

# Bomb Busters

## Modélisation logique et encodage SAT

Côme Perin

January 2026

## 1 Introduction

On cherche à modéliser l'ensemble des états possibles d'une partie du jeu *Bomb Busters* compatibles avec les informations publiques disponibles à un instant donné.

La modélisation prend la forme d'un ensemble de contraintes portant sur la couleur et la valeur de chaque carte. Elle est destinée à être encodée en SAT.

La dynamique du jeu (succession des tours, historique des actions) n'est pas modélisée explicitement : toute information publique est traduite directement en contrainte sur l'état courant.

## 2 Données et notations

Cette section fixe les paramètres, les domaines de valeurs et les variables abstraites utilisées dans tout le document.

### 2.1 Paramètres globaux

Soit  $p \in \mathbb{N}^*$  le nombre de joueurs.

Soit  $N \in \mathbb{N}$  un paramètre représentant la valeur maximale possible des cartes bleues, donc les valeurs bleues appartiennent à  $\{1, \dots, N\}$ .

Chaque valeur bleue apparaît exactement 4 fois. On note alors :

$$n_{\text{bleu}} = 4(N + 1)$$

le nombre total de cartes bleues.

On note :

- $n_{\text{jaune}}$  le nombre de cartes jaunes, supposé pair,
- $n_{\text{rouge}}$  le nombre de cartes rouges.

### 2.2 Listes de valeurs jaunes et rouges

Les cartes jaunes et rouges possèdent des valeurs connues globalement, mais leur répartition entre les joueurs est inconnue.

On note :

$$(y_j)_{j=1}^{n_{\text{jaune}}} \quad \text{les valeurs des cartes jaunes}, \quad (r_j)_{j=1}^{n_{\text{rouge}}} \quad \text{les valeurs des cartes rouges.}$$

Ces listes peuvent contenir des doublons. Les valeurs jaunes et rouges sont distinctes des valeurs bleues.

On définit les fonctions de multiplicité, pour toute valeur  $\alpha$  :

$$m_{\text{jaune}}(\alpha) = |\{ j \in \{1, \dots, n_{\text{jaune}}\} \mid y_j = \alpha \}|, \quad m_{\text{rouge}}(\alpha) = |\{ j \in \{1, \dots, n_{\text{rouge}}\} \mid r_j = \alpha \}|.$$

## 2.3 Répartition des cartes entre joueurs

Pour chaque joueur  $i \in \{1, \dots, p\}$ , on note  $N_i$  le nombre de cartes qu'il possède. La contrainte globale de répartition est :

$$\sum_{i=1}^p N_i = n_{\text{bleu}} + n_{\text{jaune}} + n_{\text{rouge}}.$$

Les cartes d'un même joueur sont supposées ordonnées selon l'ordre croissant des valeurs.

## 2.4 Variables abstraites du modèle

Pour chaque joueur  $i$  et chaque position  $t \in \{1, \dots, N_i\}$  :

- $X_i(t)$  désigne la valeur de la carte en position  $t$ ,
- $C_i(t) \in \{\text{bleu, jaune, rouge}\}$  désigne sa couleur.

On note l'ensemble des positions de cartes :

$$\mathcal{P} = \{(i, t) \mid i \in \{1, \dots, p\}, t \in \{1, \dots, N_i\}\}.$$

On note l'ensemble des valeurs possibles :

$$\mathcal{V} = \{1, \dots, N\} \cup \{y_j\}_{j=1}^{n_{\text{jaune}}} \cup \{r_j\}_{j=1}^{n_{\text{rouge}}}.$$

## 3 Contraintes au niveau du modèle logique

Cette section regroupe les contraintes formulées sur les variables  $X_i(t)$  et  $C_i(t)$ . Elles décrivent exactement l'espace des états du jeu compatibles avec les informations disponibles.

### 3.1 Contraintes locales

#### 3.1.1 Exclusivité des couleurs

Chaque carte possède exactement une couleur :

$$\forall (i, t) \in \mathcal{P}, \quad (C_i(t) = \text{bleu}) \oplus (C_i(t) = \text{jaune}) \oplus (C_i(t) = \text{rouge}).$$

### 3.1.2 Domaine des valeurs

Toute carte prend une valeur appartenant à  $\mathcal{V}$  :

$$\forall (i, t) \in \mathcal{P}, \quad X_i(t) \in \mathcal{V}.$$

### 3.1.3 Compatibilité couleur-valeur

$$\begin{aligned} \forall (i, t) \in \mathcal{P}, \quad C_i(t) = \text{bleu} &\Rightarrow X_i(t) \in \{0, \dots, N\}, \\ \forall (i, t) \in \mathcal{P}, \quad C_i(t) = \text{jaune} &\Rightarrow X_i(t) \in \{y_j\}_{j=1}^{n_{\text{jaune}}}, \\ \forall (i, t) \in \mathcal{P}, \quad C_i(t) = \text{rouge} &\Rightarrow X_i(t) \in \{r_j\}_{j=1}^{n_{\text{rouge}}}. \end{aligned}$$

## 3.2 Contraintes structurelles

### 3.2.1 Ordre des cartes

Pour chaque joueur :

$$\forall i, \forall t < N_i, \quad X_i(t) \leq X_i(t+1).$$

## 3.3 Contraintes globales

### 3.3.1 Multiplicités

Valeurs bleues :

$$\forall v \in \{1, \dots, N\}, \quad |\{(i, t) \in \mathcal{P} \mid C_i(t) = \text{bleu} \wedge X_i(t) = v\}| = 4.$$

Valeurs jaunes et rouges :

$$\begin{aligned} \forall \alpha, \quad |\{(i, t) \in \mathcal{P} \mid C_i(t) = \text{jaune} \wedge X_i(t) = \alpha\}| &= m_{\text{jaune}}(\alpha), \\ \forall \alpha, \quad |\{(i, t) \in \mathcal{P} \mid C_i(t) = \text{rouge} \wedge X_i(t) = \alpha\}| &= m_{\text{rouge}}(\alpha). \end{aligned}$$

En particulier :

$$|\{(i, t) \in \mathcal{P} \mid C_i(t) = \text{jaune}\}| = n_{\text{jaune}}, \quad |\{(i, t) \in \mathcal{P} \mid C_i(t) = \text{rouge}\}| = n_{\text{rouge}}.$$

## 3.4 Contraintes d'observation

Les informations publiques disponibles en cours de partie sont traduites en contraintes sur l'état courant.

### 3.4.1 Observations complètes

Soit  $\mathcal{O} \subset \mathcal{P}$  l'ensemble des positions dont la valeur et la couleur sont connues. Pour chaque  $(i, t) \in \mathcal{O}$ , on note :

$$\hat{X}_{i,t} \text{ la valeur observée, } \quad \hat{C}_{i,t} \text{ la couleur observée.}$$

On impose :

$$\forall (i, t) \in \mathcal{O}, \quad X_i(t) = \hat{X}_{i,t} \wedge C_i(t) = \hat{C}_{i,t}.$$

Si l'on suppose qu'aucune carte rouge n'a été coupée :

$$\forall (i, t) \in \mathcal{O}, \quad \hat{C}_{i,t} \neq \text{rouge}.$$

### 3.4.2 Observations partielles

Soit  $\mathcal{O}_y \subset \mathcal{P}$  l'ensemble des positions connues pour être jaunes. On impose :

$$\forall (i, t) \in \mathcal{O}_y, \quad C_i(t) = \text{jaune}.$$

### 3.4.3 Cas du joueur observateur

Les cartes du joueur observateur appartiennent à  $\mathcal{O}$  : leur valeur et leur couleur sont entièrement connues et intégrées via les contraintes précédentes.

## 4 Encodage SAT

Cette section décrit la traduction des contraintes de la section précédente en un système de clauses propositionnelles en CNF.

Afin d'éviter les répétitions, on procède ainsi :

- on définit d'abord les variables SAT,
- on donne ensuite des schémas de clauses réutilisables (exactement-une, compatibilité, ordre),
- on renvoie vers ces schémas dans les sous-sections suivantes.

