Solving Vehicle Equipment Specification Problems with Answer Set Programing

Raito Takeuchi[1] Mutsunori Banbara[1] Naoyuki Tamura[2] Torsten Schaub[3]

[1]Nagoya University

[2]Kobe University

[3]Universität Potsdam

PADL2023@Boston
January 16th, 2023

Vehicle Equipment Specification

Vehicle Equipment Specification is the combination of vehicle models and equipments in automobile catalogs.

- The task of finding such combinations depends on various factors, including laws and regulations of coutries or regions, market characteristics, dependency between equipments, customer preferences, competitors, costs, and many others.
- A great deal of efforts by experienced engineers has been made.
- However, the task has become increasingly hard because of the recent globalization of automobile industry.

We tackle mono- and multi- objective vehicle equipment specification problems.

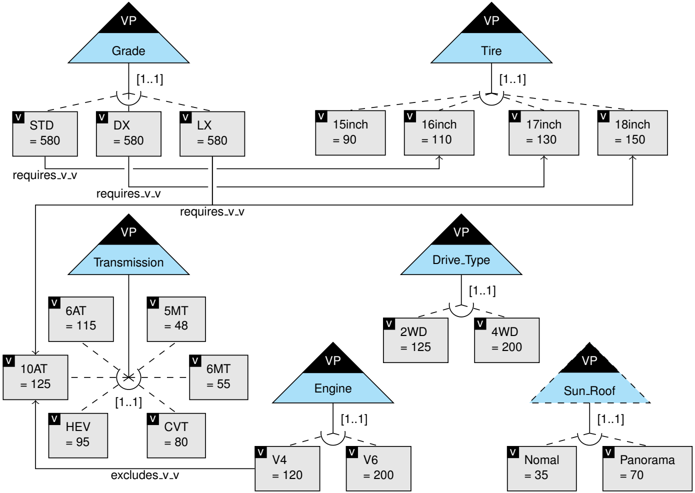
CAFE problems

CAFE problems are vehicle equipment specification problems based on CAFE standard [U.S. Congress, 1975] as a fuel economy regulation.

- **The vehicle equipment specification problem** is generally defined as the task of finding combinations of **equipment types** and **equipment options** for each vehicle model, subject to a given set of constraints on **variability**, **dependency**, and **fuel economy**.
- The objective of the problem is to find feasible solutions maximizing the **expected sales volume**.
- **The CAFE standard** (the Corporate Average Fuel Economy standard) has been adopted in the United States and the European Union, and was recently introduced in Japan.

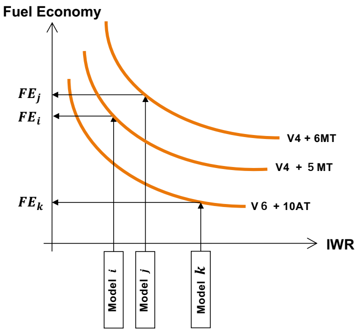
The CAFE problems are real-world applications in automobile industry.

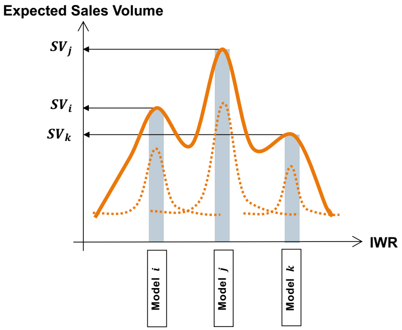
Input: a CAFE problem instance



- The example is expressed in the **OVM format**.
- 6 equipment types and 19 equipment options.
- Each type can select exact one option (variability constraints).
- 5 dependency constraints.
- All types are mandatory types except Sun_Roof.
- The value of each option indicates the **Inertial Working Rating (IWR)**, which intuitively corresponds to its weight.

Input: the number of vehicle models (n) and two curves





- For each model $1 \leq g \leq n$, the IWR value of g is the sum of IWR values of options implemented for g.
- **The fuel economy of g (FE_g)** can be obtained from the IWR value of g through a correlation curve.
- In the same way, **the expected sales volume of g (SV_g)** can be obtained through a distribution curve.

