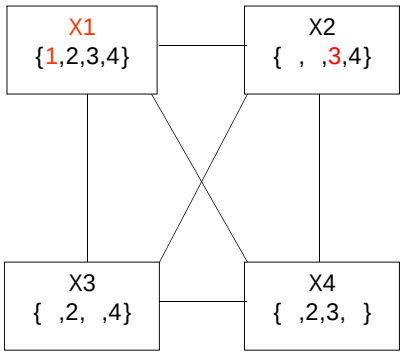
بررسی پیشرو در مسئله ۴ وزیر



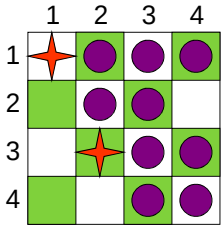


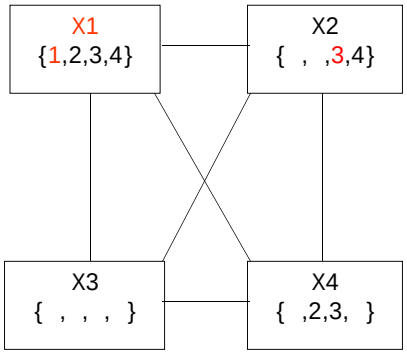
بررسی پیشرو در مسئله ۴ وزیر



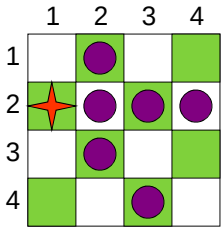


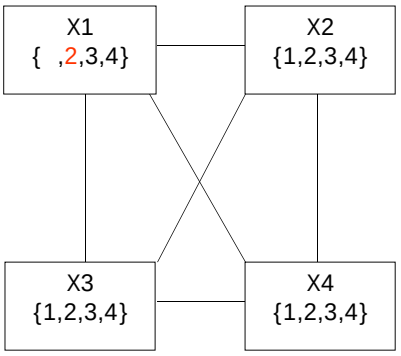
بررسی پیشرو در مسئله ۴ وزیر



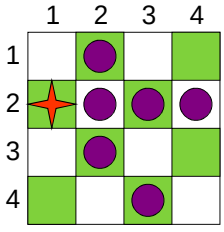


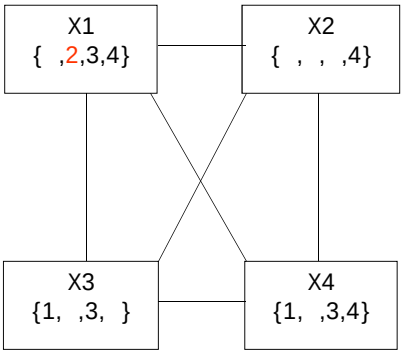
بررسی پیشرو در مسئله ۴ وزیر



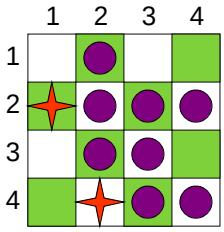


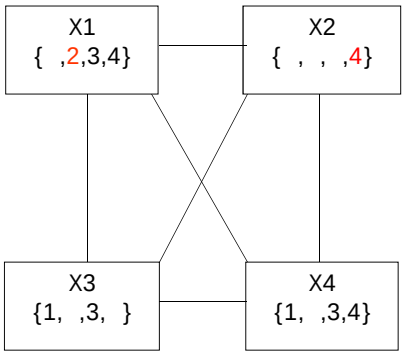
بررسی پیشرو در مسئله ۴ وزیر



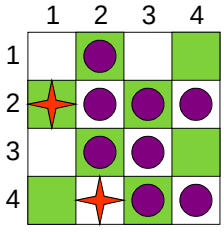


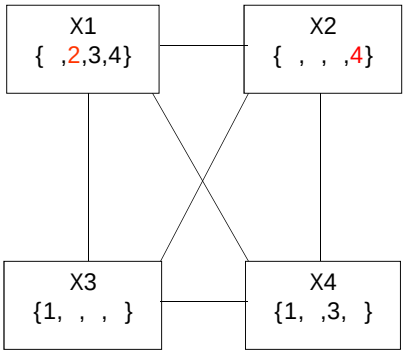
بررسی پیشرو در مسئله ۴ وزیر



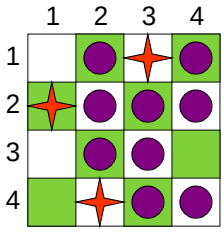


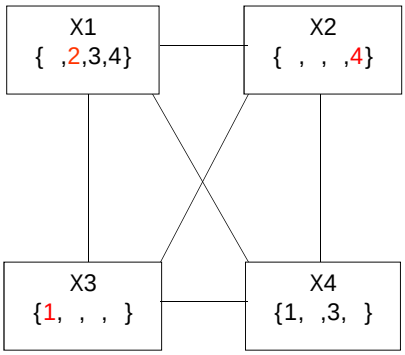
بررسی پیشرو در مسئله ۴ وزیر



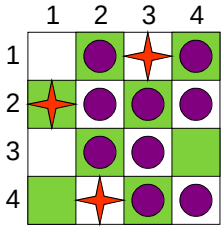


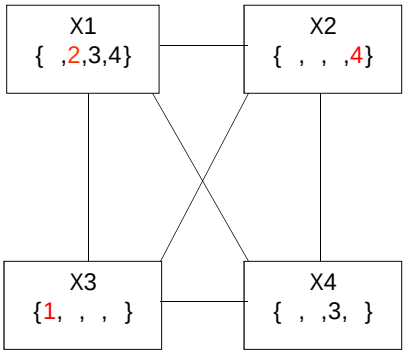