### 4.1 Variables SAT

#### 4.1.1 Variables de couleur

Pour chaque  $(i, t) \in \mathcal{P}$ , on introduit :

$$b_{i,t}, \quad j_{i,t}, \quad r_{i,t},$$

qui représentent respectivement  $C_i(t) = \text{bleu}$ ,  $C_i(t) = \text{jaune}$  et  $C_i(t) = \text{rouge}$ .

#### 4.1.2 Variables de valeur

Pour chaque  $(i, t) \in \mathcal{P}$  et  $v \in \mathcal{V}$ , on introduit :

$$x_{i,t,v},$$

avec  $x_{i,t,v} = \text{vrai} \iff X_i(t) = v$ .

## 4.2 Schémas de clauses

### 4.2.1 Contraintes exactement-une

Pour un ensemble fini de variables booléennes  $(z_k)_{k \in K}$ , la contrainte *exactement-une* est définie par :

$$\text{EO}(z_k)_{k \in K} \triangleq \left( \bigvee_{k \in K} z_k \right) \wedge \left( \bigwedge_{\substack{k, k' \in K \\ k \neq k'}} (\neg z_k \vee \neg z_{k'}) \right).$$

**Couleur.** Pour tout  $(i, t) \in \mathcal{P}$  :

$$\text{EO}(b_{i,t}, j_{i,t}, r_{i,t}).$$

**Valeur.** Pour tout  $(i, t) \in \mathcal{P}$  :

$$\text{EO}(x_{i,t,v})_{v \in \mathcal{V}}.$$

### 4.2.2 Compatibilité couleur–valeur

Ces contraintes traduisent la section correspondante du modèle logique. Pour tout  $(i, t) \in \mathcal{P}$  :

$$\begin{aligned} & (\neg b_{i,t} \vee \bigvee_{v=0}^N x_{i,t,v}), \\ & (\neg j_{i,t} \vee \bigvee_{k=1}^{n_{jaune}} x_{i,t,y_k}), \\ & (\neg r_{i,t} \vee \bigvee_{k=1}^{n_{rouge}} x_{i,t,r_k}). \end{aligned}$$

### 4.2.3 Ordre des cartes

Pour tout joueur  $i$  et tout  $t \in \{1, \dots, N_i - 1\}$ , on impose :

$$\bigwedge_{\substack{v, v' \in \mathcal{V} \\ v > v'}} (\neg x_{i,t,v} \vee \neg x_{i,t+1,v'}).$$

## 4.3 Contraintes globales de multiplicité

Les contraintes de multiplicité correspondent à des contraintes de cardinalité. Pour une valeur donnée  $v$ , imposer qu'elle apparaisse exactement  $k$  fois s'écrit :

$$\sum_{(i,t) \in \mathcal{P}} x_{i,t,v} = k.$$

À titre illustratif, la contrainte  $x_1 + x_2 + x_3 = 2$  peut se traduire en CNF, mais la traduction devient rapidement volumineuse et fastidieuse lorsque le nombre de variables augmente. On

conserve donc dans ce document la notation de cardinalité, en supposant l'usage d'un encodeur standard lors de la génération effective de la CNF.

On impose alors :

$$\begin{aligned} \forall v \in \{0, \dots, N\}, \quad & \sum_{(i,t) \in \mathcal{P}} x_{i,t,v} = 4, \\ \forall \alpha, \quad & \sum_{(i,t) \in \mathcal{P}} x_{i,t,\alpha} = m_{\text{jaune}}(\alpha), \quad \forall \alpha, \quad \sum_{(i,t) \in \mathcal{P}} x_{i,t,\alpha} = m_{\text{rouge}}(\alpha). \end{aligned}$$

#### 4.4 Résumé de l'encodage

L'encodage SAT final est la conjonction des familles de clauses suivantes :

- contraintes exactement-une (Section 4.2.1),
- compatibilité couleur–valeur (Section 4.2.2),
- ordre des cartes (Section 4.2.3),
- multiplicité (Section 4.3),
- observations, traduites en clauses unitaires ou petites conjonctions, selon qu'elles fixent une valeur, une couleur, ou les deux.