Fuel economy constraints

- Fuel economy regulations have been tightened in many countries due to recent international movements against global warming.
- The CAFE standard requires each automaker to achieve a target for the **sales-weighted fuel economy** of its entire automobile fleet in each model year.
- The following CAFE standard must be satisfied.

$$\underbrace{\frac{\sum_{g=1}^{n} FE_g \cdot SV_g}{\sum_{g=1}^{n} SV_g}}_{\text{sales-weighted fuel economy}} \geq \underbrace{t}_{\text{the CAFE standard value}}$$

Example of an optimal solution ($n = 3, \quad t = 8.5km/L$)

vehicle model	1	2	3
Grade	STD	DX	LX
Drive_Type	2WD	2WD	4WD
Engine	V6	V6	V6
Tire	16inch	17inch	18inch
Transmission	6AT	HEV	10AT
Sun_Roof	-	-	-

sum of IWR values	1,130	1,130	1,255
fuel economy (km/L)	8.8	8.8	8.0
expected sales volume	2,007	2,007	1,511
sales-weighted FE (km/L)	8.581		
sum of expected SV	5,525		

- Although the vehicle model 3 does not meet the criterion, the overall solution satisfied the CAFE standard.

Answer Set Programming (ASP)

ASP is a declarative programming paradigm, combining a rich modeling language with high performance solving capacities.

- ASP is well suited for modeling combinatorial optimization problems, and has been successfully applied in diverse areas of AI:
  - **Product Configuration**
  - Robotics
  - Systems Biology
  - Timetabling, and many more.
- Moreover, recent advances in ASP open up promising directions to make it more applicable to real world problems:
  - **Preference handling** [Brewka+, 2015]
  - Constraint ASP [Balduccini and Lierler, 2013]
  - Temporal ASP [Cabalar+, 2018]
  - Multi-shot ASP solving [Gebser+, 2019], and many others
- Such advances encourage researchers and practitioners to use ASP for solving practical applications.

The overview of our proposal

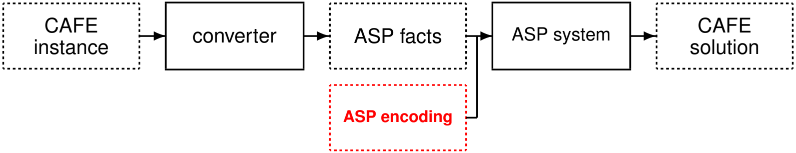
Proposal

- The *aspcafe* approach to solving mono-/multi-objective CAFE problems based on ASP,
- ASP technology allows to scale to real-world instances provided by a collaborating Japanese automaker.

The overview of our proposal

Proposal

- The *aspcafe* approach to solving mono-/multi-objective CAFE problems based on ASP,
- ASP technology allows to scale to real-world instances provided by a collaborating Japanese automaker.



1. *aspcafe* reads a CAFE instance in the OVM format and converts it into ASP facts.
2. These facts are combined with an **ASP encoding** for CAFE solving,
3. which can subsequently be solved by off-the-shelf ASP systems, in our case *clingo* and *asprin*.

The collection of *aspcafe* encodings

Mono-objective CAFE problems

- **Basic encoding** can concisely represent the constraints of the CAFE problem by using only 17 ASP rules.
- **Optimized encoding** can reduce the number of rules after grounding because it strictly calculates the upper and lower bounds of the sum of IWR values.
  - It can be more effective on large problems than the basic one.

Multi-objective CAFE problems

- It involves multiple objective functions that are considered separately and optimized simultaneously.
- **Extended Encoding** is an extension of optimized encoding to find the Pareto front (viz. the set of Pareto optimal solutions).
- This extension can be easily done by utilizing the *asprin* system, an enhancement of *clingo* with preference handling [Brewka+,2015].

ASP fact format of a CAFE instance

```
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("Panorama", "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).
```

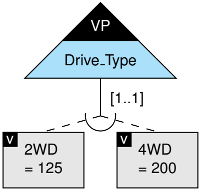
```
vp_def("Transmission").
v_def("6AT",  "Transmission", 115).
v_def("10AT", "Transmission", 125).
v_def("HEV",  "Transmission",  95).
v_def("CVT",  "Transmission",  80).
v_def("5MT",  "Transmission",  48).
v_def("6MT",  "Transmission",  55).

require_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_v_v("V4",  "10AT").

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

group(1). group(2). group(3).
```

Basic encoding: variability constraints




```
(1) { vp(VP,G) } :- vp_def(VP), group(G).
(2) 1 { v(V,G) : v_def(V,VP,_) } 1 :- vp(VP,G).
```

(1) The atom **`vp(VP,G)`** represents that the vehicle model `G` implements the equipment type `VP`. This rule introduces the atom `vp(VP,G)` as a solution candidate for each model `G` and each type `VP`.

