

解集合プログラミングを用いた多目的車両装備仕様問題の解法

Solving Multi-objective Vehicle Equipment Specification Problem with Answer Set Programming

竹内 頼人 *¹ 田村 直之 *² 番原 睦則 *¹
Raito Takeuchi Naoyuki Tamura Mutsunori Banbara

*¹名古屋大学 大学院情報学研究科
Graduate School of Informatics, Nagoya University

*²神戸大学 情報基盤センター
Information Science and Technology Center, Kobe University

Here is an abstract of up to 150 words in English. This document describes the format guidelines for Japanese manuscripts in L^AT_EX of the annual conference of JSAI. This is also a sample of a formatted manuscripts (see 2.4 for writing the abstract).

車両装備仕様とは、簡単に言うと、自動車のカatalogに記載されているモデル/グレードと装備の組合せのことである。多目的車両装備仕様問題は、与えられたモデル/グレードの個数、装備タイプの集合、装備オプションの集合などから、装備および燃費に関する制約を満たしつつ、予想販売台数の最大化や装備オプション数の最小化など、トレードオフの関係にある複数の目的関数のもとで最適な車両装備仕様を求める問題である。本発表では、CAFE 方式と呼ばれる燃費制約に基づく多目的車両装備仕様問題 (多目的 CAFE 問題) に対して、解集合プログラミングを用いてパレート最適解を列挙する方法について述べる。提案手法は、可変性モデルで表現された問題インスタンスを ASP のファクト形式に変換した後、それらファクトと多目的 CAFE 問題を解くための ASP 符号化と結合し、高速 ASP システムを用いて解を求める。企業から提供されたベンチマーク問題を用いた実行実験の結果、小規模な問題についてパレート最適解を全列挙することができた。

1. はじめに

車両装備仕様とは、簡単に言うと、自動車のカatalogに記載されているモデル/グレードと装備の組合せのことである。車両装備仕様を決めるには、販売される国や地域の法規や規制、地域や市場の特性、市場の嗜好や競争など十分に考慮する必要があり、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。そのため、車両装備仕様探索の自動化・効率化は自動車メーカーにとって重要な課題の一つである。

多目的車両装備仕様問題は組合せ最適化問題の一種であり、主に装備タイプと装備オプションから構成される。**装備タイプ**はエンジンやトランスミッションなどの装備の種類を表す。**装備オプション**は4気筒エンジン、CVTなどの具体的な装備を表す。多目的車両装備仕様問題の目的は、与えられた装備仕様の個数、装備タイプの集合、装備オプションの集合から、装備および燃費に関する制約を満たしつつ、予想販売台数の最大化や装備オプション数の最小化など、トレードオフの関係にある複数の目的関数のもとで最適な車両装備仕様を求めることである。

本研究では、燃費に関する制約として、欧米で採用され日本でも2020年度から導入されている**企業平均燃費** (Corporate Average Fuel Efficiency; CAFE [経済 18]) 方式を用いる。このCAFE方式は自動車の燃費規制で、車種別ではなくメーカー全体での出荷台数を加味した平均燃費を算出し、規制をかける方式である。CAFE方式の特長は、ある特定の車種では燃費基準を達成できなくても、他の車種の燃費を向上させることで基準を達成できることが可能な点である。本論文では、CAFE方式に基づく多目的車両装備仕様問題 (以降、**多目的 CAFE 問題**と呼ぶ) を対象とする。

解集合プログラミング (Answer Set Programming; ASP [Baral 03, Gelfond 88, 井上 08]) は、論理プログラミングから派生した宣言的プログラミングパラダイムである。ASP

言語は一階論理に基づく知識表現言語の一種である。論理プログラムはASPのルールの有限集合である。ASPシステムは論理プログラムから安定モデル意味論 [Gelfond 88] に基づく解集合を計算するシステムである。近年、SATソルバー技術を応用した高速ASPシステムが実現され、ロボット工学、システム生物学、システム検証、制約充足問題、プランニングなど様々な分野への実用的応用が急速に拡大している [Erdem 16]。

提案手法の有効性を評価するために、企業から提供を受けた小規模・中規模・大規模の多目的CAFE問題に対して実行実験を行った結果、小規模な問題では5問中4問で最適解の全列挙をすることができた。

