# Sur la complexité algorithmique des problèmes de satisfaction de contraintes disjonctifs

**Miki Hermann** **Florian Richoux**
LIX (CNRS, UMR 7161), Ecole Polytechnique
91128 Palaiseau, France
{miki.hermann, florian.richoux}@lix.polytechnique.fr

**Résumé**

Les problèmes de satisfaction de contraintes (CSP) constituent une puissante manière de capturer de nombreux problèmes combinatoires. Le CSP général est connu pour être NP-complet, mais la complexité du problème paramétré CSP($S$) dépend uniquement de son paramètre, habituellement un ensemble de relations sur lesquelles les contraintes sont construites. Suivant ce paramètre, il existe des instances de CSP "faciles" et "difficiles". Dans cet article, nous montrons un théorème dichotomique pour tous domaines finis de CSP où la disjonction entre contraintes est autorisée. Cette dichotomie est basée sur un critère simple, nous permettant de classer les CSP disjonctifs comme étant dans P ou étant NP-complet. Nous prouvons également que le méta-problème, à savoir vérifier le critère de dichotomie pour les problèmes de satisfaction de contraintes disjonctifs, est fixed-parameter tractable. De plus, nous présentons un algorithme en temps polynomial répondant à cette question pour les CSP disjonctifs sur un domaine ternaire.

Mots clés : Complexité algorithmique, problèmes de satisfaction de contraintes disjonctifs, théorème dichotomique.

## 1 Introduction

Les problèmes de satisfaction de contraintes (CSP) se présentent comme un formalisme pratique pour décrire de nombreux problèmes algorithmiques provenant de la combinatoire et de la théorie de graphes, de l'intelligence artificielle, de la bio-informatique, etc (voir [3] pour une récente étude). Ces problèmes requièrent une analyse quantitative étudiant leur complexité algorithmique. Le but de l'étude de la complexité des problèmes de satisfaction de contraintes est de reconnaître les critères permettant de distinguer les cas faciles des cas difficiles. Cette étude a été commencée par Schaefer dans son article pionnier [11], où il caractérise complètement par son théorème dichotomique la complexité des CSP booléens, distinguant les instances comme étant polynomiales ou NP-complètes. Feder et Vardi [5] ont étendu cette étude aux problèmes de satisfaction de contraintes sur les domaines de cardinalité fini et ont conjecturé l'existence d'un théorème dichotomique pour tous domaines finis. Jusqu'ici, ceci n'a été prouvé que pour les domaines ternaires par Bulatov [1], alors que la question demeure ouverte pour les domaines de cardinalité supérieure.

La principale difficulté de résoudre cette conjecture réside dans le fait de l'inexistence d'ensemble fini ou dénombrable de fonctions, appelé clone, par lesquelles les relations définissant un CSP sont closes. Cette clôture par les fonctions d'un clone donné nous fournit les critères requis pour caractériser la complexité du CSP. Dans le cas booléen, ces clones, organisés par inclusion dans le treillis de Post [9], sont en nombre infini dénombrable. De plus, les cas déterminant les instances faciles et NP-complètes apparaissent dans la partie finie de ce treillis. La caractérisation de chaque clone peut toujours être exprimée par un ensemble fini de fonctions, appelé base. Ces faits sont largement exploités par Schaefer dans [11]. Cependant, le treillis correspondant aux domaines de cardinalité 3 et plus est indénombrable, et il y existe des clones sans base ou avec une base infinie, comme l'ont prouvé Yanov and Muchnik [12]. Ainsi, ces treillis pour les domaines de cardinalité supérieure restent largement inconnus, rendant un critère dichotomique difficile à trouver.

Une façon de contourner cette difficulté est d'introduire plus de structure dans les problèmes de satisfaction de contraintes, espérant ainsi trouver un nombre

fini de cas seulement à traiter. Pour introduire plus de structure, nous avons choisit d'étendre les problèmes de satisfaction de contraintes en autorisant la disjonction de contraintes. Les CSP disjonctifs ont été aussi étudiés d'un point de vue différent par Cohen, Jeavons, Jonsson et Koubarakis dans [2]. Considérer des CSP disjonctifs peut sembler trop restrictifs. Cependant ceci est incorrect puisqu'ils représentent les CSP sur les algèbres faibles de Krasner, une structure bien connue en algèbre universelle [7, 8]. L'avantage des algèbres faibles de Krasner est qu'ils sont clos uniquement par des endomorphismes.