(2) The atom **`v(V,G)`** represents that the model `G` implements the option `V`. This rule enforces that the model `G` implements exact one option `V` of type `VP` if `vp(VP,G)` holds.

Basic encoding: fuel economy constraints

```
(5) iwr(S,G) :-
       S = #sum { IWR,V : v(V,G), v_def(V,_,IWR) }, group(G).
(6) fe(FE,G) :- iwr(S,G), fe_map(S,FE).
(7) sv(SV,G) :- iwr(S,G), sv_map(S,SV).
(8) :- not 0 #sum { (FE-t)*SV,FE,SV,G : fe(FE,G), sv(SV,G) }.
(9) #maximize { SV,G : sv(SV,G) }.
```

(5) The atom **`iwr(S,G)`** represents that `S` is the sum of IWR values of options implemented for the model `G`.

(6) The atom **`fe(FE,G)`** represents that the fuel economy of `G` is `FE`, which is obtained from the IWR value `S` of `G` (`iwr(S,G)`) through a correlation curve (**`fe_map(S,FE)`**).

(8) The fuel economy constarint is transformed into $\sum_{i=1}^{n}(FE_i - t) \cdot SV_i \geq 0$ and is then represented in ASP's weighted cardinality constraints.

(9) The rule defines the objective function of maximizing the expected sales volume.

The problem issue of basic encoding

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

- The rule (5) is very expensive for grounding and solving.
- The domain of S is quite naive: $0 \leq \mathtt{S} \leq \sum_{j \in V} w_j$. The lower bound means no option, and the upper bound means the total sum of IWR values of all possible options.
- **This issue can be resolved** by considering, for each type i, the minimum and maximum IWR values of options which i can select, whether i is a mandatory type or not.

$$\sum_{i \in VP^*} \min_{j \in V_i} w_j \leq \mathtt{S} \leq \sum_{i \in VP} \max_{j \in V_i} w_j$$

- VP: the set of types
- VP^*: the set of mandatory types
- V: the set of options
- V_i: the set of options that the type $i \in VP$ can select.
- w_j: the IWR value of option $j \in V$

Optimized Encoding

- The optimized encoding generates the atom **iwr(S,G)** based on the improved domain.
- The optimized encoding is obtained from the basic encoding by replacing the rule (5) with the following code.

```
(5') 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).

(10) ub_vp(UB,VP) :- UB = #max { IWR,V : v_def(V,VP,IWR) }, vp_def(VP).
(11) lb_vp(LB,VP) :- LB = #min { IWR,V : v_def(V,VP,IWR) }, vp_def(VP).

(12) ub_iwr(S) :- S = #sum { UB,VP : ub_vp(UB,VP) }.
(13) lb_iwr(S) :- S = #sum { LB,VP : lb_vp(LB,VP), require_vp(VP) }.
```

Experiments: Mono-objective CAFE problems

We carry out experiments on three benchmark instances based on **real data** provided by a collaborating Japanese automaker.

Instance	#Types	#Options	#Dependency constraints
`small`	8	21	4
`medium`	86	226	147
`big`	315	1,337	0

- Our empirical analysis considers all combinations (15 in total) of
  - three instances,
  - five different CAFE standard values $t \in \{8.5, 9.0, 9.5, 10.0, 10.5\}$
- The number of vehicle models is $n = 3$.
- ASP system: *clingo* version 5.5.2
- Time limit: 3 hours for each

Comparison of Basic and Optimized encodings

Instance	CAFE standard value t (km/L)	Expected sales volume	
		Basic encoding	Optimized encoding
`small`	8.5	**6,021***	**6,021***
	9.0	**5,007***	**5,007***
	9.5	**2,688***	**2,688***
	10.0	**1,318***	**1,318***
	10.5	UNSAT	UNSAT
`medium`	8.5	6,010	**6,021**
	9.0	**5,595**	**5,595**
	9.5	**3,430**	**3,430**
	10.0	2,245	**2,250**
	10.5	**1,845**	**1,845**
`big`	8.5	N.A	**3,877**
	9.0	1,038	**4,623**
	9.5	688	**3,121**
	10.0	1,634	**2,100**
	10.5	538	**1,529**
#best bounds		7	**14**

- The optimized encoding was able to produce much better bounds for all combinations of the `big` instance, than the basic encoding.

