مسائل ارضای محدودیت

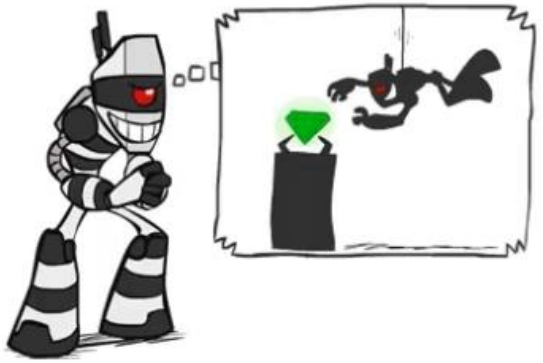
سید ناصر رضوی n.razavi@tabrizu.ac.ir
۱۳۹۷

فهرست مطالب

- تعریف مسائل ارضای محدودیت
- روش‌های جستجوی استاندارد
- جستجوی عقب‌گرد
- بررسی رو به جلو
- استفاده از هیوریستیک‌ها
- استفاده از جستجوی محلی

جستجو برای چیست؟

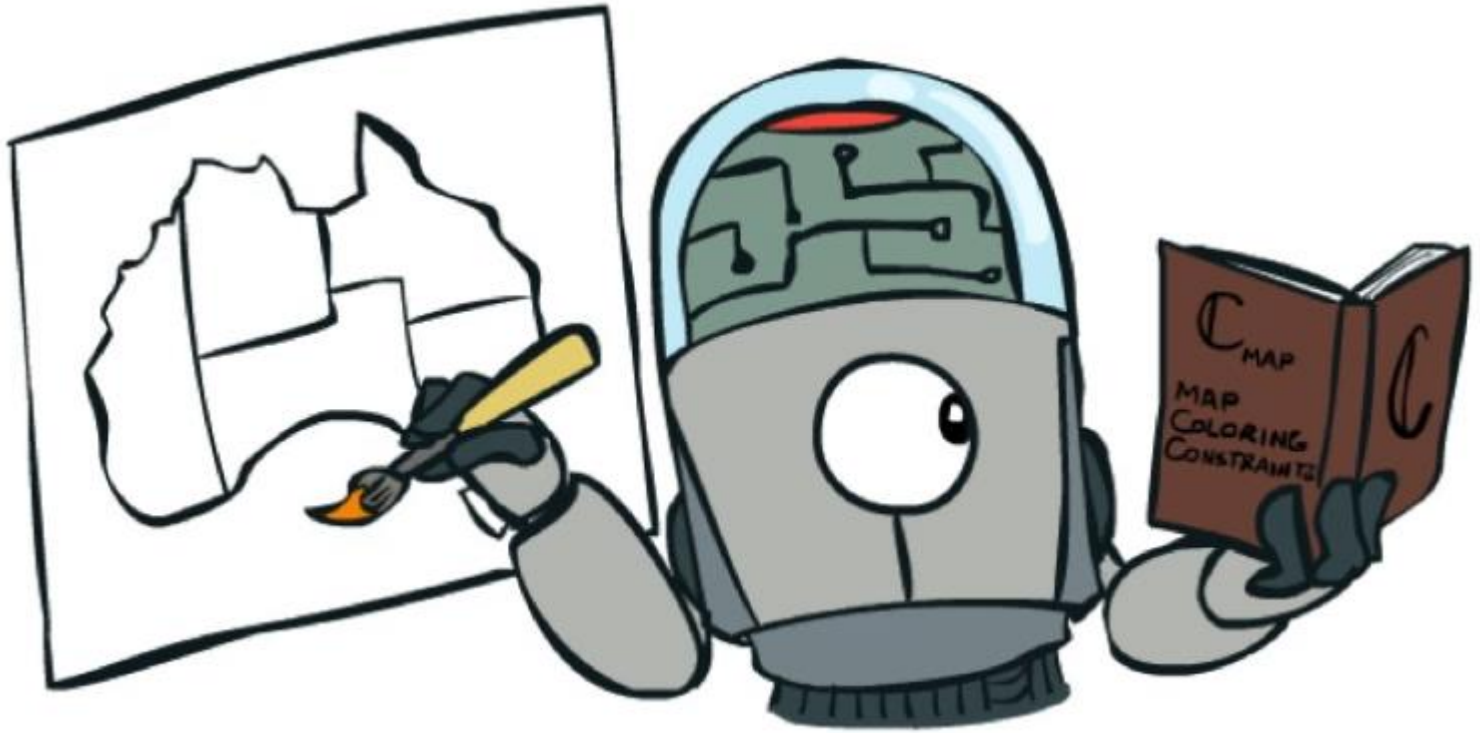
- فرضیات درباره محیط. تک عاملی، قطعی، کاملاً قابل مشاهده، گسسته



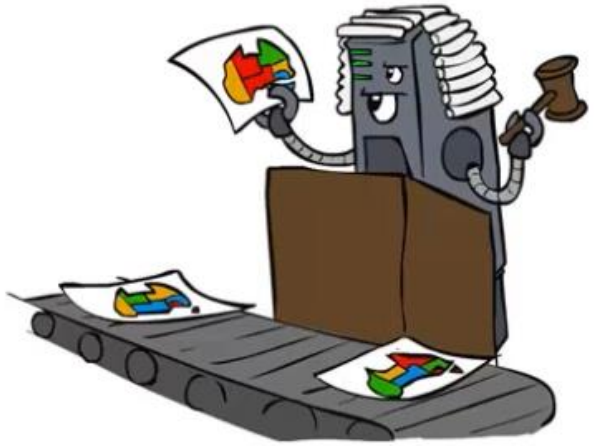
- هدف جستجو. برنامه‌ریزی [یافتن دنباله‌ای از عملیات]
  - مسیر یافته شده تا هدف مهمترین موضوع است.
  - مسیرهای مختلف دارای هزینه‌های متفاوتی هستند.
  - هیوریستیک‌ها مسیر جستجو را به سمت هدف هدایت می‌کنند.

- هدف جستجو. شناسایی [انتساب مقدار به متغیرها]
  - خود هدف مهم است و نه مسیر رسیدن به هدف.
  - همه مسیرها دارای عمق یکسانی هستند.
  - جستجوی ارضای محدودیت مختص مسائل شناسایی است.

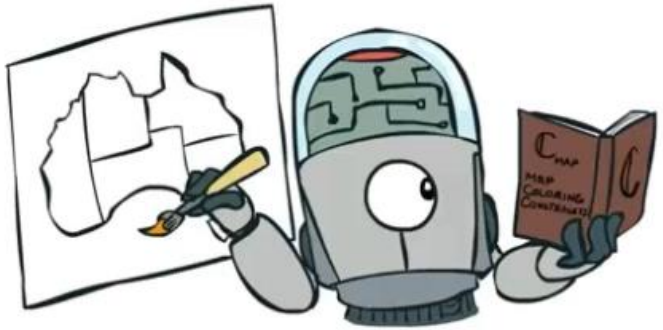
مسائل ارضای محدودیت



مسائل ارضای محدودیت



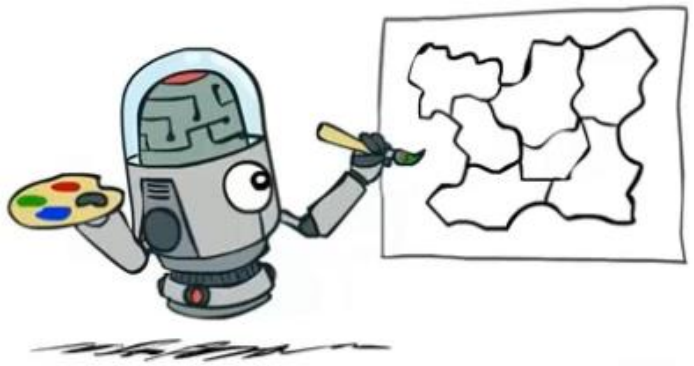
- مسائل جستجوی استاندارد.
  - حالت‌ها به صورت «جعبه سیاه» هستند: ساختمان داده‌های دلخواه
  - تابع آزمون هدف، هر تابعی می‌تواند باشد.
  - تابع جانشین نیز هر تابعی می‌تواند باشد.



- مسائل ارضای محدودیت.
  - یک زیرمجموعه خاص از مسائل جستجو
  - حالت به وسیله مجموعه‌ای از متغیرهای X_i تعریف می‌شود که مقدار خود را از یک دامنه D_i می‌گیرند.
  - آزمون هدف یک مجموعه از محدودیت‌ها است که یک ترکیب مجاز از مقادیر را برای متغیرها مشخص می‌کنند.

مثال: رنگ‌آمیزی نقشه و N-وزیر

رنگ‌آمیزی نقشه



- متغیرها

$X = \{WA, NT, Q, NSW, V, SA, T\}$

- دامنه‌ها

$D = \{Red, Green, Blue\}$

- محدودیت‌ها: نواحی مجاور باید رنگ‌های متفاوتی داشته باشند.
  - ضمنی:

$WA \neq NT$

  - صریح:

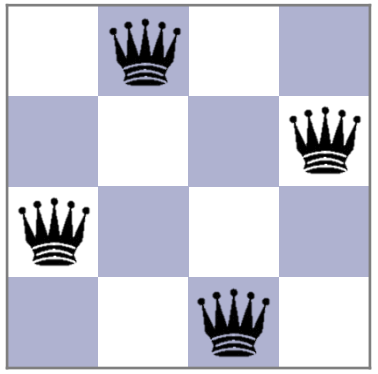
$(WA, NT) \in \{(Red, Green), (Red, Blue), \ldots\}$

رنگ‌آمیزی نقشه

- راه‌حل. یک انتساب کامل و سازگار
  - انتساب مقادیر به متغیرها به گونه‌ای که تمامی محدودیت‌ها برآورده شوند.

{WA=Red, NT=Green, Q=Red, NSW=Green, V=Red, SA=Blue, T=Green}

مسئله N-وزیر: روش اول



- متغیرها

$$X_{ij}$$

- دامنه‌ها

$$\{0,1\}$$

- محدودیت‌ها.

$$\sum_{i,j} X_{ij} = N$$

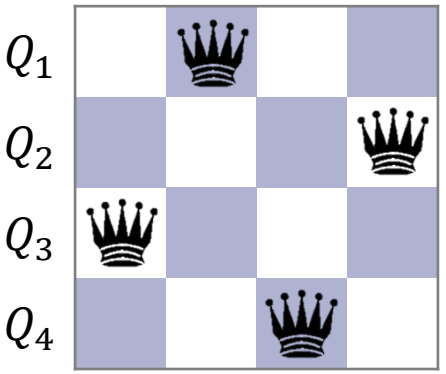
$$\forall i,j,k\ (X_{ij}, X_{ik}) \in \{(0,0),(0,1),(1,0)\}$$

$$\forall i,j,k\ (X_{ij}, X_{kj}) \in \{(0,0),(0,1),(1,0)\}$$

$$\forall i,j,k\ (X_{ij}, X_{i+k,j+k}) \in \{(0,0),(0,1),(1,0)\}$$

$$\forall i,j,k\ (X_{ij}, X_{i+k,j-k}) \in \{(0,0),(0,1),(1,0)\}$$

مسئله N-وزیر: روش دوم



- متغیرها

$$\{Q_1, Q_2, \dots, Q_N\}$$

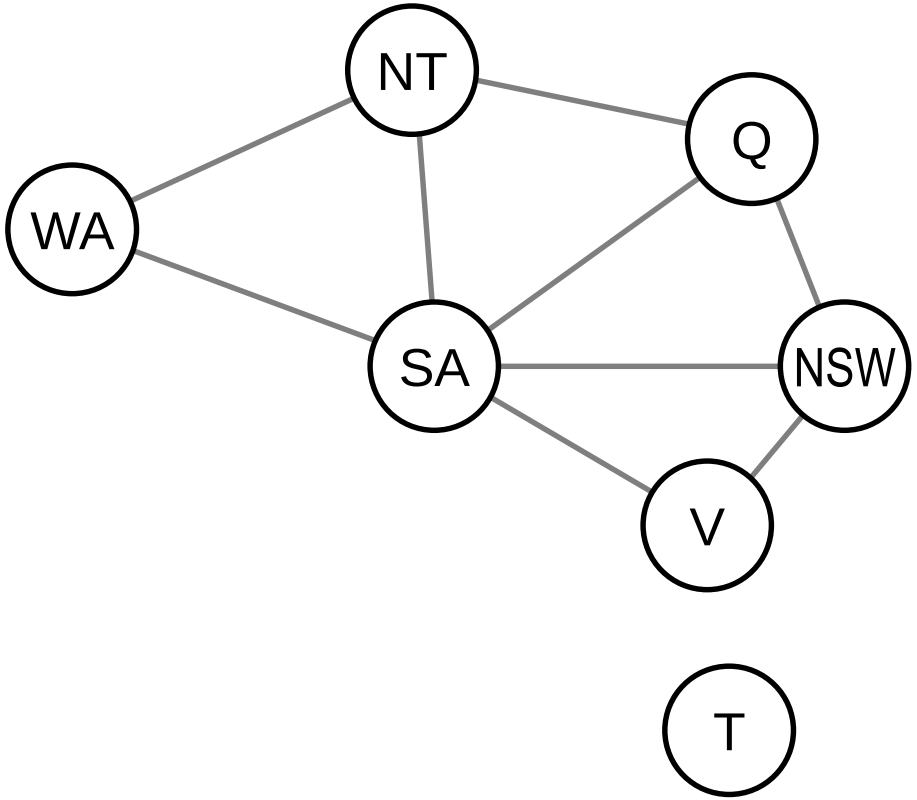
- دامنه‌ها

$$\{1,2,3, \dots, N\}$$

- محدودیت‌ها.
  - ضمنی $\forall i, j \text{ non-threatening}(Q_i, Q_j)$
  - صریح $(Q_1, Q_2) \in \{(1,3), (1,4), \dots\}$

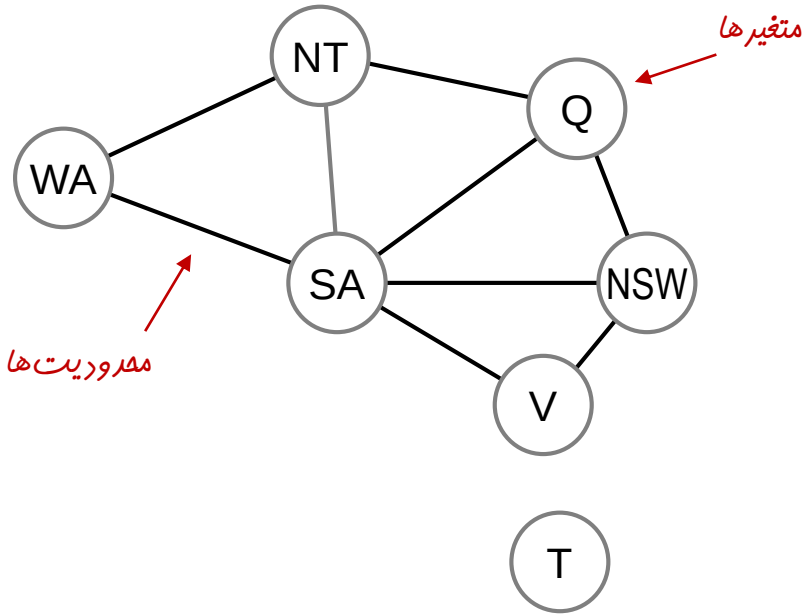
کدام یک از این در روش برای بیان مسئله N-وزیر به صورت یک مسئله ارضای محدودیت مناسب‌تر است؟

گراف‌های محدودیت



گراف‌های محدودیت

- CSP دودویی. هر محدودیت حداکثر بر روی دو متغیر تعریف می‌شود.



- گراف محدودیت دودویی.
  - گره‌ها بیانگر متغیرها
  - یال‌ها بیانگر محدودیت‌ها

- الگوریتم‌های همه منظوره CSP از ساختار گراف محدودیت برای افزایش سرعت جستجو استفاده می‌کنند -- مثلاً تاسمانیا یک زیرمسئله مستقل است!

مزیت بیان مسائل به صورت مسائل ارضای محدودیت

- تابع جانشین و تابع آزمون هدف کلی.
  - حالت‌ها: یک مجموعه از متغیرها به همراه مقدار انتساب یافته به هر کدام
  - به دلیل نمایش استاندارد حالت‌ها، می‌توان تابع جانشین و تابع آزمون هدف را به یک شکل کلی نوشت به گونه‌ای که این توابع در هر مسئله CSP قابل استفاده باشند.

- توابع هیوریستیک کلی و کارا.
  - امکان ایجاد توابع هیوریستیک کلی و کارا به گونه‌ای که ایجاد آنها نیاز به دانش و تخصص اضافی در مورد مسئله نداشته باشد.

مثال‌ها: ریاضیات رمزی و سودوکو

ریاضیات رمزی

- متغیرها.

$$F\ T\ U\ W\ R\ O\ \ X_1\ X_2\ X_3$$

- دامنه‌ها.

$$\{0,1,2,3,4,5,6,7,8,9\}$$

- محدودیت‌ها.

X_3	X_2	X_1	
	T	W	O
+	T	W	O
F	O	U	R

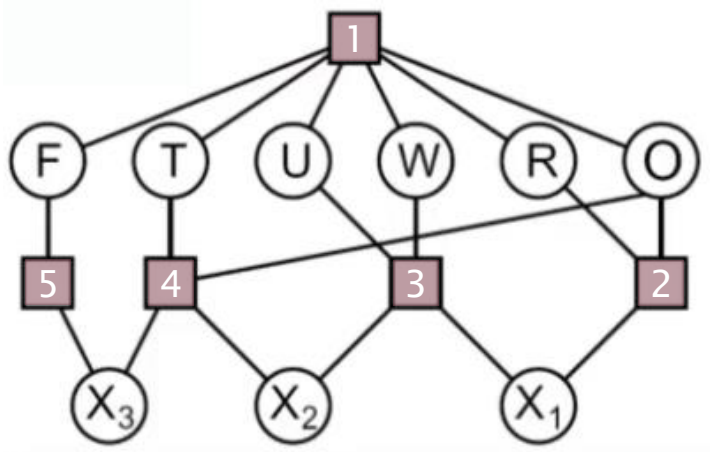
$$\text{alldiff}(F,T,U,W,R,O)$$

$$O + O = R + 10 \cdot X_1$$

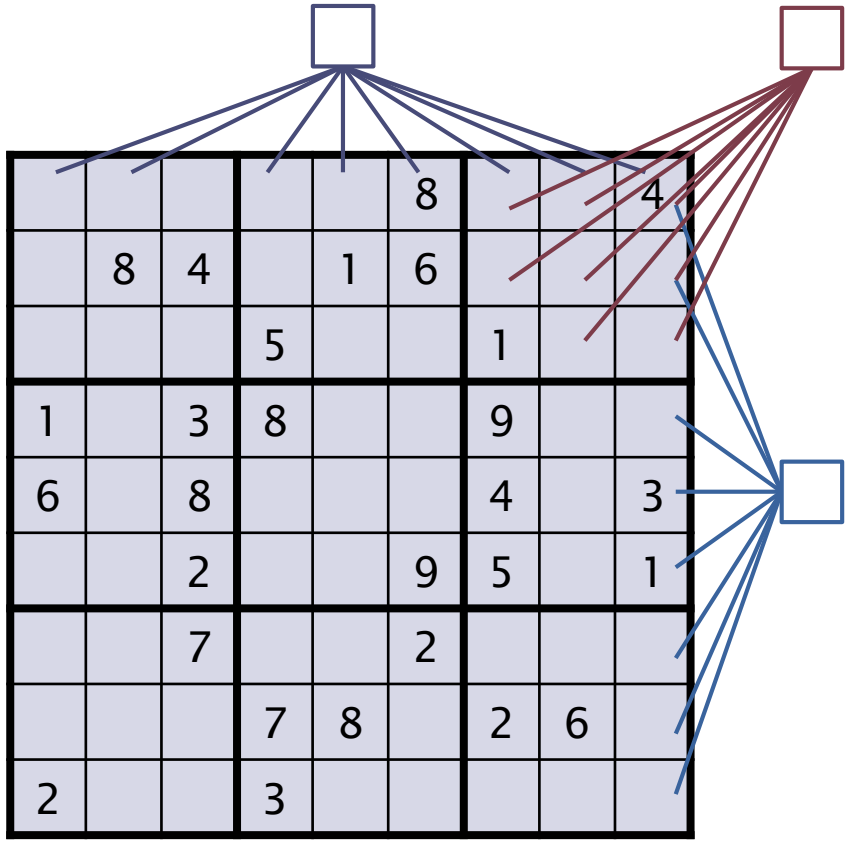
$$X_1 + W + W = U + 10 \cdot X_2$$

$$X_2 + T + T = O + 10 \cdot X_3$$

$$X_3 = F$$



سودوکو



- متغیرها.
  - خانه‌های خالی
- دامنه‌ها.
  - ارقام ۱ تا ۹
- محدودیت‌ها.
  - نُه مقدار مختلف در هر سطر
  - نُه مقدار مختلف در هر ستون
  - نُه مقدار مختلف در هر ناحیه

گونه‌های مختلف CSP

گونه‌های مختلف CSP

- متغیرهای گسسته.
  - دامنه متناهی. [مانند مقادیر بولی]
    - اندازه دامنه d به معنای $O(d^n)$ انتساب کامل است.
    - مانند CSP دودویی، رنگ‌آمیزی نقشه و n-وزیر
  - دامنه نامتناهی. [اعداد صحیح، رشته‌ها و غیره]
    - مثال: زمان شروع و پایان کارها در مسئله زمانبندی
    - محدودیت‌های خطی قابل حل، محدودیت‌های غیرخطی تصمیم ناپذیر

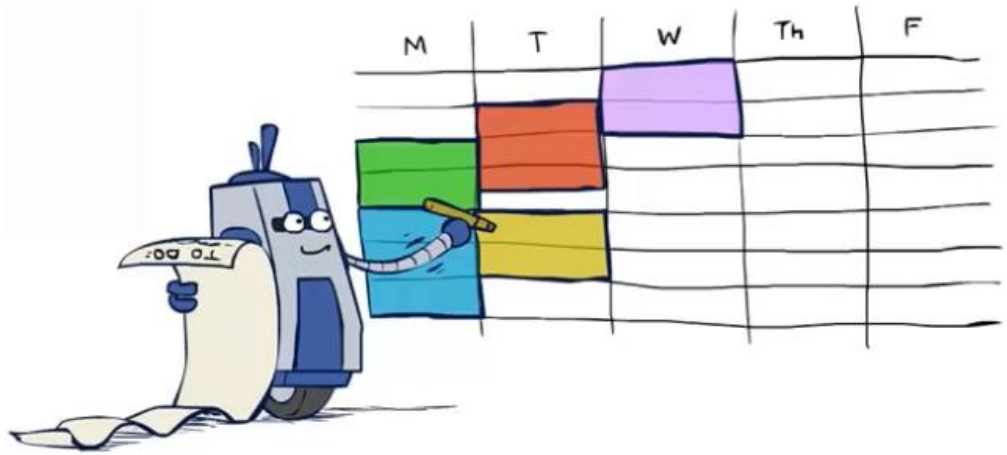
- متغیرهای پیوسته.
  - مانند زمان شروع و پایان مشاهدات برای تلسکوپ هابل
  - محدودیت‌های خطی قابل حل در زمان چندجمله‌ای

گونه‌های مختلف CSP

- گونه‌های مختلف محدودیت‌ها.
  - محدودیت‌های یکانی شامل یک متغیر
    - مانند $SA \neq green$
  - محدودیت‌های دودویی شامل یک زوج از متغیرها
    - مانند $SA \neq WA$
  - محدودیت‌های مرتبه بالاتر شامل ۳ متغیر یا بیشتر
    - مانند محدودیت‌های ستونی در ریاضیات رمزی
- محدودیت‌های نرم (اولویت‌ها).
  - مثلاً ترجیح رنگ قرمز به سبز
  - در نظر گرفتن هزینه برای انتساب مقادیر به متغیرها
  - مسائل بهینه‌سازی دارای محدودیت

چند مثال از دنیای واقعی

- مسائل انتسابی: چه کسی برای چه درسی؟
- مسائل تعیین جدول زمانبندی: کدام درس چه زمانی و در کجا ارائه می‌شود؟
- پیکره‌بندی سخت‌افزار
- زمانبندی حمل و نقل
- زمانبندی کارخانه
- طراحی مدار
- تشخیص خطا
- ...

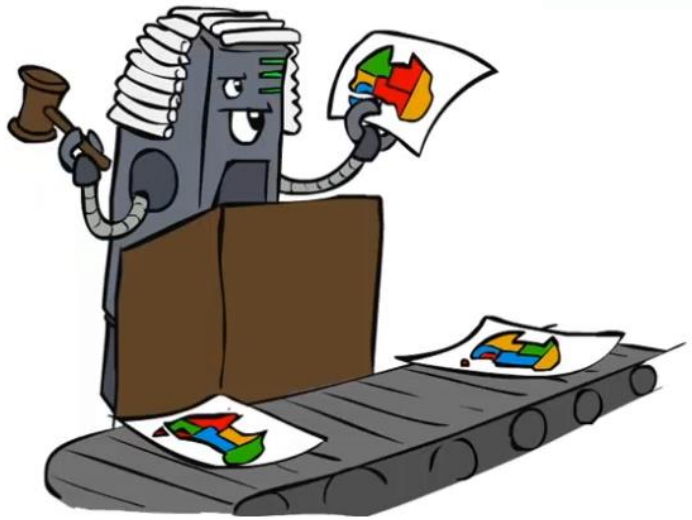


- توجه. بسیاری از مسائل دنیای واقعی شامل متغیرهایی با مقادیر حقیقی هستند. [برنامه‌ریزی خطی یا گسسته‌سازی قبل از حل]

حل مسائل ارضای محدودیت

فرموله‌سازی استاندارد جستجو

- فرموله‌سازی استاندارد جستجو برای مسائل CSP.

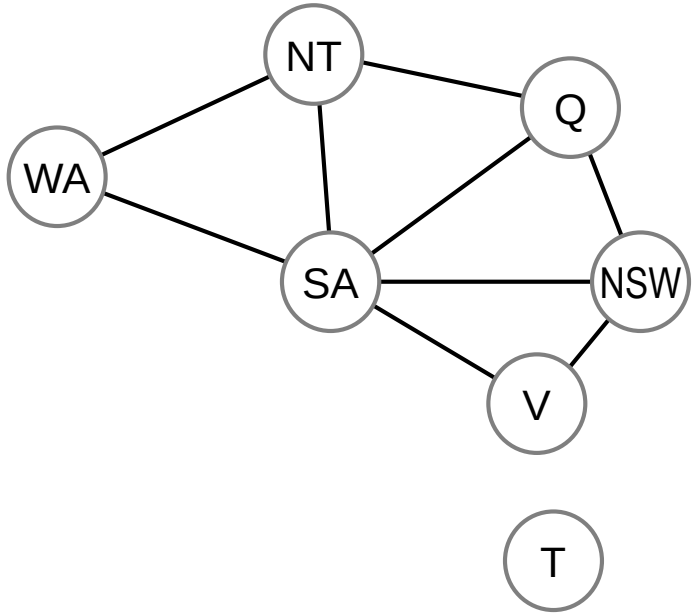


- حالت‌ها. مقادیر انتساب یافته تا کنون [انتساب جزیی]
  - حالت شروع: انتساب تهی، {}
  - تابع جانشین: انتساب مقدار به یک متغیر بدون مقدار
  - تست هدف: انتساب کنونی باید کامل و سازگار باشد.

- ابتدا با یک روش ساده و ابتدایی شروع می‌کنیم و سپس آن را بهبود می‌دهیم.

