

IRT - SystemX

Génération de scénarios pour les véhicules autonomes

Valérie Garcin, Nicoletta Prencipe, Suzanne Schlich, Dorine Tabary

SEME, Semaine d'Études Mathématiques et Entreprises de Bordeaux

26-30 octobre 2020



Présentation du problème

- ▶ **Scénario** : ensemble de paramètres du véhicule autonome et de son environnement.
- ▶ **Objectif** : élaborer une méthode mathématique permettant de générer les 1000 scénarios les plus pertinents obéissant à de multiples nécessités :
 - ▶ règles logiques entre paramètres ;
 - ▶ pondération par criticité et probabilité.

Générer les scénarios sans travail préalable entraîne une explosion combinatoire de 2^{45} solutions.

Les données

► Criticité et probabilité :

Pour chaque critère : 2 niveaux de criticités (A et B). 5 niveaux de probabilité (A, B, C, D et E)

Choix de valeurs pour la criticité : $A = 1, B = 0,5$

Choix de valeurs de probabilité :

$A = 0.9, B = 0.7, C = 0.5, D = 0.3, E = 0.1$

► Implications :

Matrice de taille 45×45 , telle que l'entrée (i, j) vaut 1 si le paramètre j implique le paramètre i

► Exclusions :

Création d'une matrice 45×45 , telle que l'entrée (i, j) vaut 1 si les paramètres i et j s'excluent dans un scénario.

Les outils utilisables

Pour résoudre les problèmes combinatoires, il existe 2 principales heuristiques de programmation basées sur des fonctions objectif :

Programmation Linéaire

- ▶ **A partir** d'une fonction linéaire + variables
- ▶ **Contraintes = Relations** linéaires entre les variables
- ▶ Construit **la solution** (exploration d'un arbre)

Les outils utilisables

Pour résoudre les problèmes combinatoires, il existe 2 principales heuristiques de programmation basées sur des fonctions objectif :

Programmation Linéaire

- ▶ **A partir** d'une fonction linéaire + variables
- ▶ **Contraintes = Relations** linéaires entre les variables
- ▶ Construit **la solution** (exploration d'un arbre)

Programmation Par Contraintes

- ▶ **A partir** de variables de décisions et de contraintes
- ▶ **Contraintes = Relations** directes entre variables (limitant leurs valeurs)
- ▶ Filtre **les solutions** (propagation des contraintes)

Les outils utilisables

Pour résoudre les problèmes combinatoires, il existe 2 principales heuristiques de programmation basées sur des fonctions objectif :

Programmation Linéaire

- ▶ **A partir** d'une fonction linéaire + variables
- ▶ **Contraintes = Relations** linéaires entre les variables
- ▶ Construit **la solution** (exploration d'un arbre)

Programmation Par Contraintes

- ▶ **A partir** de variables de décisions et de contraintes
- ▶ **Contraintes = Relations** directes entre variables (limitant leurs valeurs)
- ▶ Filtre **les solutions** (propagation des contraintes)

Notre choix : La PPC !

Simple, rapide et efficace

Une programmation par contrainte en 2 étapes

Une programmation par contrainte en 2 étapes

- ▶ Filtrage, en amont de la génération des scénarios

Une programmation par contrainte en 2 étapes

- ▶ Filtrage, en amont de la génération des scénarios
 - * l'exclusion, $\overline{A \wedge B}$,
 - * l'implication, $A \wedge \overline{B}$.

Une programmation par contrainte en 2 étapes

- ▶ Filtrage, en amont de la génération des scénarios
 - * l'exclusion, $\overline{A \wedge B}$,
 - * l'implication, $A \wedge \overline{B}$.
- ▶ Filtrage des scénarios selon leur pertinence

Pertinence des scénarios

- ▶ On a les données suivantes :
 $P(i)$: **probabilité** du critère i
 $C(i)$: **criticité** du critère i

Pertinence des scénarios

- ▶ On a les données suivantes :
 $P(i)$: **probabilité** du critère i
 $C(i)$: **criticité** du critère i
- ▶ On définit la **probabilité** et la **criticité** d'un scénario s :

$$\text{Proba}(s) = \prod_{i \in s} P(i) \prod_{i \notin s} (1 - P(i))$$

$$\text{Crit}(s) = \frac{1}{45} \sum_{i \in s} C(i)$$

Pertinence des scénarios

- ▶ On a les données suivantes :
 $P(i)$: **probabilité** du critère i
 $C(i)$: **criticité** du critère i
- ▶ On définit la **probabilité** et la **criticité** d'un scénario s :

$$\text{Proba}(s) = \prod_{i \in s} P(i) \prod_{i \notin s} (1 - P(i))$$

$$\text{Crit}(s) = \frac{1}{45} \sum_{i \in s} C(i)$$

- ▶ On regroupe les deux dans une mesure de la **pertinence d'un scénario** en fonction d'un paramètre $t \in [0, 1]$:

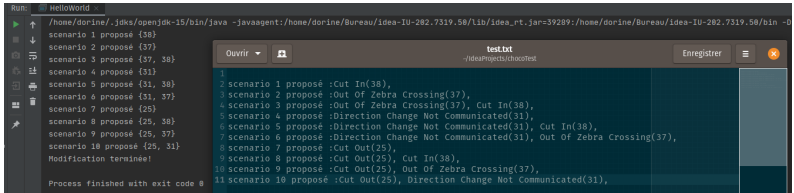
$$\pi_t(s) = t \text{Crit}(s) + (1 - t) \text{Proba}(s)$$

Formalisation du CSP (Contraint Solving Problem)

$$\begin{cases} \chi = \{\chi_i = \text{"feature"}, i \in \{1, \dots, 45\}\} \\ D = \{D_i = \{0, 1\}, i \in \{1, \dots, 45\}\} \\ \mathcal{C} = \{C_1, C_2, C_3\} \end{cases}$$

- ▶ χ est l'ensemble de variables ou paramètre du problème,
- ▶ D est l'ensemble des domaines des variables,
- ▶ \mathcal{C} est l'ensemble de contraintes

$$C_1 = \{\overline{\chi_i \wedge \chi_j}\}_{\{i,j\} \in \mathcal{E}} \quad C_2 = \{\overline{\chi_i \wedge \overline{\chi_j}}\}_{(i,j) \in \mathcal{I}} \quad C_3 = \{\pi_t(s) \geq \alpha\}$$



```
Run: HelloWorld
/home/dorine/.jdk/openjdk-15/bin/java -javaagent:/home/dorine/Bureau/idea-IU-202.7319.58/lib/idea_rt.jar=39289:/home/dorine/Bureau/idea-IU-202.7319.58/bin -D
scenario 1 proposé {38}
scenario 2 proposé {37}
scenario 3 proposé {37, 38}
scenario 4 proposé {31}
scenario 5 proposé {31, 38}
scenario 6 proposé {31, 37}
scenario 7 proposé {25}
scenario 8 proposé {25, 38}
scenario 9 proposé {25, 37}
scenario 10 proposé {25, 31}
Modification terminée!
Process finished with exit code 0

test.txt
~/IdeaProjects/chocoTest

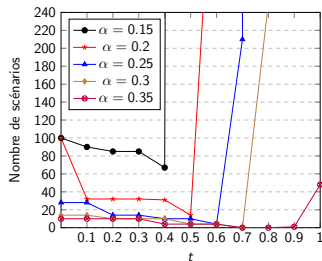
1
2 scenario 1 proposé :Cut In(38),
3 scenario 2 proposé :Out Of Zebra Crossing(37),
4 scenario 3 proposé :Out Of Zebra Crossing(37), Cut In(38),
5 scenario 4 proposé :Direction Change Not Communicated(31),
6 scenario 5 proposé :Direction Change Not Communicated(31), Cut In(38),
7 scenario 6 proposé :Direction Change Not Communicated(31), Out Of Zebra Crossing(37),
8 scenario 7 proposé :Cut Out(25),
9 scenario 8 proposé :Cut Out(25), Cut In(38),
10 scenario 9 proposé :Cut Out(25), Out Of Zebra Crossing(37),
11 scenario 10 proposé :Cut Out(25), Direction Change Not Communicated(31),
```

FIGURE – Sortie Obtenue.

α/t	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.05	704	x	x	x	x	x	x	x	x	x	x
0.075	318	33102	29211	x	x	x	x	x	x	x	x
0.1	225	240	240	21450	x	x	x	x	x	x	x
0.125	169	169	169	169	15468	x	x	x	x	x	x
0.15	100	90	85	85	67	9564	x	x	x	x	x
0.175	100	85	85	32	31	65	9397	x	x	x	x
0.2	100	32	32	32	31	14	490	1873	x	x	x
0.225	30	29	28	14	14	10	10	580	12993	x	x
0.25	28	28	14	14	10	10	4	210	5127	x	x
0.275	28	14	14	10	10	10	4	0	698	40273	x
0.3	14	14	10	10	10	4	4	0	250	6643	x
0.325	14	10	10	10	4	4	4	0	0	201	4899
0.35	10	10	10	10	4	4	4	0	0	1	48
0.4	10	10	10	4	4	0	0	0	0	0	0
0.45	10	4	4	4	0	0	0	0	0	0	0
0.5	4	4	4	4	0	0	0	0	0	0	0
0.55	4	4	4	0	0	0	0	0	0	0	0
0.6	4	4	0	0	0	0	0	0	0	0	0
0.65	4	0	0	0	0	0	0	0	0	0	0
0.7	0	0	0	0	0	0	0	0	0	0	0

Génération de scénarios

Le code fournit des scénarios selon deux variables : t et α .

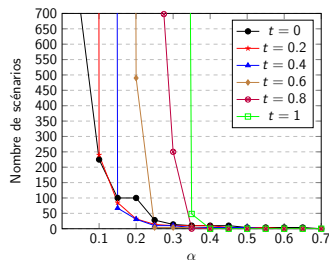
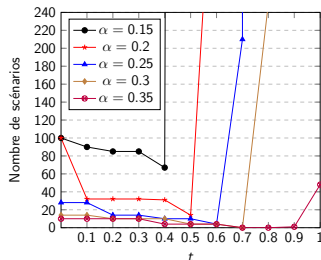


Observations :

- Augmentation du nombre de scénarios selon t

Génération de scénarios

Le code fournit des scénarios selon deux variables : t et α .



Observations :

- Augmentation du nombre de scénarios selon t

Observations :

- Diminution du nombre de scénarios selon α

Enrichissement en terme de données

- ▶ Matrices des implications et des exclusions plus fournies possiblement validées par un expert en sécurité routière.
- ▶ Regroupement des paramètres par typologie (météo, voirie, véhicule autonome, état du conducteur, ...).
- ▶ Valeurs précises pour les probabilités/criticités à la place de plages de valeurs.

Autres pistes possibles :

- ▶ Choix différent de la fonction π implémentant un choix différent quant à la pertinence.
- ▶ Matrice de covariance entre les features pour une approche probabiliste.
- ▶ Ajouter comme paramètre le niveau d'autonomie du véhicule.

Merci pour votre attention !



Des questions ?