Puisqu'il existe un nombre fini seulement d'endomorphismes sur un domaine fini donné, nous sommes assurés d'obtenir une caractérisation finie de la complexité des CSP disjonctifs.

Dans cet article, nous étudions la complexité des problèmes de satisfactions de contraintes disjonctifs, également considérés comme les CSP sur des algèbres faibles de Krasner. Nous en dérivons une caractérisation complète de leur complexité à travers un théorème dichotomique pour chaque domaine de taille finie. Une fois le critère de dichotomie énoncé, nous étudierons la complexité du méta-problème, *i.e.*, la complexité de décider si un CSP disjonctif donné satisfait le critère de dichotomie. Cette analyse est tout d'abord faite pour le cas général d'un domaine de n'importe quelle cardinalité finie. Cependant, puisque le domaine ternaire présente un comportement particulier, nous présentons une nouvelle analyse de complexité pour le méta-problème, obtenant ainsi un résultat plus poussé et précis que pour le cas général.

## 2 Notions de base

Tout au long de cet article, une fonction $f$ sera une fonction unaire surjective $f\colon D \to D'$ avec $D$ un domaine de taille fixée et $D'$ un sous-ensemble de $D$. *L'image* de $f$, noté $\operatorname{ran} f$, est l'ensemble $\{f(x) \mid x \in D\}$. Nous écrivons $f(A)$ pour un sous-ensemble $A \subseteq D$ afin de désigner l'ensemble $\{f(a) \mid a \in A\}$.

L'étude des problèmes de satisfaction de contraintes utilise souvent les notions de *relations*, de *clones* et de *co-clones*. Une **relation** $n$-aire $R$ sur un domaine $D$ est un ensemble de tuples $t$ dans $D^n$. Nous notons la $i$-ème coordonnée de $t$ par $t[i]$, tel que pour chaque tuple $n$-aire $t$, nous écrivons $t = (t[1], t[2], \ldots, t[n])$. Nous utiliserons comme raccourci $(t[i : j])$ désignant les éléments $(t[i], t[i+1], \ldots, t[j])$ d'un tuple $t$. Nous notons également par $S$ un ensemble de relations $R \subseteq D^{\operatorname{ar}(R)}$ où $\operatorname{ar}(R)$ est l'arité de $R$.

Nous disons que la relation $R$ est *$d$-valide* s'il existe $d \in D$ tel que le tuple $(d, d, \ldots, d)$ appartient à $R$. Par extension, nous disons qu'un ensemble $S$ de relations est $d$-valide s'il existe $d \in D$ tel que, pour chaque relation $R$ dans $S$, $R$ est $d$-valide.

Un **clone**, ou un **clone fonctionnel**, est un ensemble de fonctions contenant l'identité $id$ et qui est clos par composition. Le plus petit clone contenant un ensemble de fonctions $F$ est noté $[F]$.