2. 多目的CAFE問題

多目的CAFE問題は、与えられた入力から、装備および燃費に関する制約を満たしつつ、トレードオフの関係にある複数の目的関数のもとで最適な車両装備仕様を求める問題である。本節では、多目的CAFE問題の入力、制約、目的関数について説明し、例を用いて具体的な入力の内容と、そこから得られる装備仕様を示す。

多目的CAFE問題の入力は以下の通りである。以降、装備タイプを**タイプ**、装備オプションを**オプション**と簡単に書くことにする。

1. タイプの集合
2. オプションの集合
3. タイプとオプションの対応関係
4. 各オプションに付加されたIWR値
5. 各タイプで選択可能なオプション数の上下限值
6. 必須なタイプの集合
7. タイプ同士、オプション同士、および、タイプとオプション間の依存関係
8. 求めたい装備仕様の個数

連絡先: 竹内頼人, 名古屋大学 大学院情報学研究科, 住所, 電話番号, Fax 番号, takeuchi.raito@nagoya-u.jp

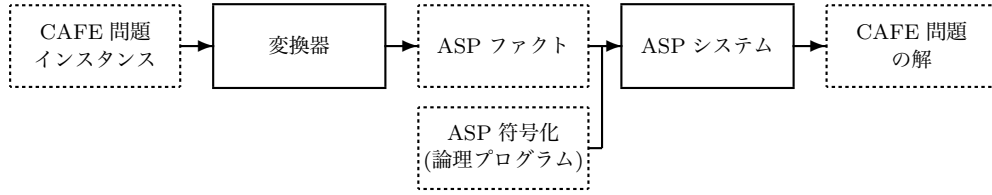


図 1: 提案する CAFE 問題ソルバーの構成

9. 各装備仕様とタイプ (あるいはオプション) 間の依存関係
10. CAFE 基準値
11. 各装備仕様に含まれるオプションの IWR 値の総和と燃費との対応表
12. 各装備仕様に含まれるオプションの IWR 値の総和と予想販売台数との対応表

入力 4 の IWR は Inertial Working Rating の略で、直観的には各オプションの重量を表す。入力 8 の個数は、求めたい派生車両の数と考えるとわかりやすい。入力 1~10 は問題ごとに異なるが、入力 11, 12 は共通のものを使用する。本論文では、これらの入力をまとめて多目的 CAFE 問題の**問題インスタンス**と呼ぶ。

多目的 CAFE 問題の制約は以下の通りである。

範囲制約：各装備仕様について、各タイプで選択されるオプション数は、入力 5 で与えられた上下限値の範囲内でないといけない。

依存制約：各装備仕様について、入力 7 で与えられた依存関係を満たさなければならない。依存制約には、**要求制約**と**排他制約**の 2 つがある。

燃費制約：入力 10 の CAFE 基準値を t 、入力 8 の装備仕様個数を n として、以下の CAFE 規制を満たさなければならない。

$$\frac{\sum_{i=1}^n FE_i \cdot SV_i}{\sum_{i=1}^n SV_i} \geq t$$

不等式の左辺は n 個の装備仕様の**平均燃費**を表している。 FE_i と SV_i は、装備仕様 i の燃費と予想販売台数を表しており、それぞれ、入力 11 と 12 の対応表を元に計算される。

初期制約：入力 9 で与えられた依存関係を満たさなければならない。

本研究では、多目的 CAFE 問題の目的関数として、以下の 2 つを用いる。

予想販売台数の最大化：各装備仕様ごとに求められる予想販売台数の合計を最大化する。

オプション数の最小化：装備仕様全体で、使用されるオプションの種類数を最小化する。これは、製造ラインの削減や、大量生産を促進することを狙いとしている。

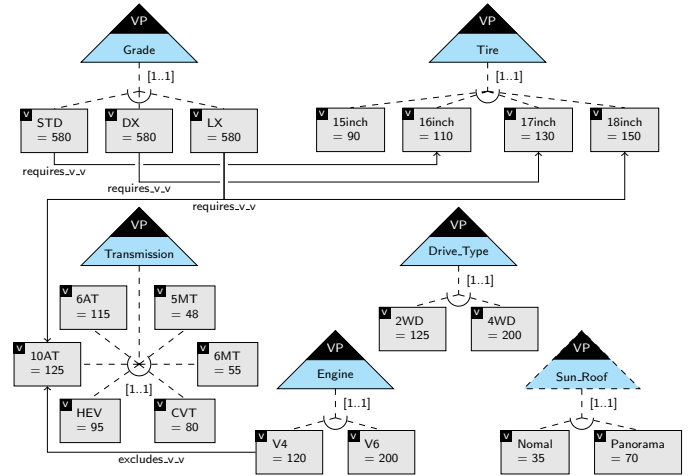


図 2: CAFE 問題の例

多目的 CAFE 問題の例を図 2 に示す。この例は、ソフトウェアプロダクトライン開発の分野で用いられる**可変性モデル** (Orthogonal Variability Model; OVM[Pohl 05]) によって記述されている。VP とタグ付けされている三角形のノードはタイプを、V とタグ付けされている四角形のノードはオプションとその IWR 値を表しており、入力 1, 2, 4 に相当する。対応関係にあるタイプとオプションは点線で結ばれており (入力 3)、付記される [1..1] のような多重度は入力 5 の各タイプで選択可能なオプション数の上下限値を意味する。タイプ同士、オプション同士、および、タイプとオプション間を結ぶ矢印は入力 7 の依存関係を表し、要求 (requires) と排他 (excludes) の 2 種類がある。

図 2 の問題例は、6 個のタイプ、19 個のオプション、5 個の依存制約から構成され、各タイプの選択可能なオプション数はすべて 1 である。本論文では、実線のノードで表すタイプは、各装備仕様に対して必須とし、Sun.Roof のような選択可能なタイプ (必須ではないタイプ) については、破線のノードで表すものとする。

図 2 の問題に対する解の例を表 1 に示す。この解は、CAFE 基準値に 8.5km/L、求めたい装備仕様の個数に 3 を与え、装備仕様とオプションの依存関係として、(装備仕様 1, STD)、(装備仕様 2, DX)、(装備仕様 3, LX) を要求して得られたものである。各装備仕様の燃費は、左から順に 8.8, 8.8, 8.0km/L と個々には CAFE 基準値を満たしていないものもあるが、3 台の平均燃費は 8.581km/L となり、CAFE 規制を満たしている。

表 1: 多目的 CAFE 問題 (図 2) の解

装備仕様	1	2	3
装備			
Grade	STD	DX	LX
Drive_Type	2WD	2WD	4WD
Engine	V6	V6	V6
Tire	16inch	17inch	18inch
Transmission	6AT	HEV	10AT
Sun_Roof	-	-	-
IWR 値の総和	1,130	1,130	1,255
燃費 (km/L)	8.8	8.8	8.0
予想販売台数	2,007	2,007	1,511
平均燃費 (km/L)	8.5		
予想販売台数 (合計)	5,525		
オプション数	12		

3. 多目的 CAFE 問題の ASP 符号化

ASP の言語は、一般拡張選言プログラム (General Extended Disjunctive Program) をベースとしている [井上 08]. 本節では、説明の簡略化のため、そのサブクラスである標準論理プログラムについて説明する。以降、標準論理プログラムを単に論理プログラムと呼ぶ。

論理プログラムは、以下の形式の**ルール**の有限集合である。

$$a_0 \leftarrow a_1, \dots, a_m, \sim a_{m+1}, \dots, \sim a_n \quad (1)$$

このルールの直観的な意味は、「 a_1, \dots, a_m がすべて成り立ち、 a_{m+1}, \dots, a_n のそれぞれが成り立たないならば、 a_0 が成り立つ」である。ここで、 $0 \leq m \leq n$ であり、各 a_i はアトム、 \sim は**デフォルトの否定**^{*1}、 $;$ は連言を表す。 \leftarrow の左側を**ヘッド**、右側を**ボディ**と呼ぶ。ボディが空のルール (すなわち $a_0 \leftarrow$) を**ファクト**と呼び、 \leftarrow を省略してよい。

