

Artificial Intelligence Hw2 - Constraint Satisfaction Problems (CSPs)

B036060017 資管大四 謝威廷

1. 問題定義

Constraint Satisfaction Problems (CSPs) , 包含三個部分：

- A set of Variables
- A set of Domains for each variable
- A set of Constraint of values that each variable can assigned

目標是找到一組或多組可行且與限制不衝突的解。由於其將限制納入考量的特性，生活中許多常見的問題皆可以化為 CSPs 問題來解決，像是工作排程、資源分配等。

化為 CSPs 問題後，所有潛在可能為解的組合會形成搜尋空間，通常會利用搜尋的方式找出真正可行的解，常用的方式主要可分為無資訊（Uninformed）以及啟發式

（Heuristic）兩種，無資訊中以回溯法（Backtracking）為經典，但是其以暴力破解(Brute Force)類似於 Trial and Error 方式搜尋，時間複雜度相當高，若要加速搜尋效率，可以額外給予資訊用啟發式來實作，衍生出三種啟發式類型：挑選 Variable、挑選 Value 和限制傳播（Constraint Propagation）。

在這項作業當中，設定以 Radio Link Frequency Assignment Problem（RLFAP）作為案例進行分析。當今的時代，資通訊科技早已與我們的生活緊密連結，其中資訊傳遞多半倚靠無線電波段來進行傳輸，但是無線電波段並非無盡的資源，若要同時應付周遭所有地區、網路設備等上網需求，勢必要進行妥善的頻段分配，避免不必要的頻段衝突，造成資料傳輸的不穩定。

限縮探討美國 50 州的緊急通報頻域配置 CSP 問題，各州需要有自己的通訊頻段，惟頻段有限，無法切分出 50 個獨立段，設定僅有 5 段可供使用，且相鄰州間不可共用同頻段避免抵觸，進一步考量有些州的通訊設備老舊，僅適合特定的頻段，例如加州僅適用 A 段、德州僅適用 B 段等，以下化為 CSP 形式：

- Variable : 美國 50 州
- Domain : A, B, C, D, E 5 個無線電頻段
- Constraint : 相鄰州使用頻段不可一致、特定州僅適用特定頻段

2. 文獻探討（檔案附於資料夾中）

- Dr. M. Grötschel (2003), Solving Frequency Assignment Problems with Constraint Programming

針對 GSM 無線電網路收發站(Base Transceiver Stations, BTSs)，每個 BTS 共有三個發送接收器(Transmitter/Receiver Units, TRXs)，TRX 與無線電頻道串接來進行資訊傳遞，惟可用無線電頻道的數量相較於 TRXs 相差懸殊，因此本研究即是在探討何種演算法能高效率、妥善的分配無線電波段。

使用 TRX-based 及 Cell-based 等 Optimization Programming Language(OPL) 比較分析（表一），並應用啟發式的方式，嘗試將 Interference 的數量納入考量（表二）。結論為 Cell-based 結合啟發式的方法可以應付所有測試實驗，雖然 TRX-based 無法應付某些特定情況，但是其所需的搜尋時間僅為 Cell-based 的一半。

表一：TRX-based 和 Cell-based OPL 比較

	OPL script with TRX-based model			OPL script with cell-based model			Perl script with cell-based model		
Scenario	RAM [MB]	time [s]	costs	RAM [MB]	time [s]	costs	RAM [MB]	time [s]	costs
K	56	21.56	2.3	60	61.71	1.7	54	84.52	1.7
siemens1	106	65.82	8.8	80	65.00	12.6	86	81.03	12.6
siemens2	515	157.61	40.4	297	223.13	29.3	119	204.02	29.3
siemens3	-	-	-	204	93.53	55.6	237	99.41	55.6
siemens4	885	1472.36	252.8	500	1153.90	206.9	286	1271.11	206.9

Table 6.8: Minimizing total interference by means of OPL

表二：T-Coloring 和 Smallest Domain Size 啟發式比較

	with the heuristic <i>T-Coloring</i>		with the heuristic <i>Smallest Domain Size</i>		
Scenario	time [s]	costs	time [s]	costs	ref. value
K	13.57	336.3	0.16	41.9	0.45
K(0.02)	14.97	2.5	0.79	2.0	0.45
siemens1	127.63	290.0	1.40	52.0	2.30
siemens2	751.31	156.3	2.05	56.8	14.28
siemens3	>3600	-	>3600	-	5.19
siemens4	>3600	-	13.85	297.9	80.97
Swisscom	323286	-	27565	-	27.21
bradford_nt-10-eplus	>3600	-	20.34	1119.0	146.20

Table 6.7: Comparing the heuristics *T-Coloring* and *Smallest Domain Size*

- Cabon, Bertrand & de Givry, Simon & Lobjois, Lionel & Schiex, Thomas & P. Warners, Joost. (1999). Radio Link Frequency Assignment. Constraints.