**Exemple 2.1** *Soit $F = \{f, g\}$ un ensemble de fonctions. Alors $[F]$ est tel que*

$$[F] = \{id, f, g, ff, fg, gf, gg, fff, ffg, \ldots\}$$

■

Dans notre étude, un **co-clone**, ou un **clone relationnel**, est un ensemble de relations contenant l'égalité $eq = \{(d, d) \mid d \in D\}$ et qui est clos par les opérations suivantes : (1) la conjonction de relations, *i.e.* le produit cartésien $R_1 \times R_2$. Soit $R_1$ et $R_2$ deux relations d'arité $n$ et $m$, respectivement. Alors $R = R_1 \times R_2$ est une relation d'arité $n+m$ définie par $R = \{(t[1 : n+m])\}$ tel que $(t[1 : n]) \in R_1$ et $(t[n+1 : n+m]) \in R_2$ ; (2) la disjonction de relations. Soit $R_1$ et $R_2$ deux relations d'arité $n$ et $m$, respectivement. Alors $R = R_1 \vee R_2$ est une relation d'arité $n+m$ définie par $R = \{(t[1 : n], d[n+1 : n+m])\} \cup \{(d[1 : n], t[n+1 : n+m])\}$ tel que $(t[1 : n]) \in R_1$, $(t[n+1 : n+m]) \in R_2$, $(d[1 : n]) \in D^n$ et $(d[n+1 : n+m]) \in D^m$ ; (3) l'identification de variables $\nu_{i,j}$. Soit $R_1$ une relation $n$-aire. Alors $R = \nu_{i,j}(R_1)$ est une relation $n$-aire définie par $R = \{t' \mid t \in R_1,\ t'[k] = t[k]$ *avec* $i \neq k \neq j, t'[i] = t'[j] = t[i]\}$ ; (4) la quantification existentielle, *i.e.* la projection $\pi_{x_1, x_2, \ldots, x_k}$. Soit $R_1$ une relation $n$-aire. Alors $R = \pi_{x_1, x_2, \ldots, x_k}(R_1)$ est une relation d'arité $n-k$ définie par $R = \{(t[x_1 : x_k]) \mid t \in R_1\}$. Remarquez qu'une structure vérifiant les conditions numéros 1, 3 et 4 est appelée une **algèbre relationnelle**. Une structure vérifiant chacune des quatre conditions est appelée une **algèbre faible de Krasner**. Ainsi tout au long de cet article, un co-clone sera une algèbre faible de Krasner. Le plus petit co-clone contenant un ensemble $S$ de relations est noté $\langle S \rangle$.

L'algèbre universelle utilise souvent la notion de *kernel*. Nous définissons cette notion à l'aide des classes d'équivalence. Par soucis de clarté et de simplicité, nous excluons les éléments réflexifs $(d, d)$ pour chaque $d \in D$ puisqu'ils n'enrichissent pas ici la notion de kernel. Soit $f\colon D \to D$ une fonction d'image $\operatorname{ran} f$. Une *$d$-classe d'équivalence*, pour $d \in \operatorname{ran} f$, est l'ensemble des éléments $[d]_f = \{x \in D \mid f(x) = d\}$. Le **kernel** d'une fonction $f$, noté $\ker f$, est l'ensemble de toutes les classes d'équivalence d'une cardinalité strictement supérieure à 1 pour tout $d \in \operatorname{ran} f$, *i.e.*

$$\ker f \quad = \quad \{[d]_f \mid d \in \operatorname{ran} f \text{ and } |[d]_f| \geq 2\}.$$

**Exemple 2.2** *Soit $f$, $g$ et $h$ des fonctions sur le domaine $D = \{0, 1, 2, 3\}$ tel que*

| $x$ | $f(x)$ | $g(x)$ | $h(x)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 0 | 1 |
| 3 | 3 | 1 | 1 |

*Alors* $\ker f = \emptyset$, $\ker g = \{(0, 2); (1, 3)\}$ *et* $\ker h = \{(0, 1, 2, 3)\}$. ■

Nous avons également besoin de la notion de profondeur d'une fonction afin de décrire la complexité algorithmique d'un clone $[F]$. La *profondeur* d'une fonction $f$ *par rapport à* un ensemble $F$, notée $\mathrm{Depth}(F, f)$, est la longueur de la plus courte composition de fonctions $f_i \in F$ requise pour obtenir $f$. La *profondeur* d'une fonction $f$, notée $\mathrm{Depth}(f)$, est définie par $\mathrm{Depth}(f) = \max(\mathrm{Depth}(F, f))$ pour tout ensemble $F$ tel que $f \in [F]$. Ainsi, si nous souhaitons obtenir une fonction $f$ à partir d'un ensemble $F$, il est inutile de faire des compositions de fonctions plus longue que $\mathrm{Depth}(f)$. Si toutes les combinaisons de longueur inférieure ou égale à $\mathrm{Depth}(f)$ ne nous permettent pas d'obtenir la fonction $f$ recherchée, cela signifie que $f$ n'appartient pas au clone généré par $F$.