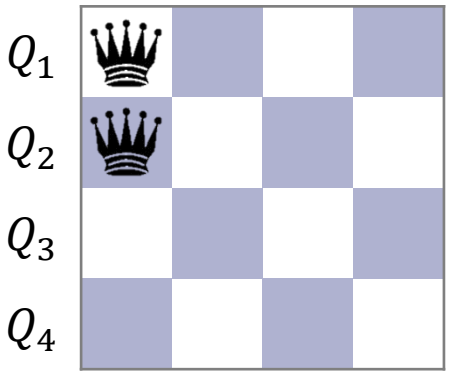
روش‌های استاندارد جستجو

- جستجوی سطحی.
  - از آنجا که گره‌های هدف در آخرین سطح درخت قرار دارند، برای یافتن راه‌حل باید تمامی درخت جستجو را تولید کنیم.



- جستجوی عمقی. یک انتخاب بهتر!
  - عمق درخت جستجو: n (تعداد متغیرها)
  - عمق کم عمق‌ترین راه‌حل: n (همه متغیرها مقدار دارند)
  - فاکتور انشعاب: $\Sigma_i|D_i|$ (در بالای درخت)

بهبود جستجوی عمقی

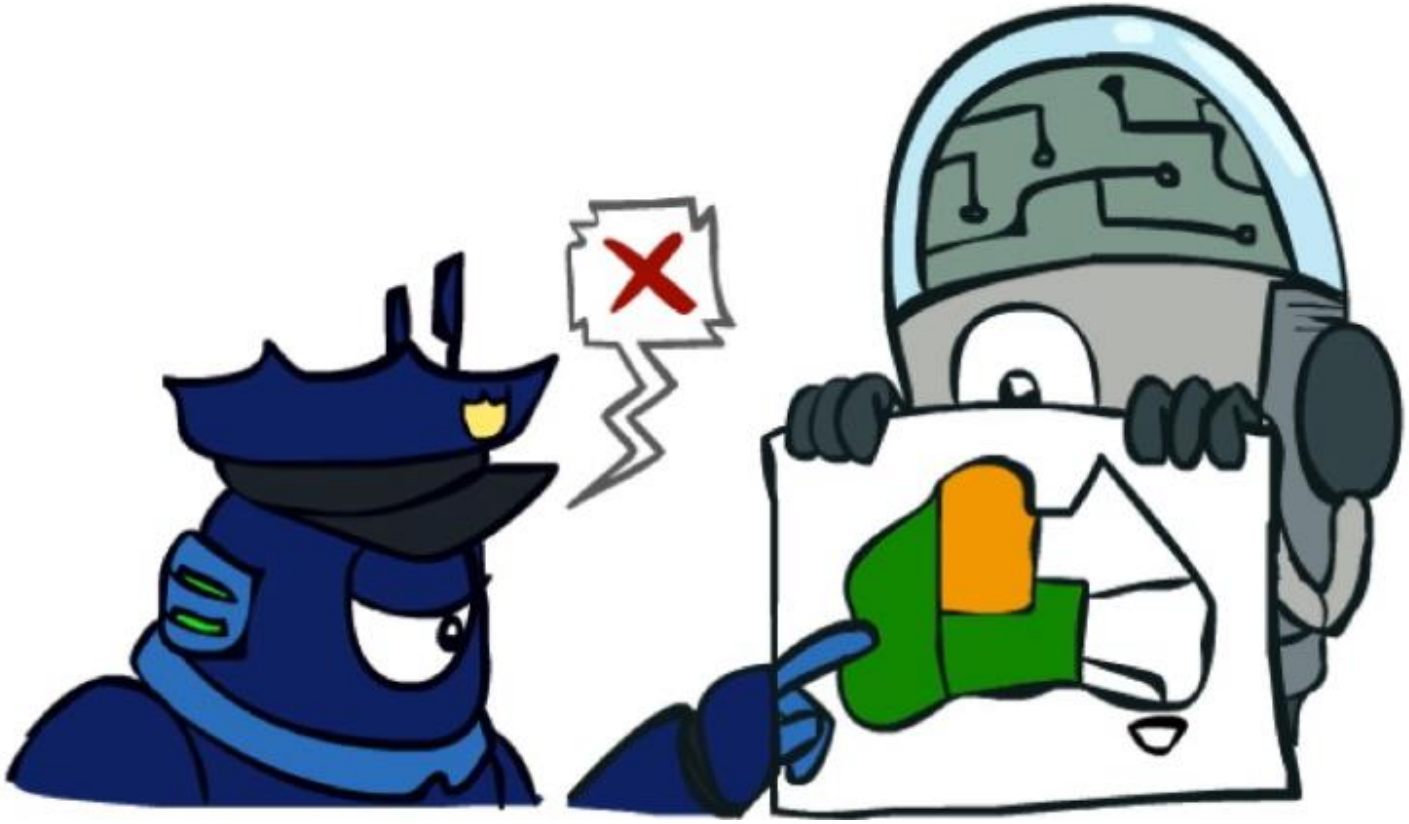


- ترتیب انتساب مقدار به متغیرها اهمیت ندارد.
  - در نتیجه بسیاری از مسیرها معادل یکدیگر هستند!
  - کاهش تعداد مسیرها از $۸^۸ \times ۸!$ به $۸^۸$ در مسئله n-وزیر.

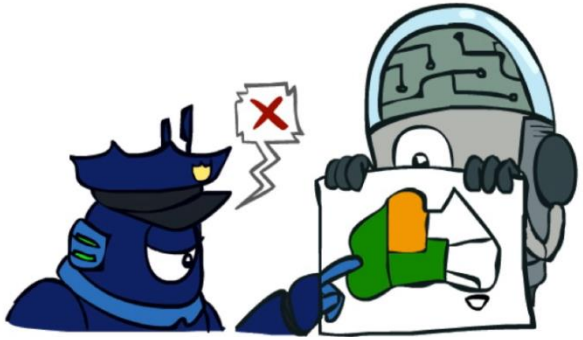
- انتساب‌های بعدی نمی‌توانند یک محدودیت نقض شده را تصحیح کنند.
  - به عنوان مثال اگر در مسئله ۸-وزیر دو وزیر اول در یک سطر قرار داده شوند، جستجوی عمقی باید تمام $۸^۶$ چیدمان باقیمانده را بررسی کند تا بفهمد این راه‌حل صحیح نیست.

- راه‌حل. استفاده از جستجوی عقب‌گرد به جای جستجوی عمقی

جستجوی عقب‌گرد



جستجوی عقب‌گرد



- جستجوی عقب‌گرد. یک الگوریتم پایه ناآگاهانه برای حل مسائل CSP
- ایده ۱. یک متغیر در هر مرحله.
  - انتساب متغیرها جابجاپذیر است، بنابراین یک ترتیب ثابت برای انتساب‌ها در نظر بگیر.
    - مثلاً در مسئله n-وزیر ترتیب قرار دادن وزیرها در صفحه مهم نیست.
    - در نتیجه در هر مرحله تنها باید یک متغیر مقداردهی شود.
- ایده ۲. در حین پیشروی محدودیت‌ها را در نظر بگیر.
  - فقط مقادیری را در نظر بگیر که با انتساب‌های قبلی سازگارند
  - نیاز به انجام مقداری محاسبه برای بررسی محدودیت‌ها
  - «آزمون هدف به صورت افزایشی»

الگوریتم جستجوی عقب‌گرد

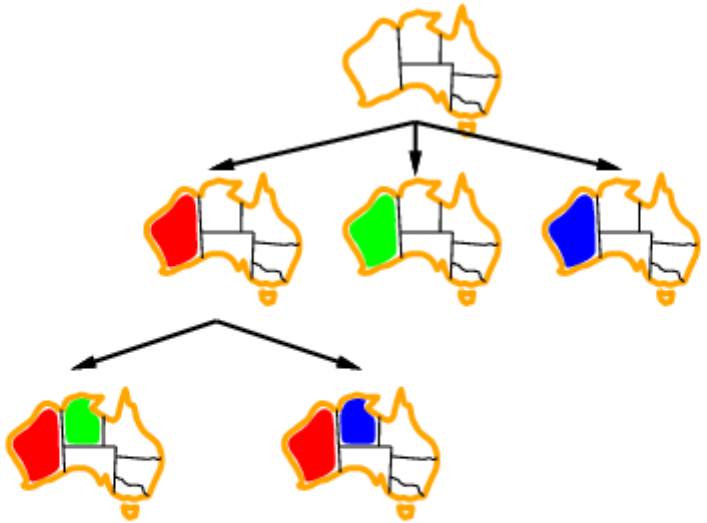
function BACKTRACKING-SEARCH(csp) **returns** solution/failure
  return RECURSIVE-BACKTRACKING({}, csp)

function RECURSIVE-BACKTRAKING($assignment$, csp) **returns** soln/failure
  if $assignment$ is complete **then return** $assignment$
  $var \leftarrow$ SELECT-UNASSIGNED-VARIABLE(VARIABLES[csp], assignment, csp)
  for each $value$ **in** ORDER-DOMAIN-VALUES(var, $assignment$, csp) **do**
    if $value$ is consistent with $assignment$ given CONSTRAINTS[csp] **then**
      add $\{var = value\}$ to $assignment$
      $result \leftarrow$ RECURSIVE-BACKTRACKING($assignment$, csp)
      if $result \neq$ failure **then return** $result$
      remove $\{var = value\}$ from $assignment$
  return failure

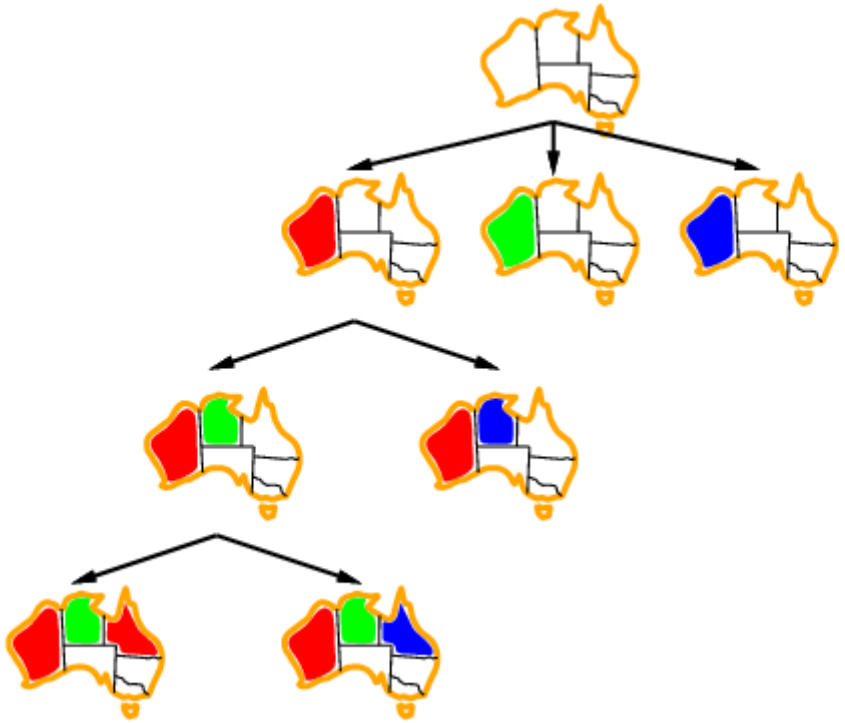
جستجوی عقبگرد

جستجوی عقبگرد

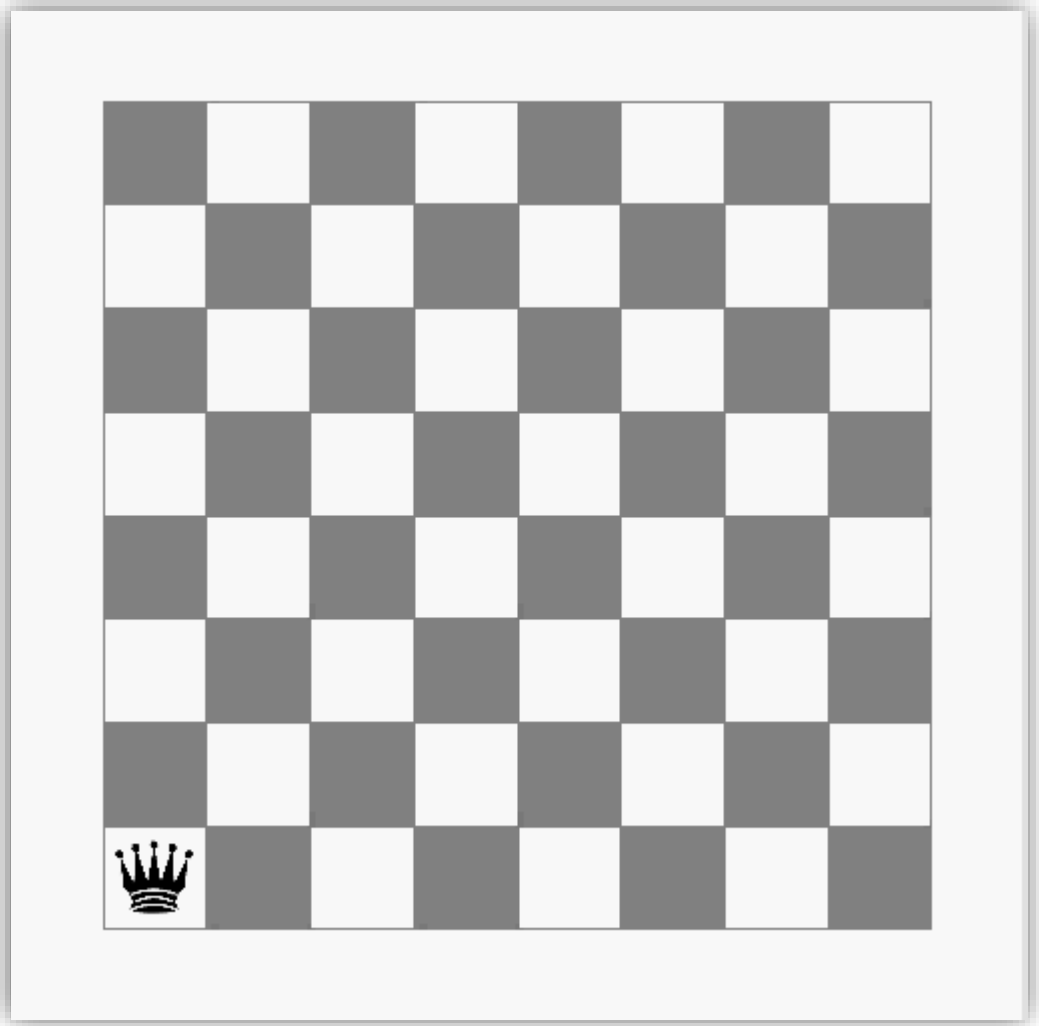
جستجوی عقبگرد



جستجوی عقبگرد



جستجوی عقب‌گرد: اجرای نمایشی



بهبود عقب‌گرد

- ایده‌های همه-منظوره باعث افزایش چشمگیری در سرعت اجرا می‌شوند.

- ترتیب‌دهی.
  - متغیر بعدی که باید مقداردهی شود کدام است؟
  - مقدار بعدی که باید امتحان شود کدام است؟

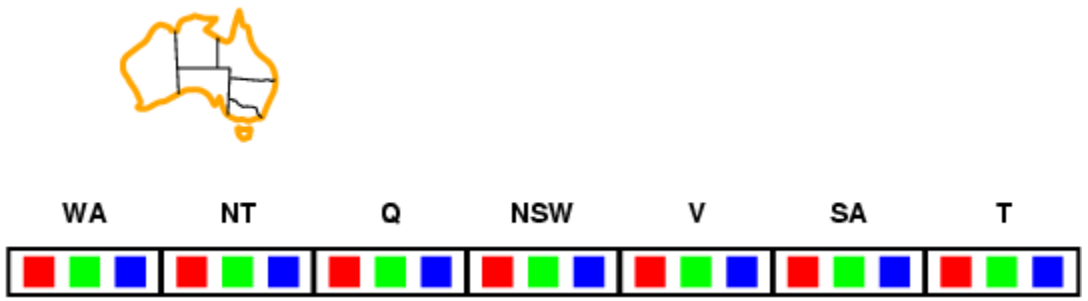
- فیلتر کردن. آیا می‌توان شکست‌های حتمی را زودتر تشخیص داد؟
  - فرض کنید در عقب‌گرد در مسئله ۸-وزیر، ۶ وزیر اول به گونه‌ای قرار گرفته‌اند که قرار دادن هشتمین وزیر را غیرممکن می‌سازند. عقب‌گرد تمام مکان‌های ممکن برای وزیر هفتم را بررسی می‌کند، اگرچه مسئله غیر قابل حل است.

- ساختار. آیا می‌توان از ساختار مسئله بهره برد؟

فیلتر کردن

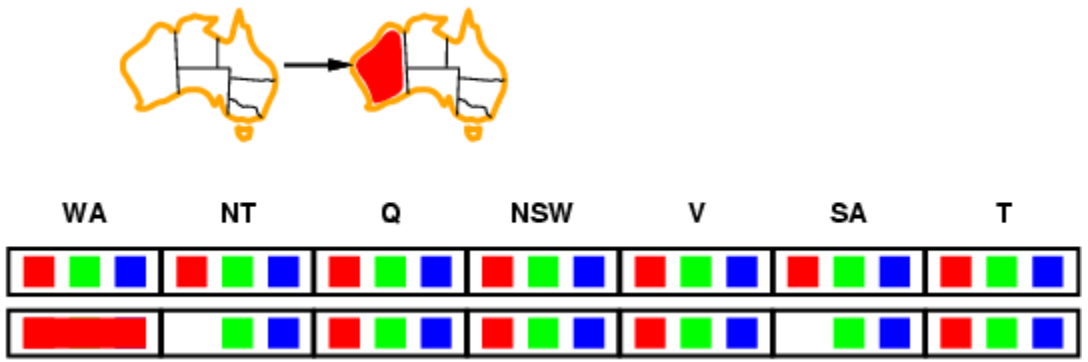
فیلتر کردن: بررسی رو به جلو

- فیلتر کردن. حذف مقادیر بی‌فایده از دامنه متغیرهایی که هنوز مقداردهی نشده‌اند.
- بررسی رو به جلو. حذف مقادیری که اگر به انتساب‌های فعلی اضافه شوند، باعث نقض محدودیت‌ها می‌شوند.
  - مقادیر مجاز باقیمانده برای متغیرهای مقداردهی نشده را در نظر بگیر.
  - زمانی که یک متغیر هیچ مقدار مجازی ندارد، به جستجو پایان بده.



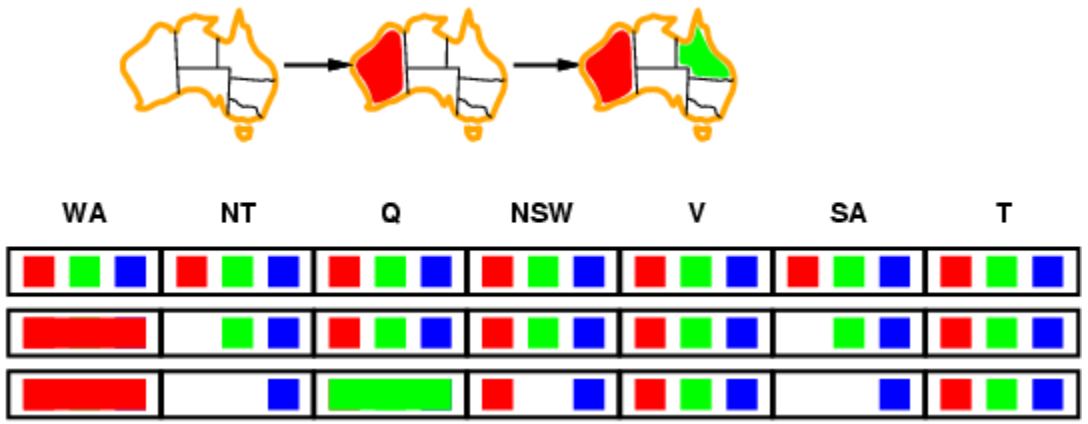
فیلتر کردن: بررسی رو به جلو

- فیلتر کردن. حذف مقادیر بی‌فایده از دامنه متغیرهایی که هنوز مقداردهی نشده‌اند.
- بررسی رو به جلو. حذف مقادیری که اگر به انتساب‌های فعلی اضافه شوند، باعث نقض محدودیت‌ها می‌شوند.
  - مقادیر مجاز باقیمانده برای متغیرهای مقداردهی نشده را در نظر بگیر.
  - زمانی که یک متغیر هیچ مقدار مجازی ندارد، به جستجو پایان بده.



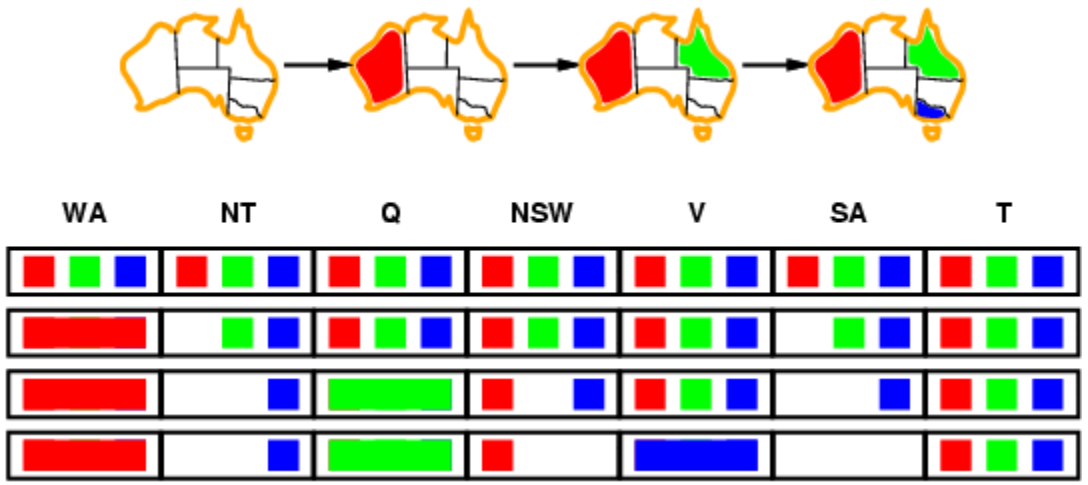
فیلتر کردن: بررسی رو به جلو

- فیلتر کردن. حذف مقادیر بی‌فایده از دامنه متغیرهایی که هنوز مقداردهی نشده‌اند.
- بررسی رو به جلو. حذف مقادیری که اگر به انتساب‌های فعلی اضافه شوند، باعث نقض محدودیت‌ها می‌شوند.
  - مقادیر مجاز باقیمانده برای متغیرهای مقداردهی نشده را در نظر بگیر.
  - زمانی که یک متغیر هیچ مقدار مجازی ندارد، به جستجو پایان بده.



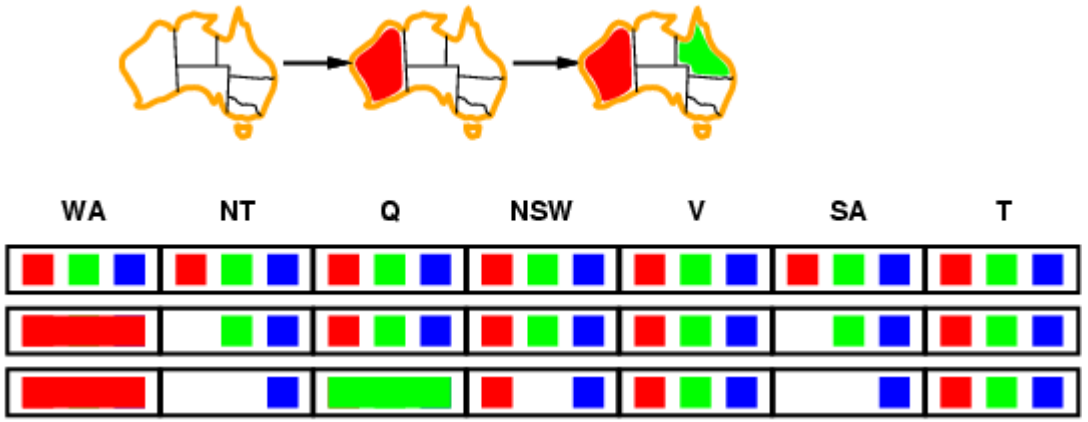
فیلتر کردن: بررسی رو به جلو

- فیلتر کردن. حذف مقادیر بی‌فایده از دامنه متغیرهایی که هنوز مقداردهی نشده‌اند.
- بررسی رو به جلو. حذف مقادیری که اگر به انتساب‌های فعلی اضافه شوند، باعث نقض محدودیت‌ها می‌شوند.
  - مقادیر مجاز باقیمانده برای متغیرهای مقداردهی نشده را در نظر بگیر.
  - زمانی که یک متغیر هیچ مقدار مجازی ندارد، به جستجو پایان بده.



فیلتر کردن: انتشار محدودیت

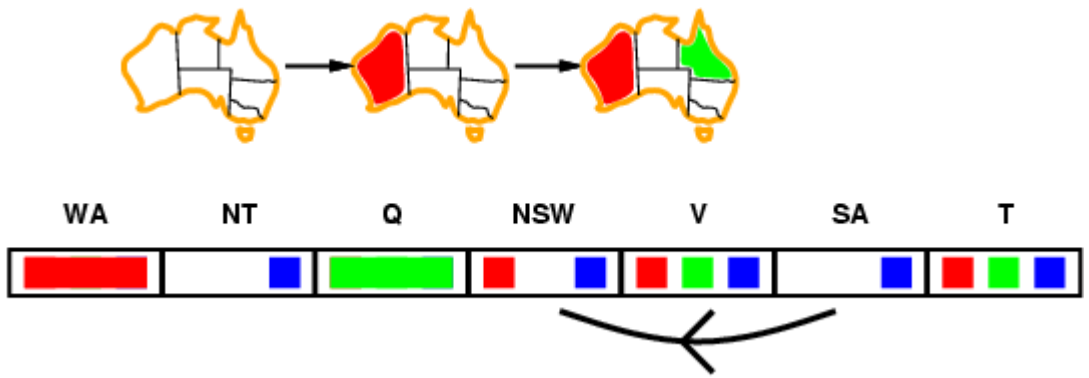
- بررسی رو به جلو محدودیت‌ها را از متغیرهای انتساب یافته به متغیرهای انتساب نیافته منتشر می‌کند، اما تمام شکست‌ها را نمی‌تواند در زودترین زمان ممکن تشخیص دهد.



- NT و SA هر دو نمی‌توانند آبی باشند!
- انتشار محدودیت به طور مکرر بر سازگاری محدودیت‌ها به طور محلی تأکید دارد.

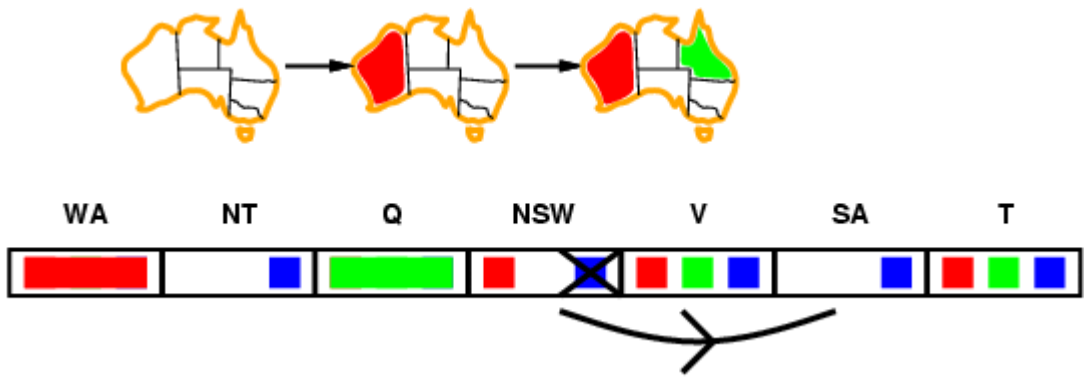
سازگاری کمان

- سازگاری کمان. کمان $X \rightarrow Y$ سازگار است اگر به ازای هر مقدار x حداقل یک مقدار y وجود داشته باشد که با x سازگار باشد.



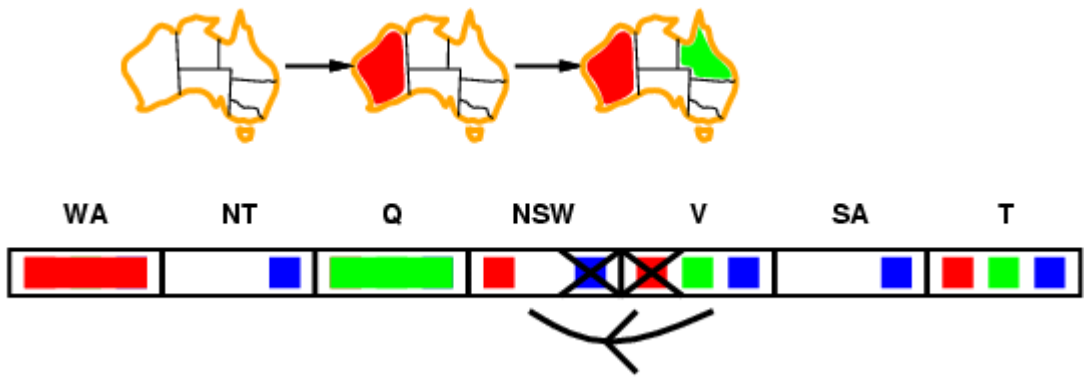
سازگاری کمان

- سازگاری کمان. کمان $X \rightarrow Y$ سازگار است اگر به ازای هر مقدار x حداقل یک مقدار y وجود داشته باشد که با x سازگار باشد.



سازگاری کمان

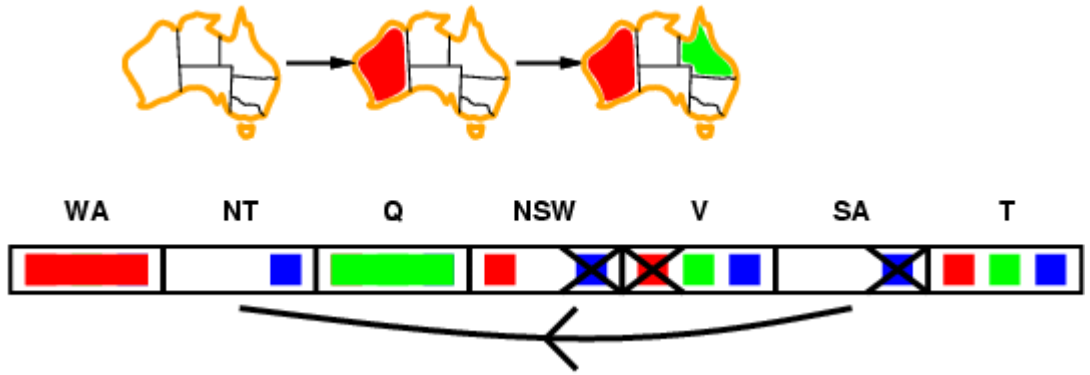
- سازگاری کمان. کمان $X \rightarrow Y$ سازگار است اگر به ازای هر مقدار x حداقل یک مقدار y وجود داشته باشد که با x سازگار باشد.



- توجه. اگر مقداری از دامنه X حذف شود، همسایه‌های X نیاز به بررسی مجدد دارند.

سازگاری کمان

- سازگاری کمان. کمان $X \rightarrow Y$ سازگار است اگر به ازای هر مقدار x حداقل یک مقدار y وجود داشته باشد که با x سازگار باشد.



- توجه. سازگاری کمان شکست‌ها را زودتر از FC تشخیص می‌دهد و می‌تواند به عنوان یک پیش‌پردازش و یا بعد از هر انتساب اجرا شود.

الگوریتم سازگاری کمان

function AC-3(csp) **returns** the CSP, possibly with reduced domains

$O(n^2 d^3)$

inputs: csp, a binary CSP with variables $\{X_1, X_2, \dots, X_n\}$

local variables: $queue$, a queue of arcs, initially all the arcs in csp

while $queue$ is not empty **do**

$(X_i, X_j) \leftarrow$ REMOVE-FIRST(queue)

if REMOVE-INCONSISTENT-VALUES(X_i, X_j) **then**

for each X_k **in** NEIGHBORS[X_i] **do**

add (X_k, X_i) to queue

---

function REMOVE-INCONSISTENT-VALUES(X_i, X_j) **returns** true iff succeeds

$removed \leftarrow false$

for each x **in** DOMAIN[X_i] **do**

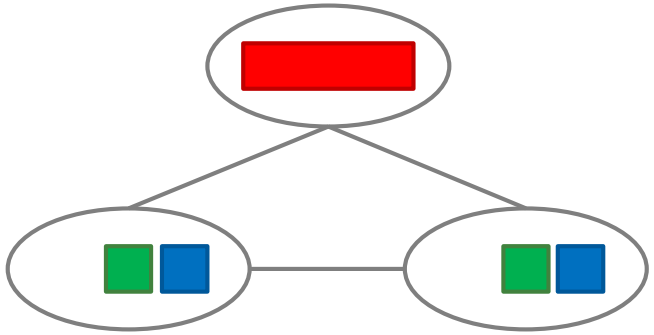
if no value y in DOMAIN[X_j] allows (x, y) to satisfy the constraint $X_i \leftrightarrow X_j$ **then**

delete x from DOMAIN[X_i]; $removed \leftarrow true$

return $removed$

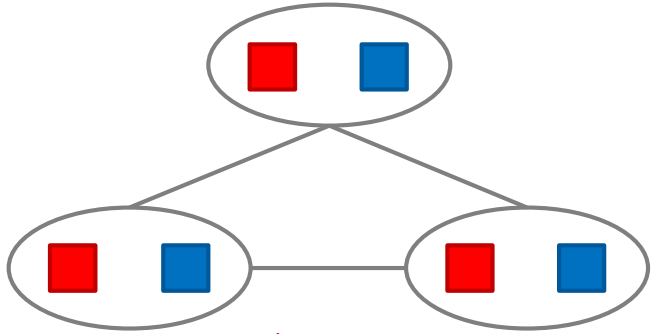
محدودیت‌های الگوریتم سازگاری کمان

- پس از سازگار کردن همه کمان‌ها.
  - ممکن است تنها یک راه‌حل باقی بماند.
  - ممکن است بیش از یک راه‌حل باقی بماند.
  - ممکن است هیچ راه‌حلی باقی نماند. [و ما ندانیم]



۲ راه‌حل

- سازگار کردن کمان‌ها همه شکست‌ها را تشخیص نمی‌دهد.
  - بنابراین، سازگاری کمان درون عقب‌گرد اجرا می‌شود.
  - سازگاری کمان پس از هر انتساب اجرا می‌شود.



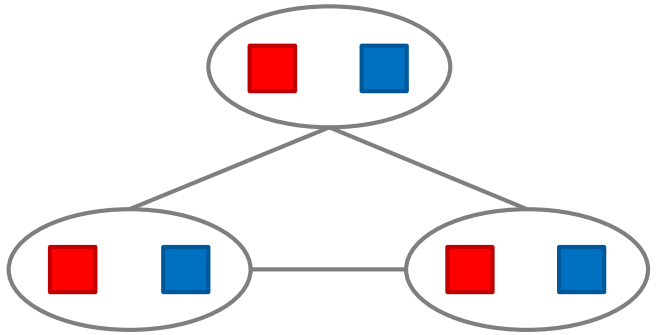
بدون راه‌حل

k-سازگاری

k-سازگاری

- افزایش درجه سازگاری.
  - ۱-سازگاری (سازگاری گره): دامنه هر گره شامل مقداری است که همه محدودیت‌های یکانی بر روی آن گره را برآورده می‌سازد.
  - ۲-سازگاری (سازگاری کمان): به ازای هر زوج از گره‌ها، هر انتساب سازگار به یک گره را می‌توان به گره دیگر تعمیم داد.
  - k-سازگاری: به ازای هر k گره، هر انتساب سازگار به k-1 گره را می‌توان به kاُمین گره تعمیم داد.

- افزایش k باعث افزایش پیچیدگی محاسبات می‌شود.



k-سازگاری قوی

- k-سازگاری قوی. k - 1، k - 2 و ... ۱ سازگار هم هست.
- ادعا. n-سازگاری قوی یعنی می‌توان مسئله را بدون عقب‌گرد حل نمود.
- اثبات.
  - یک متغیر انتخاب کن.
  - بر اساس ۱-سازگاری، یک انتخاب سازگار برای این متغیر وجود دارد.
  - یک متغیر جدید انتخاب کن.
  - بر اساس ۲-سازگاری، یک انتخاب سازگار با انتخاب اول وجود دارد.
  - یک متغیر جدید انتخاب کن.
  - بر اساس ۳-سازگاری، یک انتخاب سازگار با دو انتخاب اول وجود دارد.
  - ...

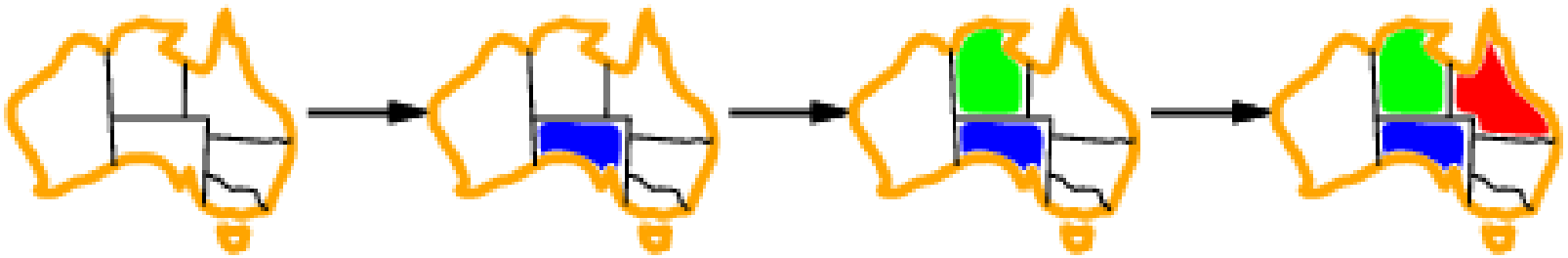
ترتیب‌دهی متغیرها

ترتیب‌دهی: کمترین مقادیر باقیمانده

- ترتیب‌دهی متغیرها. کمترین مقادیر باقیمانده (MRV)
  - متغیری را انتخاب کن که کمترین مقادیر باقیمانده را دامنه خود دارد.

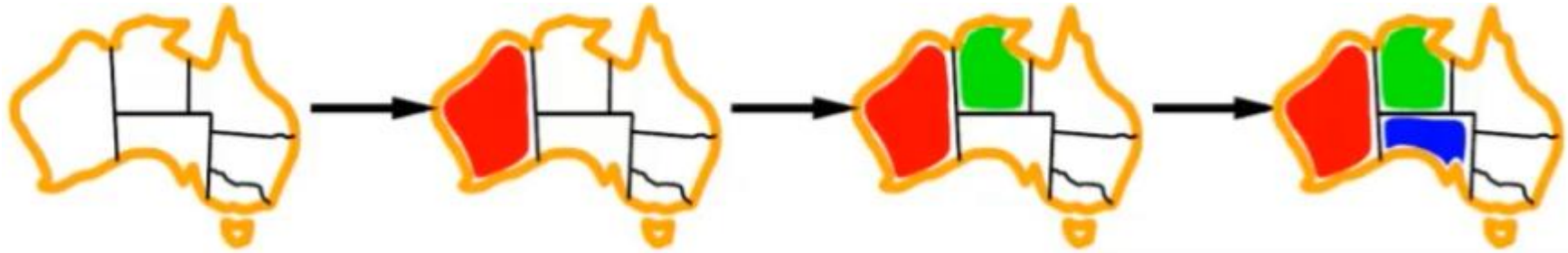
هیوریستیک درجه

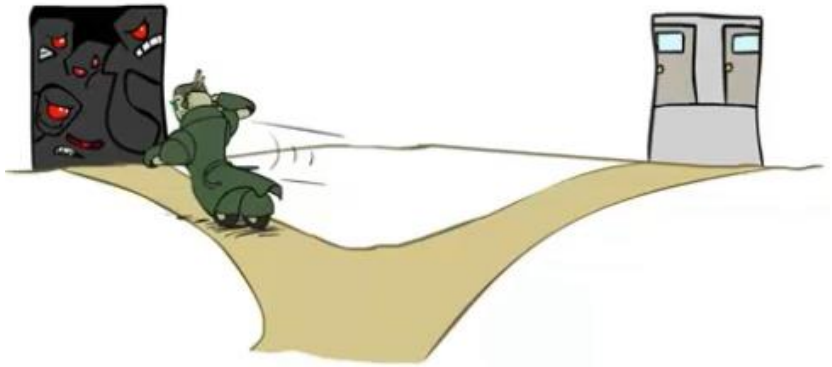
- هیوریستیک درجه.
  - رفع تصادم بین متغیرهایی که کمترین مقادیر باقیمانده را در دامنه خود دارند.
  - انتخاب متغیر با بیشترین درجه در گراف محدودیت‌ها
  - یعنی، انتخاب متغیری که بیشترین محدودیت را دارد.



ترتیب‌دهی: کمترین مقادیر باقیمانده

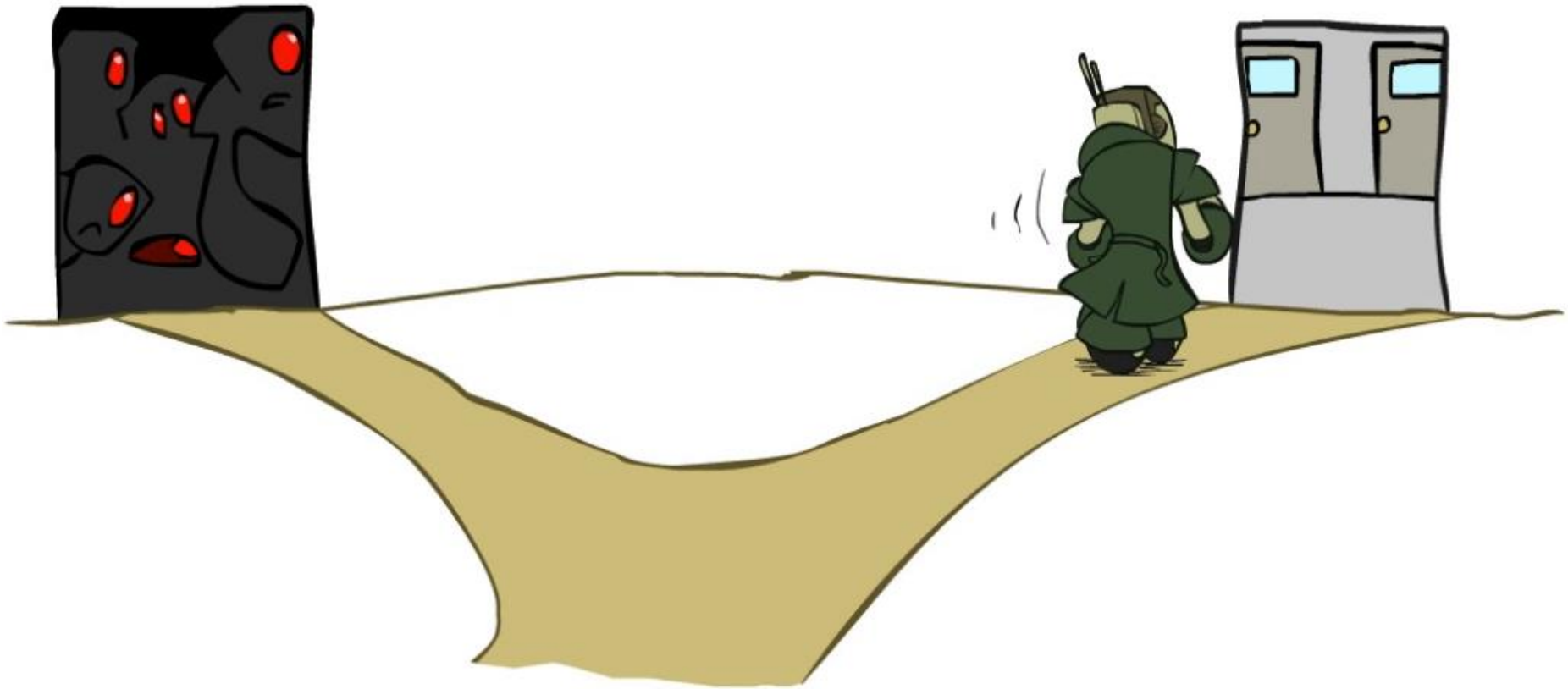
- ترتیب‌دهی متغیرها. کمترین مقادیر باقیمانده (MRV)
  - متغیری را انتخاب کن که کمترین مقادیر باقیمانده را دامنه خود دارد.





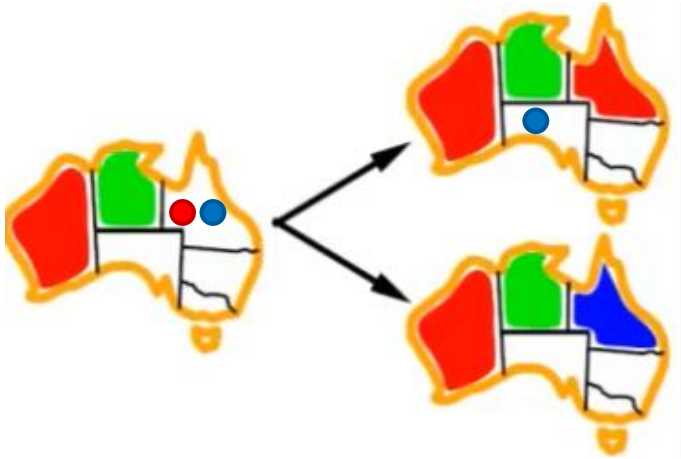
- س. چرا اول سخت‌ترین متغیر؟
- نام دیگر: «متغیر با بیشترین محدودیت»
- ترتیب دهی «اول-شکست»

ترتیب‌دهی مقادیر



ترتیب‌دهی مقادیر: مقدار با کمترین محدودیت

- ترتیب‌دهی مقادیر. مقدار با کمترین محدودیت
  - پس از انتخاب یک متغیر، مقداری را انتخاب کن که کمترین محدودیت را ایجاد می‌کند.
  - یعنی، مقداری که کمترین مقادیر را از دامنه متغیرهای باقیمانده حذف می‌کند.
  - تعیین چنین مقداری به اندکی محاسبات نیاز دارد (فیلتر کردن مجدد)



مقدار برای SA = آبی

مقدار برای SA = سبز

- با استفاده از ترتیب‌دهی مسئله ۱۰۰۰-وزیر قابل حل می‌شود.