同樣針對無線電波段配置進行分析，此研究以 1993 年 CELAR 建置無線電環境為例，環境中共有 916 個無線電使用需求，但最多僅有 48 個波段可供使用，當中如何不衝突，在短時間內調節分配頻段就是關鍵性的問題，研究中提出不同的衡量指標提供評斷分配的好壞依據（表三）。

表三：CELAR 案例中，以不同指標（Criterion, Crit.）來衡量

Table 1. The CELAR instances

Inst. #	# of Var.	# of Con.	Graph Dens.	Feas.	AC Enf. result	Crit.	Best found	Opt. proof
01	916	5 548	0.0132	Yes	AC	CARD	16	Yes
02	200	1 235	0.0620	Yes	AC	CARD	14	Yes
03	400	2 760	0.0345	Yes	AC	CARD	14	Yes
04	680	3 967	0.0171	Yes	13 868	CARD	46	Yes
05	400	2 598	0.0325	Yes	12 046	SPAN	792	Yes
06	200	1 322	0.0664	No	WO	MAX	3 389	Yes
07	400	2 866	0.0359	No	WO	MAX	343 592	No
08	916	5 745	0.0137	No	WO	MAX	262	No
09	680	4 103	0.0177	No	WO	MAX	15 571	14 875
10	680	4 103	0.0177	No	WO	MAX	31 516	31 204
11	680	4 103	0.0177	Yes	AC	CARD	22	Yes

3. 問題解法與改進

初步以無資訊搜尋中的回溯法為基礎，回溯法屬於暴力破解及深度優先(Depth-first Search, DFS) 解法，循序依節點賦值遞迴驗證是否可行，若是不可行即退回上一個節點，尋找其他可行方案，最終回傳可行解或是無解之結果，以下為 Pseudo-code：

```

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

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

```

Figure 6.5 A simple backtracking algorithm for constraint satisfaction problems. The algorithm is modeled on the recursive depth-first search. By varying the functions SELECT-UNASSIGNED-VARIABLE and ORDER-DOMAIN-VALUES, we can implement the general-purpose heuristics for better efficiency.

惟無資訊搜尋法需要遍歷所有搜尋空間，因此需耗費相當多的時間運算，可透過啟發式來改進效率，在上述問題定義中有提及三種類型，以下詳細介紹各類型：

1. 挑選 Variable：

a. Minimum Remaining Value, MRV：

優先以擁有最多限制的變數進行賦值，核心思想為解決較難的變數後，其他有較大彈性的變數即可迎刃而解，進而更快找到可行解。採行於作業中，即為先衡量那些僅支援特定頻寬的州，而後考量其鄰近州。

b. Degree Heuristic：

優先以對其他變數限制最大的變數進行賦值，核心思想為從擁有較大的分支度出發，來降低未來選擇其他變數的分支度，進而限縮搜尋空間。採行於作業當中，即為先衡量擁有最多周遭鄰近州的州開始。

2. 挑選 Value：

a. Least Constraining Value, LCV：

進行賦值時，優先考量賦值後依然擁有最大彈性的選擇，核心思想是為後續的變數配置，預留最大的空間，減少落入不可行解的可能性。

3. Forward Checking 限制傳播：

提早偵測未來可能產生的失敗，針對尚未配置的變數，持續的保持其合法值的追蹤(Arc Consistency)，當在特定的狀態下，有任何某個變數已經沒有任何可行解時，就中斷搜尋。

以上四種啟發式的搜尋法，皆可以提供相當的資訊量，讓搜尋方向更快的往可行空間前進，更有效率的找到解法。

4. 分析與討論

RLFAP 無線電頻段配置問題，其實就是教科書中 CSPs 經典的 Map Coloring 地圖著色問題，地圖著色中的顏色，即是可用的無線電頻段，而鄰近區域不能擁有相同顏色，即是要求鄰近區域不可共用同一頻段，將看似童趣的地圖著色問題，擴大到解決社會上存在的問題，帶來實際的價值。

可見資訊科學中的演算法，能解決的問題並不僅僅存在於電腦之中，也能同樣解決生活上的問題。美國電腦科學家 Brain Christian 及認知科學家 Tom Griffiths 出版了

Algorithms to Live By (中譯: 決斷的演算) 即講述如何將應用演算法於生活中, 例如應用 OS 排程管理中 Priority Scheduling, SJF 等, 來協助人們將時間管理最佳化。

最後, 經過這次作業深入熟悉 CSPs 問題後, 認為最難能可貴的是其將問題化約為三個部分, 考量既有條件下, 不斷提出許多方式, 嘗試在有限的時間和資源下, 找到可行解。未來生活中, 當我遇到相類似有限制條件的問題, 會嘗試用啟發式的方式, 思考如何解決, 將資訊思維(Computational Thinking)帶入生活上。

5. 實作

以 Python 來實作 RLFAP 無線電頻段分配問題, 演算法採用回溯法以及啟發式中的 Degree Heuristic。

第一步, 程式會先讀取美國 50 州及其鄰近州之資料, 並讀取外部限制條件之資料。第二步, 先依照限制條件分配頻段, 再依鄰近州數量由多到少排序, 由較多鄰近州的州開始分配頻段 (Degree Heuristic), 檢查頻段與周遭相異後才會確定分配。第三步, 若是無法分配與鄰近州皆相異之頻段會採取回溯法, 重新分配頻段。最後, 程式會輸出可行解, 以及當中回溯的次數, 供使用者參考。

Command Line Execute : `python RadioCSP.py constraints_example`