ヘッドが空のルールを**一貫性制約**と呼び、以下のように表す。

$$\leftarrow a_1, \dots, a_m, \sim a_{m+1}, \dots, \sim a_n \quad (2)$$

例えば、一貫性制約 “ $\leftarrow a_1, a_2$ ” は、「 a_1 と a_2 が両方同時に成り立つことはない」を意味し、“ $\leftarrow a_1, \sim a_2$ ” は、「 a_1 が成り立つならば、 a_2 が成り立つ」を意味する。

ASP 言語には、組合せ問題を簡潔に記述するために、**アグリゲート** (aggregate) と呼ばれる拡張構文がいくつか用意されている。例えば、**選択子** “ $\{a_1; \dots; a_n\}.$ ” は、アトム集合 $\{a_1, \dots, a_n\}$ の任意の部分集合を解集合に含めることを意味する。**個数制約**は選択子の両端に選択可能な個数の上下限を付けたものである。例えば、“ $lb \{a_1; \dots; a_n\} ub \leftarrow Body$ ” と書くと、「 $Body$ が成り立つならば、 a_1, \dots, a_n のうち、 lb 個以上 ub 個以下が成り立つ」を意味する。**重み付き個数制約** “ $t = \#sum \{w_1 : a_1; \dots; w_n : a_n\}.$ ” は、 a_1, \dots, a_n のうち真となるアトムの重み和が項 t に等しくなることを意味する。項 w_i は重みを表し、演算子としては “=” 以外にも “ \leq ”, “ \geq ” などを使用できる。さらに、重み付き個数制約の “ $\#sum$ ” を、“ $\#max$ ” や “ $\#min$ ” に書き換えると、重み和ではなく、真となるアトムの重みの最大値や最小値を求めることができる。

近年、*clingo*^{*2}、*DLV*^{*3}、*WASP*^{*4} など、SAT ソルバー技術を応用した高速な ASP システムが開発されている。なか

*1 失敗による否定とも呼ばれる。述語論理で定義される否定 (\neg) とは意味が異なる。

*2 <https://potassco.org/>

*3 <http://www.dlvsystem.com/dlv/>

*4 <https://www.mat.unical.it/ricca/wasp/>

表 2: 問題インスタンスのアトム

アトム	対応する入力番号
$vp_def(vp)$	入力 1
$v_def(v, vp, x)$	入力 2, 3, 4
$require_vp(vp)$	入力 6
$require_v.v(v_1, v_2)$	入力 7
$exclude_v.v(v_1, v_2)$	入力 7
$group(1..n)$	入力 8
$require_g.v(i, v)$	入力 9
$fe_map(s, fe)$	入力 11
$sv_map(s, fe)$	入力 12

でも *clingo* は、高性能かつ高機能な ASP システムとして世界中で広く使われている。これらの高速 ASP システムは、変数を含む論理プログラムを変数を含まない論理プログラムに変換 (**基礎化**) したのち、ASP ソルバーを用いて解集合を計算する。本論文で使用する ASP システム *clingo* は、基礎化のためのグラウンダー *gringo* と ASP ソルバー *clasp* をシームレスに結合したものである。

本論文では、論理プログラムの解集合の中で、選好関係の宣言・評価を可能にするシステム *asprin* [Brewka 15] も使用する。選好関係は、**選好文**と呼ばれるプログラムの中で、次のように宣言される。

$$\#preference(s, t) \{e_1, \dots, e_n\} \quad (3)$$

ここで、 s と t はそれぞれ選好の名称とタイプであり、引数の各 e_j は選好の要素である。*asprin* では、新たに選好タイプを定義することも可能だが、いくつかの選好タイプはあらかじめライブラリに定義されており、それを利用することで簡潔に選好関係を記述することができる。そして、次のように最適化命令を記述することで、選好関係 s について最適解集合を得ることができる。

$$\#optimize(s) \quad (4)$$

このとき、*asprin* では繰り返し ASP ソルバーを呼び出すことで、最適解集合を段階的に計算する。

本論文で提案する多目的 CAFE 問題の解法では、与えられた問題インスタンスを ASP のファクト形式に変換した後、それらファクトと多目的 CAFE 問題を解くための ASP 符号化 (論理プログラム) を結合した上で、高速 ASP システム *clingo* と *asprin* を用いて解を求める (図 1 参照)。本論文では、説明の簡略化のため、各タイプが選択可能なオプション数の上下限値を 1 とする。