## 3 Problèmes de satisfaction de contraintes disjonctifs

Les problèmes de satisfaction de contraintes sont souvent présentés comme des conjonctions de contraintes. Le problème paramétrique $\mathrm{CSP}(S)$, avec $S$ un ensemble de relations, est un CSP où les contraintes sont construites sur les relations de $S$. Les CSP sont connus pour être NP-complet en général, mais la complexité des problèmes paramétriques $\mathrm{CSP}(S)$ dépend uniquement que de l'ensemble de relations $S$. Schaefer [11] a donné un théorème dichotomique concernant les contraintes booléennes. Si $S$ vérifie l'une des six conditions présentées par Schaefer, alors $\mathrm{CSP}(S)$ est polynomial. Autrement il est NP-complet.

Nous centrerons notre intérêt sur la complexité des problèmes paramétriques $\mathrm{DCSP}(S)$ où DCSP est une généralisation de CSP définie ainsi : les contraintes atomiques jouent le rôle des littéraux et les contraintes sont construites par des conjonctions et des disjonctions de contraintes atomiques.

Dans cet article, nous présentons une dichotomie aux conditions simples pour le problème $\mathrm{DCSP}(S)$ avec $S$ un ensemble de relations sur $D$. Dans cette perspective, nous définissons le problème $\mathrm{DCSP}(S)$ comme ci-dessous.

Soit $R \subseteq D^k$ une relation sur $D$. Un *littéral* est une contrainte $R(x_1, \ldots, x_k)$ formée par la relation $R$ et les variables $x_1$, …, $x_k$, où $\mathrm{ar}(R) = k$. Une *formule disjonctive de contraintes* est définie inductivement de la manière suivante : (1) VRAI et FAUX, respectivement noté $\top$ et $\bot$, sont des formules disjonctives de contraintes ; (2) un littéral $l$ est une formule disjonctive de contraintes ; (3) si $\varphi_1$ et $\varphi_2$ sont des formules disjonctives de contraintes alors $\varphi_1 \vee \varphi_2$ et $\varphi_1 \wedge \varphi_2$ le sont également ; (4) si $\varphi$ est une formule disjonctive de contraintes contenant une variable libre $x$ alors $\exists x\ \varphi$ est une formule disjonctive de contraintes ; (5) enfin, la relation d'égalité $eq = \{(a, a) \mid a \in D\}$ est incluse dans chaque ensemble $S$ de relations. Nous ne considérons pas la négation dans nos formules disjonctives de contraintes car cela amènerait à une généralisation triviale du problème $\mathrm{CSP}(S)$ que est NP-complète pour tout domaine fini de cardinalité supérieure ou égale à trois, quelque soit l'ensemble $S$. Voir Pöschel [8] et le troisième paragraphe de la preuve de la proposition 4.6 de cet article.

Une *interprétation* $I : V \to D$ satisfait le littéral $R(x_1, \ldots, x_k)$, noté $I \vDash R(x_1, \ldots, x_k)$, si $(I(x_1), \ldots, I(x_k)) \in R$. Cette notion est étendue aux formules disjonctives de contraintes de la manière suivante :

- $I \vDash R_1(\vec{x}) \vee R_2(\vec{x})$ si $I \vDash R_1(\vec{x})$ ou $I \vDash R_2(\vec{x})$,
- $I \vDash R_1(\vec{x}) \wedge R_2(\vec{x})$ si $I \vDash R_1(\vec{x})$ et $I \vDash R_2(\vec{x})$.

Remarquons que pour toute formule disjonctive de contraintes $\varphi$, nous avons $\top \vDash \varphi$ et $\bot \nvDash \varphi$.

Nous pouvons maintenant, à travers les formules disjonctives de contraintes $\varphi$, définir les **problèmes de satisfaction de contraintes disjonctifs**.

**Problème :** $\mathrm{DCSP}(S)$
*Entrée :* Une formule disjonctive de contraintes $\varphi(\vec{x})$ définie sur $S$.
*Question :* La formule $\varphi(\vec{x})$ est-elle satisfaisable ?

Notre étude des CSP disjonctifs est rendue plus facile par l'existence d'une correspondance de Galois entre les clones et les co-clones. Avant de définir ce qu'est une correspondance de Galois, nous avons besoin des notions suivantes.

Soit $R$ une relation et $f$ une fonction d'arité $\mathrm{ar}(f) = k$. Nous disons que $f$ est un polymorphisme de $R$ si $R$ est clos par $f$ appliquée composant par composant, comme l'on peut le voir dans la figure 1.