Multi-objective CAFE Problem

We here extend the *aspcafe* encoding to find the **Pareto front** (viz. the set of Pareto optimal solutions) of two-objective CAFE Problems.

- One criterion is the **maximization of the expected sales volume**.
- The other is the **minimization of the number of equipment options**.
- The latter aims at reducing the number of production lines as well as at promoting mass production.
- This extension can be easily done by utilizing the *asprin*'s **preference statement**.

asprin's preference statement

$$\texttt{\#preference(}s\texttt{,}t\texttt{)\{}e_1\texttt{;...;}e_n\texttt{\}.}$$

s is the preference name, t is the preference type, and e_i is the preference element.

Extended Encoding : objective functions

```
(a) #preference (max_sv, more(weight)) { SV,G :: sv(SV,G) }.
(b) #preference (min_op, less(weight)) { 1,V :: used_v(V) }.
(c) #preference (all, pareto) { **max_sv; **min_op}.
(d) #optimize(all).
```

(a) The preference **max_sv** aims at maximizing the expected sales volume.

(b) The preference **min_op** aims at minimizing the number of equipment options.

(c) These two preferences are combined as **all**, according to the `pareto` type.

(d) The combined preference **all** is declared subject to optimization.

Example of a Pareto front ($n = 3$, $t = 8.5km/L$)

	Solution 1			Solution 2			Solution 3			Solution 4		
Vehicle model	1	2	3	1	2	3	1	2	3	1	2	3
Grade	STD	DX	LX	STD	DX	LX	STD	DX	LX	STD	DX	LX
Drive_Type	2WD	2WD	4WD	2WD	2WD	4WD	2WD	2WD	2WD	2WD	2WD	2WD
Engine	V6	V6	V6	V6	V6	V6	V6	V6	V6	V6	V6	V6
Tire	16	17	18	16	17	18	16	17	18	16	17	18
Transmission	6AT	HEV	10AT	10AT	HEV	10AT	10AT	HEV	10AT	10AT	10AT	10AT
Sun_Roof	-	-	-	-	-	-	-	-	-	-	-	-
Sum of IWR values	1,130	1,130	1,255	1,140	1,130	1,255	1,140	1,130	1,180	1,140	1,160	1,180
Fuel economy(km/L)	8.8	8.8	8.0	8.8	8.8	8.0	8.8	8.8	8.5	8.8	8.6	8.5
Expected sales volume	2,007	2,007	1,511	1,957	2,007	1,511	1,957	2,007	1,171	1,957	1,595	1,171
Sales-weighted FE	8.5			8.5			8.7			8.6		
Sum of SV	**5,525**			**5,475**			**5,135**			**4,723**		
The number of options	**12**			**11**			**10**			**9**		

- The Pareto front consists of four Pareto optimal solutions, not dominated by each other.
- The expected sales volume decreases from left to right.
- In contrast, the number of options increases from right to left.
- The extended encoding succeeds in finding **well-balanced** solutions like Solution 2–3, which are not obtained by lexicographic optimization.

Benchmark results of finding Pareto optimal solutions

The extended encoding with *asprin*-3.1.1

Instance	CAFE standard value (km/L)	#feasible solutions	#Pareto optimal solutions	CPU time(s)
small	8.5	41,217	8*	34.697
	9.0	3,961	5*	1,102.318
	9.5	374	1*	93,336.660
	10.0	28	1*	1.959
	10.5	0	0	0.295

- The extended encoding was able to find the Pareto front of all combinations of the `small` instance and CAFE standard values.
- On the other hand, we met the difficulty of finding Pareto optimal solutions for both `medium` and `big` instances.

Conclusion

We presented an ASP-based approach to solving mono- and multi-objective CAFE problems.

aspcafe

`https://github.com/banbaralab/aspcafe`

Future work

Our declarative approach can be extended to a wide range of regulations and constraints, such as:

- Multi-objective CAFE solving considering **minimal perturbation problems**,
- The **ZEV** (Zero-Emission Vehicle) regulation.