بررسی پیشرو در مسئله ۴ وزیر



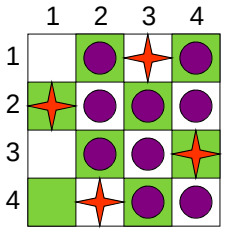


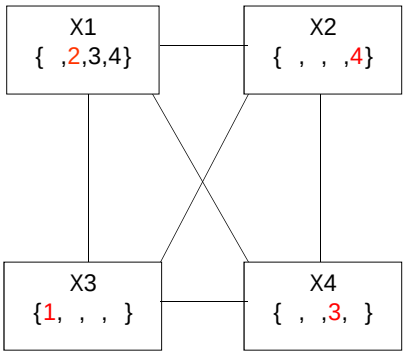
بررسی پیشرو در مسئله ۴ وزیر





بررسی پیشرو در مسئله ۴ وزیر



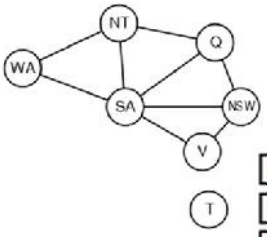


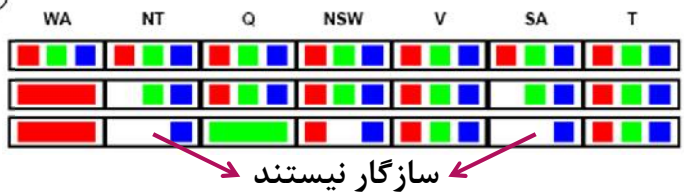
ترکیب جستجوی عقبگرد و بررسی پیشرو

- برای عملکرد بهتر، می توان جستجوی عقبگرد، همراه با توابع اکتشافی را با بررسی پیشرو ترکیب نمود.
- کوییز:
- روش ترکیب جستجوی عقبگرد (همراه با توابع اکتشاف) و بررسی پیشرو را برای مسئله رنگ آمیزی نقشه استفاده کنید.
  - اکتشاف ها:
    - The Most Constrained Variable
    - The Most Constraining Variable
    - The Least Constraining Value

پخش محدودیت

- پخش محدودیت = Constraint Propagation
  - پخش الزام محدودیت های یک متغیر به متغیرهای دیگر
- بررسی پیشرو می تواند برخی از ناسازگاری ها را تشخیص دهد، اما تشخیص همه ناسازگاری ها با استفاده از روش بررسی پیشرو ممکن نیست.





سازگار نیستند

سازگاری یال (Arc Consistency)

- گوییم متغیر X با متغیر Y سازگاری یال دارد، اگر و تنها اگر:
  - به ازای هر یک از مقادیر مجاز متغیر X، حداقل یک مقدار مجاز برای Y موجود باشد
  - اگر X با Y سازگار باشد، با نماد X→Y نشان می دهیم
  - اگر X→Y، آنگاه لزوما نمی توان نتیجه گرفت که Y→X
    - در روش سازگاری یال، گراف محدودیت به صورت گراف جهت دار فرض می شود
- در روش سازگاری یال (AC)، مقادیری از متغیرها که باعث وجود عدم سازگاری بین متغیرها می شود، حذف می شود.
  - روش AC، از روش بررسی پیشرو (FC) قدرتمندتر عمل می کند.

سازگاری یال (Arc Consistency)

- الگوریتم سازگاری یال:
  - ۱- ابتدا سازگاری تمام ترکیب های دوتایی از متغیرها را بررسی کنید. مقادیری از متغیرها که باعث وجود عدم سازگاری بین متغیرها شود، حذف می گردد.
  - ۲- استفاده از روش جستجوی عقبگرد (و توابع اکتشاف) برای جستجوی گراف
    - ۲-۱- در هر مرحله از جستجو، متغیر و مقدار مناسب آن را انتخاب کنید.
    - ۲-۲- سازگاری تمام متغیرهایی که با متغیر انتخاب شده ارتباط دارند (بین آنها محدودیت وجود دارد) بررسی و مقادیر دامنه آنها به روز شود.
- قسمت های ۱ و ۲-۲ خود یک زیرالگوریتم **تکراری** هستند

سازگاری یال (Arc Consistency)

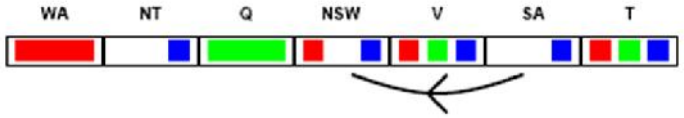
- در الگوریتم سازگاری یال (AC) گراف محدودیت به شکل گراف جهت دار فرض می شود
- در ابتدا صفی از زوج گره های (X_i,X_j) که با یکدیگر رابطه محدودیت دارند، تولید می شوند. X_i و X_j متغیرهای فاقد مقدار هستند.
- سپس هر یک از اعضای این صف به ترتیب مورد بررسی قرار گرفته و عناصری از دامنه X_i که برای آنها، حداقل یک مقدار سازگار در X_j نباشد، از دامنه X_i حذف می شوند.
- اگر دامنه متغیر X_i تغییر کرد، باید تمام متغیرهای X_k که با X_i رابطه محدودیت دارند، دوباره مورد بررسی قرار گیرند. به عبارت دیگر اگر (X_k,X_i) در صف نبود، باید (X_k,X_i) به انتهای صف افزوده شود.
- این مراحل آنقدر تکرار می شود تا صف خالی شود.

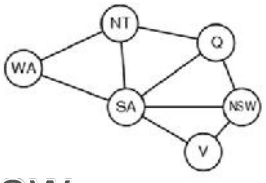
مرتبه زمانی سازگاری یال

- یک مسئله CSP دودویی (با محدودیت های دوگانه)، حداکثر $n(n-1)/2$ یال دارد
- برای هر متغیر (گره) حداکثر d مقدار مختلف وجود دارد، بنابراین، هر یال (X_k,X_i) حداکثر d بار در صف قرار می گیرد
- بررسی سازگاری هر یال (X_i,X_j) از مرتبه زمانی $O(d^2)$ است.
- در بدترین شرایط، مرتبه زمانی بررسی سازگاری یال در یک مسئله CSP دودویی (با محدودیت دوگانه) چنین خواهد شد:

$$O(n^2d^3)$$

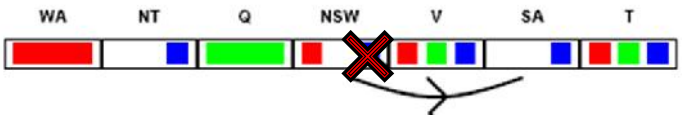
سازگاری یال در رنگ آمیزی نقشه

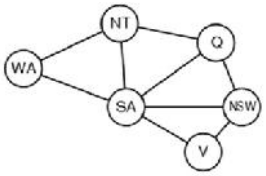




SA is arc consistent with NSW

سازگاری یال در رنگ آمیزی نقشه

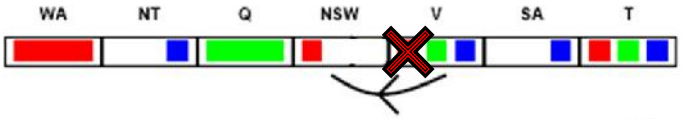


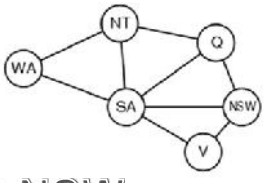


NSW is not arc consistent with SA

T

سازگاری یال در رنگ آمیزی نقشه

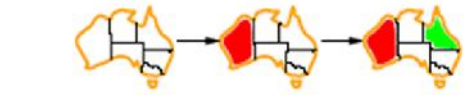


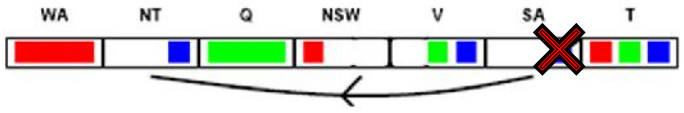


V is not arc consistent with NSW

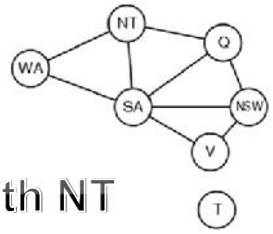
T

سازگاری یال در رنگ آمیزی نقشه





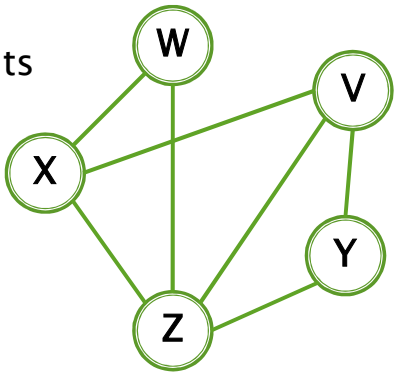
این مقداردهی به متغیرها سازگار نیست!



SA is not arc consistent with NT

مثال سازگاری یال

- Variable: V,W,X,Y,Z
- Constraint: different digits
- Domain: {1,2,3}

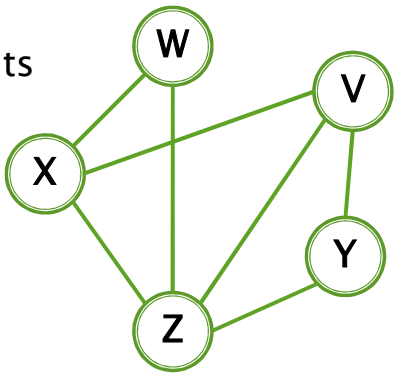


- X=1,Y=2

?

مثال سازگاری یال

- Variable: V,W,X,Y,Z
- Constraint: different digits
- Domain: {1,2,3}



- X=1,Y=2
- D_V={3}
- D_W={2,3}
- D_Z={3}
- {(Z,W),(W,Z), (V,Z),(Z,V)}

مثال سازگاری یال

- 1st Stage: (Z,W) is arc consistent
  - X=1,Y=2
  - D_V={3}
  - D_W={2,3}
  - D_Z={3}
  - {(Z,W),(W,Z),(V,Z),(Z,V)}
- 2nd Stage: (W,Z) is not arc consistent
  - X=1,Y=2 But can make it consistent
  - D_V={3}
  - D_W={2,3} -> D_W={2}
  - D_Z={3}
  - {(W,Z),(V,Z),(Z,V)} -> {(V,Z),(Z,V),(Z,W)}

مثال سازگاری یال

- 3rd Stage: (Z,W) is not arc consistent
  - X=1,Y=2 And cannot make it consistent
  - D_V={3} -> D_V={}
  - D_W={2}
  - D_Z={3}
  - {(V,Z),(Z,V),(Z,W)}

X=1 and Y=2 is not a consistent assignment!

سازگاری K (K-Consistency)

- یک مسئله CSP سازگاری K دارد، اگر برای هر k-1 متغیر و برای هر انتساب سازگار با آن متغیرها، همیشه یک مقدار سازگار برای متغیر kام وجود داشته باشد.
- K=1
  - سازگاری گره (node consistency): یعنی هر گره با خودش سازگار باشد. به عبارت دیگر مقادیر دامنه آن متغیر محدودیت های آن متغیر را نقض نکند
- K=2
  - سازگاری یال (arc consistency)
- K=3
  - سازگاری مسیر (path consistency)

سازگاری قوی K (Strong K-Consistency)

- می گوییم یک مسئله CSP قویا سازگاری K دارد اگر:
  - سازگاری K داشته باشد
  - سازگاری K-1 داشته باشد
  - سازگاری K-2 داشته باشد
  - ...
  - سازگاری ۱ داشته باشد

- اگر یک مسئله CSP سازگاری قوی n داشته باشد، می توان آن مسئله CSP را بدون عقبگرد (BT) در مرتبه زمانی O(nd) حل کرد. **چرا؟**
  - کافیست برای هر متغیر، یک مقدار سازگار انتخاب کرده و به آن متغیر منتسب شود
- بررسی سازگاری قوی n برای یک مسئله CSP واقعی، بسیار زمان بر است.

جستجوی محلی برای حل CSP

- مزایا:
  - حجم حافظه کمی لازم دارد
  - سرعت اجرای آن زیاد است (بستگی به تابع برازش دارد!)

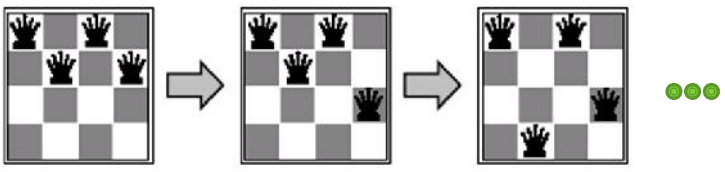
- نحوه استفاده:
  - نمایش حالت کامل مسئله (complete state)
    - در مرحله اول، تمام متغیرها مقداردهی می شوند (هرچند ممکن است مقداردهی مجاز نباشد)
  - تابع جانشین
    - تغییر مقدار یک متغیر

جستجوی محلی برای حل CSP

- نحوه انتخاب متغیر در هر مرحله از جستجوی محلی
  - انتخاب تصادفی یکی از متغیرهایی که محدودیت های موجود را نقض می کند
- نحوه انتخاب مقدار برای متغیر
  - استفاده از تابع اکتشاف کمترین برخورد (min conflicts)
  - در اکتشاف کمترین برخورد، مقداری انتخاب می شود که کمترین تعداد محدودیت را نقض کند
- استفاده از اکتشاف کمترین برخورد بسیار کارمد است و می تواند سخت ترین مسائل CSP را در مدت محدود پاسخ دهد
  - مسئله ۱ میلیون وزیر حداکثر با چندصد مرحله حل خواهد شد

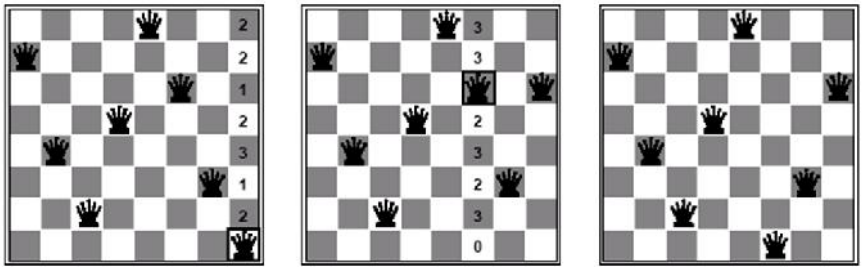
مثال: مسئله ۴ وزیر

- استفاده از الگوریتم تپه نوردی و اکتشاف کمترین برخورد



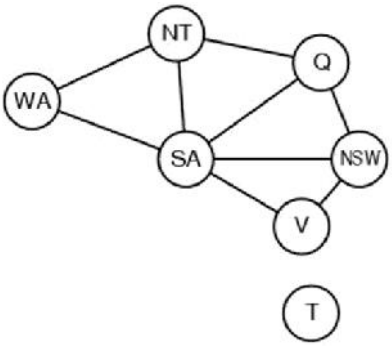
مثال: مسئله ۸ وزیر

- استفاده از الگوریتم تپه نوردی و اکتشاف کمترین برخورد
- حل مسئله در دو مرحله



نقش ساختار مسئله CSP در حل آن

- در مسئله رنگ آمیزی نقشه استرالیا، رنگ آمیزی Tasmania مستقل از رنگ آمیزی سایر قسمت های نقشه است



نقش ساختار مسئله CSP در حل آن

- اگر گراف محدودیت یک مسئله CSP همبند نباشد و شامل m جز منفک باشد، می توان این مسئله را به m مسئله CSP مستقل (independent) تفکیک و هر یک را جداگانه حل کرد.

- در حالت کلی، اگر یک مسئله CSP با n متغیر، قابل تفکیک به چند جز مستقل از هم باشد که هر جز c متغیر داشته باشد:
  - مرتبه زمانی حل مسئله بدون در نظر گرفتن ساختار منفک مسئله: $O(d^n)$
  - مرتبه زمانی حل مسئله با در نظر گرفتن ساختار منفک مسئله: $O(\frac{n}{c} d^c)$

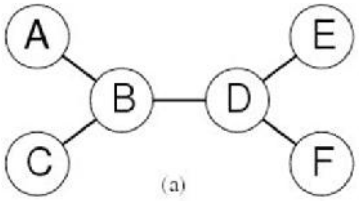
نقش ساختار مسئله CSP در حل آن

- به عنوان مثال اگر داشته باشیم:
  - n=80
  - d=2
  - c=20

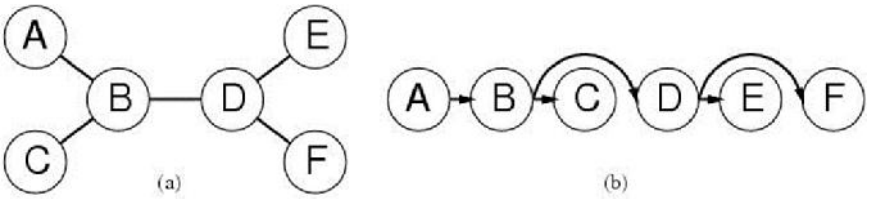
- حل مسئله بدون در نظر گرفتن ساختار منفک آن:
  2^{80} = 4 billion years at 1 million nodes/sec.
- حل مسئله بدون در نظر گرفتن ساختار منفک آن:
  $4 * 2^{20}$= 0.4 second at 1 million nodes/sec

حل مسائل CSP با ساختار درختی

- **قضیه**: اگر گراف محدودیت یک مسئله CSP درخت باشد (بدون سیکل باشد)، می توان این مسئله CSP را به جای $O(d^n)$ در $O(nd^2)$ حل کرد.



حل مسائل CSP با ساختار درختی



- الگوریتم حل مسئله CSP با ساختار درخت

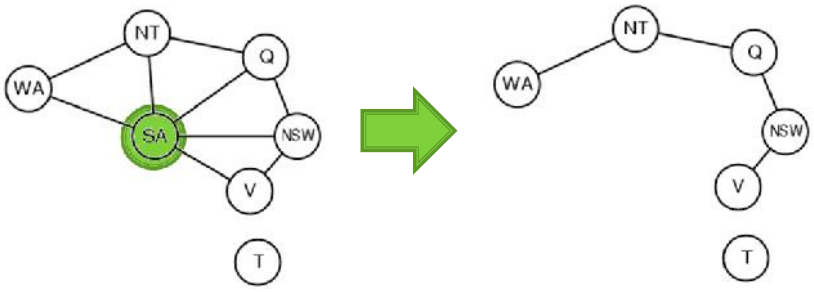
Ÿ) انتخاب یک گره به عنوان ریشه درخت و مرتب کردن گره ها با پیمایش VLR

φ) برای j از n تا ۲، سازگاری یال (X_i,X_j) را اعمال کن، به طوری که X_i والد X_j است

ϖ) برای j از ۱ تا n، متغیر X_j را مقداردهی کن، به طوری که مقدار آن با مقدار متغیر X_i که والد X_j است، سازگار باشد

رهنمودهایی برای حل مسائل CSP

- با انتخاب مناسب یک یا چند متغیر و مقداردهی آنها، می توان یک مسئله CSP با ساختار گراف را به مسئله CSP با ساختار درخت تبدیل نمود



رهنمودهایی برای حل مسائل CSP

- با انتخاب مناسب یک یا چند متغیر و مقداردهی آنها، می توان یک مسئله CSP را به چند مسئله CSP مستقل تبدیل نمود