L'ensemble des polymorphismes d'une relation $R$ est noté $\mathrm{Pol}\, R$. Par une légère surcharge de notation, l'ensemble des polymorphismes d'un ensemble $S$ de relations est également noté $\mathrm{Pol}\, S$, définis par $\mathrm{Pol}\, S = \bigcap_{R \in S} \mathrm{Pol}\, R$. Nous disons que $R$ est un invariant de $f$ si $f$ est un polymorphisme de $R$, et nous

$$\begin{array}{cccccc} & t_1 & t_2 & \dots & t_n & & t \\ & \shortparallel & \shortparallel & & \shortparallel & & \shortparallel \\ \mathrm{f}( & a_1^1 & a_1^2 & \dots & a_1^n & ) = & a_1 \\ \mathrm{f}( & a_2^1 & a_2^2 & \dots & a_2^n & ) = & a_2 \\ & \vdots & \vdots & & \vdots & & \vdots \\ \mathrm{f}( & a_k^1 & a_k^2 & \dots & a_k^n & ) = & a_k \end{array}$$

FIG. 1 – Si $t_1, \dots, t_n \in R$ alors $t \in R$.

notons $\operatorname{Inv} f$ l'ensemble des invariants de $f$ et $\operatorname{Inv} F$ l'ensemble des invariants de l'ensemble $F$ de fonctions.

Dans l'article de Pöschel [8], nous pouvons voir qu'utiliser la disjonction entre contraintes implique le fait que les relations satisfaisant ces formules disjonctives sont closes par des fonctions unaires uniquement. Les polymorphismes de ces relations sont ainsi des endomorphismes. Nous les noterons End au lieu de Pol. De plus, Pöschel [8] spécifie que les opérateurs End et Inv constituent une correspondance de Galois où $\operatorname{End} S$ est l'ensemble des endomorphismes $E$ de $S$, tel que chaque relation $R \in S$ est close par chaque $e \in E$.

Notons respectivement par $A$ et $B$ les ensembles de relations et de fonctions. L'étude de Pöschel [8] précise que les applications $\operatorname{End}\colon A \to B$ et $\operatorname{Inv}\colon B \to A$ présentent une correspondance de Galois entre les structures ordonnées $(A, \subseteq)$ et $(B, \subseteq)$, donc ils vérifient les conditions suivantes :

(i) pour tout $a, b \in A$ tels que $a \subseteq b$, nous avons $\operatorname{End} a \supseteq \operatorname{End} b$.

(ii) pour tout $c, d \in B$ tels que $c \subseteq d$, nous avons $\operatorname{Inv} c \supseteq \operatorname{Inv} d$.

(iii) pour tout $a \in A$ et $c \in B$, nous avons $a \subseteq \operatorname{Inv}\operatorname{End} a$ et $c \subseteq \operatorname{End}\operatorname{Inv} c$.

De plus, pour toutes les relations $S$ et les fonctions $F$, on a $\operatorname{Inv}\operatorname{End} S = \langle S \rangle$ et $\operatorname{End}\operatorname{Inv} F = [F]$. Pöschel souligne qu'en fait, les ensembles $\langle S \rangle$ et $[F]$ sont respectivement une algèbre faible de Krasner et un **monoïde d'endomorphismes**.

L'existence de la correspondance de Galois mentionnée ci-dessus nous permet de décider facilement de la complexité des CSP disjonctifs. L'analyse de la complexité est basée sur le résultat suivant.

**Proposition 3.1** *Soit $S_1$ et $S_2$ deux ensembles de relations sur le domaine $D$. L'inclusion* $\operatorname{End} S_1 \subseteq \operatorname{End} S_2$ *implique* $\mathrm{DCSP}(S_2) \leq_p \mathrm{DCSP}(S_1)$.

*Preuve :* De $\operatorname{End} S_1 \subseteq \operatorname{End} S_2$, nous en déduisons $\langle S_1 \rangle = \operatorname{Inv}\operatorname{End} S_1 \supseteq \operatorname{Inv}\operatorname{End} S_2 = \langle S_2 \rangle$. Puisque chaque co-clone (équivalent à une algèbre faible de Krasner) contient l'égalité $eq$ et est close par conjonction, disjonction, identification de variables et quantification existentielle, nous en concluons immédiatement la réduction polynomiale $\mathrm{DCSP}(S_2 \cup \{eq\}) \leq_p \mathrm{DCSP}(S_1 \cup \{eq\})$. Comme défini précédemment, l'ensemble $S$ de relations doit inclure la relation d'égalité $eq$ si $S$ est une entrée d'un problème DCSP. Ainsi, nous en déduisons la réduction $\mathrm{DCSP}(S_2) \leq_p \mathrm{DCSP}(S_1)$, terminant la preuve. □

Selon la proposition 3.1, un résultat de complexité prouvé pour un ensemble de relations $S$ s'étend immédiatement à toute relation du co-clone $\langle S \rangle$.

La présence de la relation quaternaire disjonctive

$$J \quad = \quad \{(x, y, z, w) \in D^4 \mid (x = y) \vee (z = w)\}$$

dans un CSP classique réduit ce dernier à un problème DCSP . Cette observation provient du résultat suivant.

**Proposition 3.2** *Soit $S$ une ensemble de relations sur le domaine $D$. Si $J \in S$ alors* $\operatorname{Pol} S = \operatorname{End} S$.

*Preuve :* Supposons qu'il existe une fonction binaire $g \in \operatorname{Pol} S$ dépendant de deux arguments, *i.e.*, il existe les valeurs $a_0, a_1, a_2, b_0, b_1, b_2 \in D$ telles que $a_0 \neq a_1$, $b_1 \neq b_2$, $g(a_0, a_2) \neq g(a_1, a_2)$ et $g(b_0, b_1) \neq g(b_0, b_2)$. Clairement, les vecteurs $j_1 = (a_0, a_1, b_0, b_0)$ et $j_2 = (a_2, a_2, b_1, b_2)$ appartiennent à $J$. Cependant, le vecteur

$$\big(g(a_0, a_2),\ g(a_1, a_2),\ g(b_0, b_1),\ g(b_0, b_2)\big)$$

est absent de $J$. Ainsi la fonction $g$ ne peut pas être un polymorphisme d'un ensemble de relations contenant $J$. À partir de toute autre fonction $f$ d'arité supérieure à 2 nous pouvons toujours produire une fonction binaire par identification de variables. □

Puisque le nombre d'endomorphismes sur un domaine fini est toujours fini, nous sommes assurés d'obtenir une caractérisation finie de la complexité de DCSP.

## 4 Complexité de DCSP$(S)$

Dans cette section, nous exhiberons une dichotomie de la complexité de $\mathrm{DCSP}(S)$ caractérisée par un critère simple. Cependant nous verrons qu'il est plus facile de prouver cette dichotomie pour le problème $\mathrm{DCSP}(f(S))$ où $f$ est une fonction unaire gardant invariant l'ensemble $S$. Nous avons besoin de la proposition suivante adaptée d'une proposition de Jeavons [6] démontrant que les problèmes $\mathrm{DCSP}(S)$ et $\mathrm{DCSP}(f(S))$ sont logspace équivalent. Ainsi, quand $\mathrm{DCSP}(f(S))$ est NP-complet, il en est de même pour $\mathrm{DCSP}(S)$.

**Proposition 4.1 ([6])** *Soit $S$ un ensemble de relations sur $D$ et $f$ une fonction unaire sur $D$. Soit $f(S) = \{f(R) \mid R \in S\}$. Si chaque relation $R \in S$ est close par $f$ alors* $\mathrm{DCSP}(f(S))$ *est logspace équivalent à* $\mathrm{DCSP}(S)$.

*Preuve :* Supposons que chaque relation $R \in S$ est close par $f$. Soit $\varphi(\vec{x})$ une instance de $\mathrm{DCSP}(f(S))$. Nous avons une formule $\varphi(\vec{x})$ où les littéraux $R(x_1, x_2, \ldots, x_k)$ sont construits à partir des relations $R \in f(S)$. D'après l'hypothèse, toutes les relations $R \in S$ sont closes par $f$, *i.e.* l'inclusion $f(S) \subseteq S$ est vérifiée. