

車両装備仕様問題に対する 解集合プログラミングの適用

竹内 頼人 (名古屋大学情報学研究科)
番原 睦則 (名古屋大学情報学研究科)

2020 年 7 月 X 日

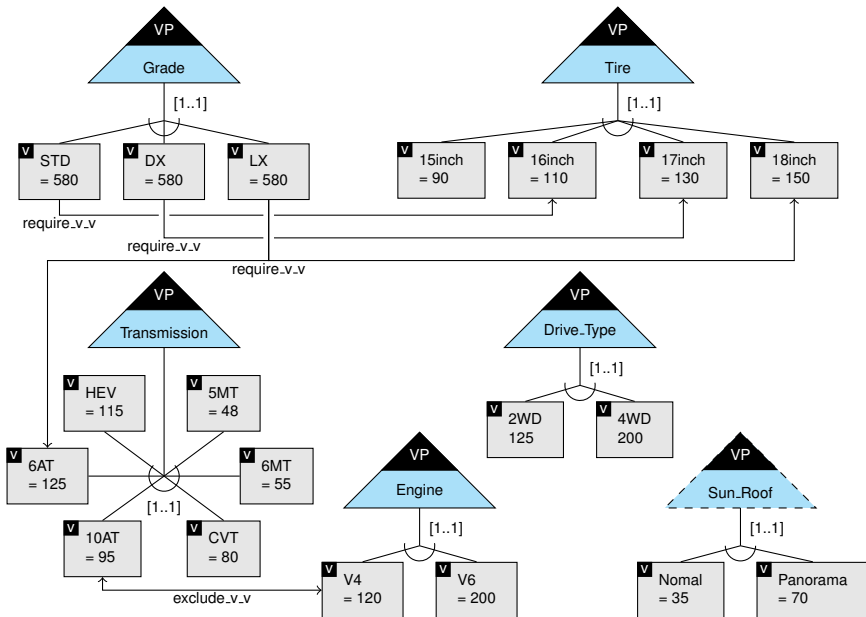
車両装備仕様問題

- 車両の装備を決定するには、販売される国や地域の法規や規制、地域や市場の特性、市場の嗜好や競合など十分に考慮する必要がある。
- 現状では専門知識をもつ技術者の多大な労力が費やされている。
- 日本では 2020 年度から、**企業別平均燃費** (Corporate Average Fuel Efficiency; **CAFE**) 基準と呼ばれる燃費規制が採用されている。

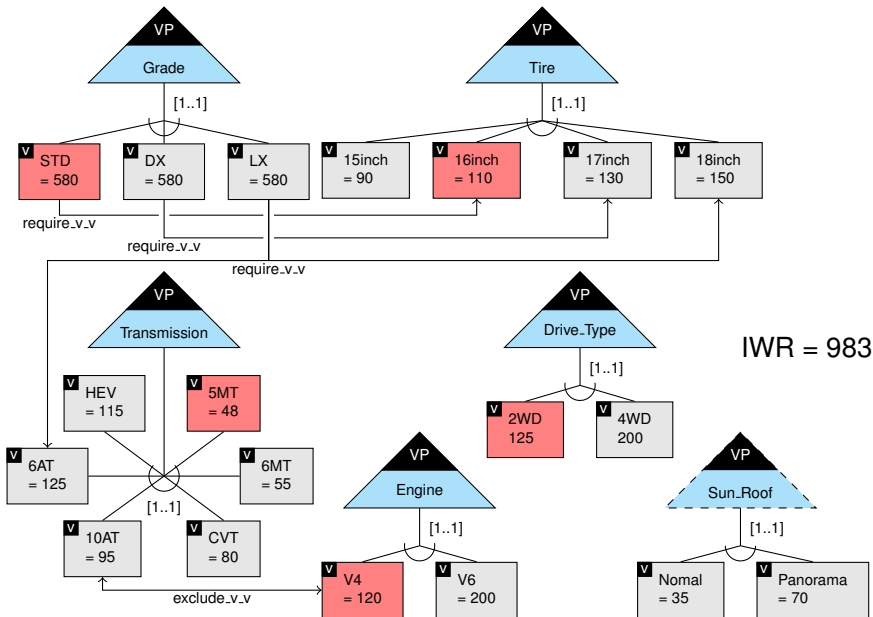
車両装備仕様問題

- 組合せ最適化問題として定式化される。
- **個数制約**, **要求・排他制約**, **燃費制約** (CAFE 基準) から構成される。
- **販売台数を最大化**する装備仕様 (装備タイプと装備オプションの組合せ) を求めることが目的である。
- 本研究では CAFE 基準に基づく車両装備仕様問題 (CAFE 問題) を対象とする。

CAFE 問題の例



CAFE 問題の例



燃費制約 (CAFE 基準) と目的関数

- 装備仕様 g の燃費を FE_g , 予想販売台数を SV_g , CAFE 基準値を X とする.

$$\frac{FE_1 \cdot SV_1 + FE_2 \cdot SV_2 + FE_3 \cdot SV_3}{SV_1 + SV_2 + SV_3} \geq X \quad SV_1 + SV_2 + SV_3 \rightarrow \text{最大}$$

- FE_g と SV_g は, 装備仕様 g の **IWR 値の和** から算出される.

最適解の例

- 装備仕様の数 G=3 (Grade はそれぞれ STD,DX,LX になるように設定)
- CAFE 基準値 X=9.0

装備	装備仕様		
	1	2	3
Grade	STD	DX	LX
Drive_Type	2WD	2WD	2WD
Engine	V4	V6	V6
Tire	16	17	18
Transmission	5MT	6MT	10AT
Sun_Roof	-	normal	-
IWR	983	1125	1180
燃費 (km/L)	10.2	8.9	8.5
予想販売台数	745	1988	1171
平均燃費 (km/L)	9.0		
合計販売台数	3904		

解集合プログラミング (Answer Set Programing; ASP)

- **ASP 言語**は、一階論理に基づく知識表現言語の一種である。
- **ASP プログラム**は、ASP ルールの有限集合である。
- **ASP システム**は、安定モデル意味論 [Gelfond and Lifschitz '88] に基づく解集合を計算するシステムである。
- 近年、SAT 技術を利用した高速な ASP システムが開発され、スケジューリング、システム生物学など様々な分野への実用的応用が急速に拡大している。

車両装備仕様問題に対して ASP を用いる利点

- ASP 言語の高い表現力により、各種制約を簡潔に記述できる。
- 高速な ASP システムを利用できる。
 - 一階 ASP プログラムを命題 ASP プログラムに**基礎化**した後、解集合を求めるシステムが主流
- 解の最適性を保証できる。最適解の列挙も可能である。

研究目的

目的

ASP 技術を活用して、大規模な車両装備仕様問題を効率よく解くシステムを実現すること。

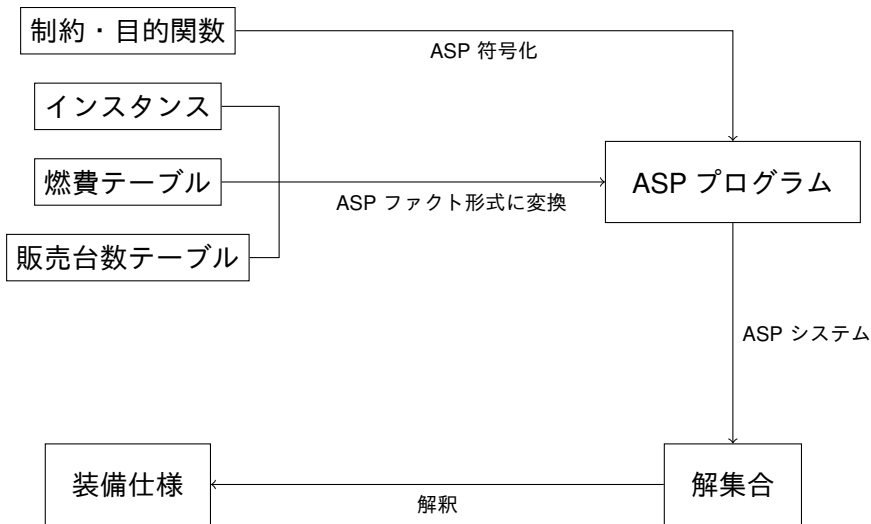
既存研究

- CAFE 問題に対する ASP 符号化 (基本符号化と呼ぶ)
- Babieca・神戸大学・名古屋大学 共同研究 (2019)

研究内容

- ① 改良符号化を提案
 - 基本符号化での問題点
 - **IWR 値の和の上下限を厳密に計算する**ように改良
- ② オプション数最小化
 - 販売台数の等しい複数の最適解に優劣をつけることが目的
 - コスト削減が見込める
- ③ 排他制約への対応

提案する車両装備ソルバーの概要



CAFE 問題の ASP ファクト形式

% VariationPoint & Variant

```
vp_def("Drive_Type").  
v_def("2WD", "Drive_Type", 125).  
v_def("4WD", "Drive_Type", 200).
```

```
vp_def("Engine").  
v_def("V4", "Engine", 120).  
v_def("V6", "Engine", 200).
```

```
vp_def("Grade").  
v_def("STD", "Grade", 580).  
v_def("DX", "Grade", 580).  
v_def("LX", "Grade", 580).
```

```
vp_def("Sun_Roof").  
v_def("Nomal", "Sun_Roof", 35).  
v_def("Pnrorama", "Sun_Roof", 70).
```

```
vp_def("Tire").  
v_def("15_inch_Tire", "Tire", 90).  
v_def("16_inch_Tire", "Tire", 110).  
v_def("17_inch_Tire", "Tire", 130).  
v_def("18_inch_Tire", "Tire", 150).
```

```
vp_def("Transmission").  
v_def("6AT", "Transmission", 115).  
v_def("10AT", "Transmission", 125).
```

```
v_def("HEV", "Transmission", 95).  
v_def("CVT", "Transmission", 80).  
v_def("5MT", "Transmission", 48).  
v_def("6MT", "Transmission", 55).
```

```
require_vp("Drive_Type").  
require_vp("Engine").  
require_vp("Grade").  
require_vp("Tire").  
require_vp("Transmission").
```

% Require

```
require_v_v("STD", "16_inch_Tire").  
require_v_v("DX", "17_inch_Tire").  
require_v_v("LX", "18_inch_Tire").  
require_v_v("LX", "10AT").
```

% Exclude

```
exclude_v_v("V4", "10AT").
```

```
group(1).  
group(2).  
group(3).  
:-not v("STD", 1).  
:-not v("DX", 2).  
:-not v("LX", 3).
```

基本符号化

% 企業平均燃費規制の値

```
#const t = 90.
```

```
{ vp(VP,G) } :- vp_def(VP), group(G).
```

% 個数制約

```
1 { v(V,G) : v_def(V,VP,_) } 1 :- vp(VP,G).
```

% 燃費制約 (CAFE)

```
iwr(S,G) :- S = #sum { IWR,V : v(V,G), v_def(V,_,IWR) }, group(G).
```

```
fe(FE,G) :- iwr(S,G), fe_map(S,FE).
```

```
sv(SV,G) :- iwr(S,G), S1 = ((S+2)/5)*5, sv_map(S1,SV).
```

```
:- not 0 <= #sum { (FE-t)*SV,FE,SV,G : fe(FE,G), sv(SV,G) }.
```

% 要求制約

```
:- not vp(VP,G), require_vp(VP), group(G).
```

```
:- require_v_v(V,W), v(V,G), not v(W,G).
```

```
:- require_vp_v(VP,V), vp(VP,G), not v(V,G).
```

```
:- require_v_vp(V,VP), v(V,G), not vp(VP,G).
```

% 販売台数を最大化

```
#maximize { SV,G : sv(SV,G) }.
```


改良符号化

% 企業平均燃費規制の値

#const t = 90.

{ vp(VP,G) } :- vp_def(VP), group(G).

% 個数制約

1 { v(V,G) : v_def(V,VP,_) } 1 :- vp(VP,G).

% IWR 値の和の上下限の計算 (前処理)

ub_vp(UB,VP) :- UB = #max { IWR,V : v_def(V,VP,IWR) }, vp_def(VP).

ub_iwr(S) :- S = #sum { UB,VP : ub_vp(UB,VP) }.

lb_vp(LB,VP) :- LB = #min { IWR,V : v_def(V,VP,IWR) }, vp_def(VP).

lb_iwr(S) :- S = #sum { LB,VP : lb_vp(LB,VP), require_vp(VP) }.

% 燃費制約 (CAFE)

iwr(S,G) :- S = #sum { IWR,V : v(V,G), v_def(V,_,IWR) },

LB <= S, S <= UB, lb_iwr(LB), ub_iwr(UB), group(G).

fe(FE,G) :- iwr(S,G), fe_map(S,FE).

sv(SV,G) :- iwr(S,G), S1 = ((S+2)/5)*5, sv_map(S1,SV).

:- not 0 <= #sum { (FE-t)*SV,FE,SV,G : fe(FE,G), sv(SV,G) }.

% 要求仕様制約

:- not vp(VP,G), require_vp(VP), group(G).

:- require_v_v(V,W), v(V,G), not v(W,G).

:- require_vp_v(VP,V), vp(VP,G), not v(V,G).

:- require_v_vp(V,VP), v(V,G), not vp(VP,G).

% 排他仕様制約

:- exclude_v_v(V,W), v(V,G), v(W,G).

:- exclude_v_vp(V,VP), vp(VP,G), v(V,G).

:- exclude_vp_vp(X,Y), vp(X,G), cp(Y,G).

% 販売台数を最大化

実験 1

考案した ASP 符号化の有効性を評価するために実験を行なった。

- ベンチマーク問題 (計 15 問)
 - 企業から提供された問題 (3 問) に対して
 - 5 通りの CAFE 基準値 $X \in \{8.5, 9.0, 9.5, 10.0, 10.5\text{km/L}\}$ で生成
 - 求める装備仕様の数 $G = 3$

問題	装備タイプ数	装備オプション数	要求制約数
small	8	21	4
medium	86	226	147
big	315	1,337	0

- ASP システム: *clingo-5.4.0*
- 制限時間: 1 問あたり 2 時間
- 実験環境: Mac mini, 3.2GHz, 64GB メモリ

実験結果: 得られた最適値と最良値

問題	CAFE 基準値	販売台数	
		基本符号化	改良符号化
small	8.5	6,021*	6,021*
small	9.0	5,007*	5,007*
small	9.5	2,688*	2,688*
small	10.0	1,318*	1,318*
small	10.5	UNSAT	UNSAT
medium	8.5	6,010	6,021
medium	9.0	5,595	5,595
medium	9.5	3,447	3,430
medium	10.0	2,245	2,250
medium	10.5	1,690	1,845
big	8.5	TO	3,877
big	9.0	1,038	4,623
big	9.5	688	3,121
big	10.0	1,634	2,064
big	10.5	538	904
最適値・最良値の数		6	13

- 改良符号化は、基本符号化と比較して、多くの問題に対してより良い解を得た。
- 大規模な問題に対する改良符号化の優位性が確認できた。

グループ数を大きくして比較を行った.

- ベンチマーク問題
 - 小規模な問題 (small) に対して
 - 5 通りの CAFE 基準値 $X \in \{8.5, 9.0, 9.5, 10.0, 10.5\text{km/L}\}$
 - 3 通りの装備仕様の数 $G = 3, 6, 12$ で生成
 - 要求制約を追加
- ASP システム: *clingo-5.4.0*
- 制限時間: 1 問あたり 1 時間
- 実験環境: Mac mini, 3.2GHz, 64GB メモリ

オプション数最小化

- インスタンスによっては最適解が複数得られる



装備	解 1			解 2			解 3		
	STD	DX	LX	STD	DX	LX	STD	DX	LX
Drive_Type	4WD	2WD	4WD	2WD	2WD	4WD	2WD	2WD	4WD
Engine	V4	V6	V6	V6	V6	V6	V6	V6	V6
Tire	16	17	18	16	17	18	16	17	18
Transmission	5MT	HEV	10AT	CVT	HEV	10AT	6AT	HEV	10AT
Sun_Roof	Panorama	-	-	Normal	-	-	-	-	-
IWR	1128	1130	1255	1130	1130	1255	1130	1130	1255
燃費 (km/L)	8.9	8.8	8.0	8.8	8.8	8.0	8.8	8.8	8.0
予想販売台数	2007	2007	1511	2007	2007	1511	2007	2007	1511
平均燃費 (km/L)	8.6			8.5			8.5		
合計販売台数	5525			5525			5525		
オプション数	14			13			12		

提案

最適解の中で優劣をつけるために目的関数として、オプション数の最小化を追加

オプション数最小化の実装例

- 選択された装備オプションの種類をカウントし、最小化する.
- 優先度を販売台数最大化の次にする.