多目的 CAFE 問題の問題インスタンスは、CAFE 基準値を除いてアトムに変換され、ファクトとして表現される。CAFE 基準値は ASP の定数 t で表すものとし、実行時に *clingo* のオプションから値を指定する。各アトムは、表 2 のように第 2 節の各入力に対応する。

多目的 CAFE 問題の各制約は、ASP の個数制約および一貫性制約を使用して簡潔に表現される (コード 1)。また、コード 1 の 8~12 行目では、各タイプが選択可能なオプション数の上下限値、および、必須かどうかの別を考慮することで、IWR 値の和である S の上下限をより厳密に計算するように工夫がされている。これにより、 $iwr(S, G)$ に関する基礎化後のルール数を少なく抑えることができる。

多目的 CAFE 問題の選好文は、コード 2 のように定義される。目的関数である予想販売台数の最大化とオプション数の

```

1 { vp(VP,G) } :- vp_def(VP), group(G).
2 :- not vp(VP,G), require_vp(VP), group(G).
3
4 % 範囲制約
5 1 { v(V,G) : v_def(V,VP,_) } 1 :- vp(VP,G).
6
7 % 燃費制約
8 ub_vp(UB,VP) :- UB = #max { IWR,V : v_def(V,VP,IWR) },
9   vp_def(VP).
10 lb_vp(LB,VP) :- LB = #min { IWR,V : v_def(V,VP,IWR) },
11   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)
14   }.
15 iwr(S,G) :- S = #sum { IWR,V : v(V,G), v_def(V,_,IWR) },
16   LB <= S, S <= UB, lb_iwr(LB), ub_iwr(UB), group(G).
17 fe(FE,G) :- iwr(S,G), fe_map(S,FE).
18 sv(SV,G) :- iwr(S,G), sv_map(S,SV).
19 :- not 0 #sum { (FE-t)*SV,FE,SV,G : fe(FE,G), sv(SV,G) }.
20
21 % 要求制約
22 :- require_v_v(V1,V2), v(V1,G), not v(V2,G).
23 :- require_v_vp(V,VP), v(V,G), not vp(VP,G).
24 :- require_vp_v(VP,V), vp(VP,G), not v(V,G).
25 :- require_vp_vp(VP1,VP2), vp(VP1,G), not vp(VP2,G).
26
27 % 排除制約
28 :- exclude_v_v(V1,V2), v(V1,G), v(V2,G).
29 :- exclude_v_vp(V,VP), v(V,G), vp(VP,G).
30 :- exclude_vp_v(VP,V), vp(VP,G), v(V,G).
31 :- exclude_vp_vp(VP1,VP2), vp(VP1,G), vp(VP2,G).
32
33 % 初期制約
34 :- not v(V,G), require_g_v(G,V).

```

コード 1: 制約の符号化

最小化は、それぞれ選好名 `max_sv`, `min_op` として定義している。9 行目で使用している選好タイプ `pareto` は、選好文中で定義された選好名を引数に取り、それらの選好関係についてパレートな解を得ることができる。すなわち、解集合 S_1 が S_2 よりも厳密に好まれるならば、かつそのときに限り、引数のすべての選好関係に関して、少なくとも S_1 は S_2 と同等で、かつ、少なくとも 1 つの選好関係に関して S_1 は S_2 より厳密に優れている。

第 2 節と同様に多目的 CAFE 問題の例 (図 2) を用いて、最適解をすべて列挙した結果を表 3 に示す。予想販売台数とオプション数の値から、得られた各装備仕様が互いにトレードオフの関係にあることが分かる。本解法の利点は、このような定性的に得られた複数の解を人が見比べることで、定量的な判断により最終的な決定をすることができる点である。

```

1 % 予想販売台数の最大化(max_sv)
2 #preference (max_sv, more(weight)) { SV,G ::
3   sv(SV,G) }.
4
5 % オプション数の最小化(min_op)
6 used_v(V) :- v(V,G).
7 #preference (min_op, less(weight)) { 1,V ::
8   used_v(V) }.
9
10 % max_sv と min_op のパレート(all)
11 #preference (all, pareto) { **max_sv; **min_op
12   }.
13
14 % all を最適化
15 #optimize(all).