% 販売台数を最大化

```
#maximize { SV@2,G : sv(SV,G) }.
```

% 共通オプション数を最大化

```
used_v(V) :- v(V,G).
```

```
#minimize {1@1,V : used_v(V) }.
```

オプション数最小化

実験概要

- オプション数最小化の有無で最適解の個数を比較
- インスタンスは OVM で示した CAFE 問題の例を使用

実験結果

問題	CAFE 基準値	販売台数	最適解の個数	
			改良符号化	+共通部品数最大化
small	8.5	5,525	3	1
small	9.0	3,904	2	2
small	9.5	1,699	1	1
small	10.0	UNSAT	-	-
small	10.5	UNSAT	-	-

- CAFE 基準値 8.5km/L では、最適解の個数が減少した
- CAFE 基準値 9.0km/L では、最適解が複数のままだった

- 車両装備仕様問題に対する 2 種類の **ASP** 符号化を考案
 - 問題を簡潔に記述できることを確認した (ASP のルール **11** 個).
 - 改良符号化は、基本符号化と比較して、**IWR 値の和の上下限を厳密に計算する**ように改良されている.
- 企業から提供された実データを用いた実験・評価
 - 小規模な問題に対して、その最適解を得ることができた.
 - 大規模な問題に対して、改良符号化の優位性が確認できた.

今後の課題

- 企業の技術者へのヒアリングに基づく問題拡張
 - 装備オプション間の排他制約などの新たな制約の追加
 - 共通部品の最大化などの新たな目的関数の追加

基本符号化のベースとなる制約モデル

問題の入力 (一部)

- V : 装備オプションの集合
- w_j : 装備オプション $j \in V$ の IWR 値
- G : 求めたい装備仕様の数

IWR 値の和に関する制約

$$x_{jg} \in \{0, 1\} \quad j \in V, g \in \{1, \dots, G\}$$

$$y_g \in \{0, \dots, \sum_{j \in V} w_j\} \quad g \in \{1, \dots, G\}$$

$$y_g = \sum_{j \in V} w_j x_{jg} \quad g \in \{1, \dots, G\}$$

- 0-1 変数 x_{jg} は、装備オプション j が装備仕様 g で選択されることを意味
- 整数変数 y_g は、装備仕様 g の IWR 値の和を表す。

改良符号化での制約モデル

必須装備タイプと個数制約の性質を利用して，IWR 値の和 (y_g) の上下限を厳密に計算することができる。

問題の入力 (一部)

- VP : 装備タイプの集合
- $VP^* \subseteq VP$: 必須装備タイプの集合
- $V_i \subseteq V$: 装備タイプ $i \in VP$ が選択できる装備オプションの集合

IWR 値の和に関する制約

$$\begin{aligned}x_{jg} &\in \{0, 1\} & j \in V, g \in \{1, \dots, G\} \\y_g &\in \left\{ \sum_{i \in VP^*} \min_{j \in V_i} w_j, \dots, \sum_{i \in VP} \max_{j \in V_i} w_j \right\} & g \in \{1, \dots, G\} \\y_g &= \sum_{j \in V} w_j x_{jg} & g \in \{1, \dots, G\}\end{aligned}$$

要求制約

要求制約 $X \rightarrow Y$ は、装備オプション X が選択されるならば、装備オプション Y も選択されなければならないことを意味する。

STD \rightarrow 16 インチ, DX \rightarrow 17 インチ, LX \rightarrow 18 インチ, LX \rightarrow AT

装備タイプ	装備オプション	IWR 値	装備仕様		
			1	2	3
グレード	STD	700	✓		
	DX	700		✓	
	LX	700			✓
エンジン	V4	120			✓
	V6	200	✓	✓	
タイヤ	16 インチ	110	✓		
	17 インチ	130		✓	
	18 インチ	150			✓
トランス ミッション	MT	55			
	HEV	95	✓	✓	
	AT	115			✓

IWR 値の和

装備タイプ	装備オプション	IWR	装備仕様		
			1	2	3
グレード	STD	700			
	DX	700			
	LX	700			
エンジン	V4	120			
	V6	200			
タイヤ	16 インチ	110			
	17 インチ	130			
	18 インチ	150			
トランス ミッション	MT	55			
	HEV	95			
	AT	115			

- 基本符号化: $y_g \in \{0, \dots, 3075\}$
- 改良符号化: $y_g \in \{985, \dots, 1165\}$

比較結果: 基礎化後のルール数

IWR 値の和に関する制約の基礎化後のルール数

問題	基本符号化	改良符号化	削減率 (%)
small	2,820	576	80% ↓
medium	7,545	1,758	80% ↓
big	62,745	1,488	98% ↓

- 改良符号化では、IWR 値の和の上下限を厳密に計算することで、基礎化後のルール数が大幅に削減されることが確認できた。

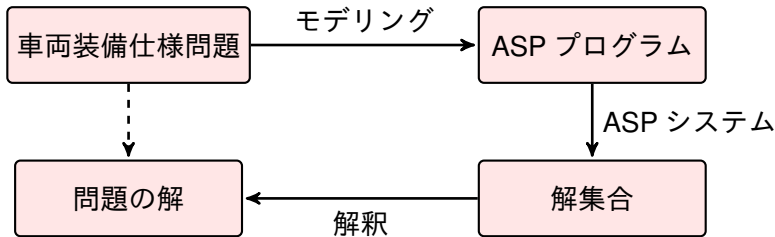
比較結果: CPU 時間

最適解が求まった問題インスタンスでの CPU 時間を比較する。

問題	CAFE 基準値 (km/L)	基本符号化 (s)	改良符号化 (s)
small	8.5	37.868	23.318
small	9.0	48.965	43.362
small	9.5	95.110	173.172
small	10.0	99.954	0.343
small	10.5	439.613	0.080
	平均	144.302	48.055

- 5 問中 4 問に対して、改良符号化の方が高速に最適解を求めた。
- 平均では、改良符号化の CPU 時間は基本符号化の約 1/3 であった。

ASP を用いた問題解法



- ① 問題を **ASP プログラム** としてモデリングする。
- ② ASP システムは、一階 ASP プログラムを命題 ASP プログラムに基礎化した後、
- ③ 解集合を解釈して元の問題の解を得る。 **解集合** (一種の最小モデル) を計算する。

車両装備仕様問題の例

装備タイプ	装備オプション	IWR 値	装備仕様		
			1	2	3
グレード	STD	700			
	DX	700			
	LX	700			
エンジン	V4	120			
	V6	200			
タイヤ	16 インチ	110			
	17 インチ	130			
	18 インチ	150			
トランス ミッション	MT	55			
	HEV	95			
	AT	115			

- 装備タイプはすべて**必須**とする。

車両装備仕様問題の例

装備タイプ	装備オプション	IWR 値	装備仕様		
			1	2	3
グレード	STD	700	✓		
	DX	700		✓	
	LX	700			✓
エンジン	V4	120			✓
	V6	200	✓	✓	
タイヤ	16 インチ	110	✓		
	17 インチ	130		✓	
	18 インチ	150			✓
トランス ミッション	MT	55			
	HEV	95	✓	✓	
	AT	115			✓

- 装備タイプはすべて**必須**とする。
- 実行可能解の例を **✓** マークで示す。

個数制約

各装備仕様 g , 各装備タイプ i に対して, i が g で選択されるならば, i の装備オプションのうち, ちょうど 1 つが g で選択される.

装備タイプ	装備オプション	IWR 値	装備仕様		
			1	2	3
グレード	STD	700	✓		
	DX	700		✓	
	LX	700			✓
エンジン	V4	120			✓
	V6	200	✓	✓	
タイヤ	16 インチ	110	✓		
	17 インチ	130		✓	
	18 インチ	150			✓
トランス ミッション	MT	55			
	HEV	95	✓	✓	
	AT	115			✓