```

コード 2: 選好文

4. 実行実験

提案手法の有効性を評価するために、実行実験を行った。ベンチマーク問題 (表 4) には、企業から提供を受けた小規模 (small)・中規模 (medium)・大規模 (big) の CAFE 問題 (3 問) に対して、5 種類の CAFE 基準値 (8.5, 9.0, 9.5, 10.0, 10.5km/L) を適用した問題インスタンス (合計 15 問) を使用した。また、求めたい装備仕様の個数は 3 とした。

ASP システムには、*clingo*-5.4.0 と、*asprin*-3.1.1 を利用した。実験環境は、Mac OS(3.2GHz Intel Core i7, 64GB メモリ) である。*asprin* はすべての最適解集合を探索するように設定し、一問あたりの制限時間は 10800 秒 (3 時間) とした。

表 5 に小規模な問題での実験結果を示す。左の列から順に、問題名、CAFE 基準値、得られた最適解の個数、CPU 時間となっている。記号 ‘*’ 付きの最適解の個数は、全列挙が完了しておりこれ以上最適解が存在しないことを表す。制限時間内に最適解の全列挙が完了しなかった場合は、CPU 時間の欄に Time Out と示し、それまでに求めることができた最適解の個数を表す。また、**UNSAT** は制約を満たす実行可能解 (装備仕様) が存在しないことを意味する。

実験の結果、小規模な問題では、5 問中 4 問に対して最適解の全列挙あるいは UNSAT を得ることに成功した。しかし、中・大規模な問題では、すべての問題インスタンスに対して最適解を得ることができなかった。

5. おわりに

参考文献

- [Baral 03] Baral, C.: *Knowledge Representation, Reasoning and Declarative Problem Solving*, Cambridge University Press (2003)
- [Brewka 15] Brewka, G., Delgrande, J. P., Romero, J., and Schaub, T.: *asprin: Customizing Answer Set Preferences without a Headache*, in *AAAI*, pp. 1467–1474, AAAI Press (2015)
- [Erdem 16] Erdem, E., Gelfond, M., and Leone, N.: *Applications of ASP*, *AI Magazine*, Vol. 37, No. 3, pp. 53–68 (2016)
- [Gelfond 88] Gelfond, M. and Lifschitz, V.: *The Stable Model Semantics for Logic Programming*, in *Proceedings of the Fifth International Conference and Symposium on Logic Programming*, pp. 1070–1080, MIT Press (1988)
- [井上 08] 井上 克巳, 坂間 千秋: 論理プログラミングから解集合プログラミングへ, コンピュータソフトウェア, Vol. 25, No. 3, pp. 20–32 (2008)
- [経済 18] 経済産業省, 国土交通省: 乗用車燃費規制の現状と論点について, 第 5 回自動車燃費基準小委員会 (2018)
- [Pohl 05] Pohl, K., Böckle, G., and Linden, van der F.: *Software Product Line Engineering*, Springer (2005)

表 3: CAFE 問題 (図 2) のパレート解

		解 1			解 2			解 3			解 4		
装備仕様		1	2	3	1	2	3	1	2	3	1	2	3
装備	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	16inch	17inch	18inch	16inch	17inch	18inch	16inch	17inch	18inch	16inch	17inch	18inch
	Transmission	6AT	HEV	10AT	10AT	HEV	10AT	10AT	HEV	10AT	10AT	10AT	10AT
	Sun_Roof	-	-	-	-	-	-	-	-	-	-	-	-
予想販売台数 (合計)		5,525			5,475			5,135			4,723		
オプション数		12			11			10			9		

表 4: ベンチマーク問題

問題名	タイプ数	オプション数	要求制約数
small	8	21	4
medium	86	229	147
big	315	1,337	0

表 5: 実験結果

問題名	CAFE	最適解の数	CPU 時間 (秒)
small	8.5	8*	35.136
small	9.0	5*	1085.354
small	9.5	0	Time Out
small	10.0	1*	1.863
small	10.5	UNSAT	0.221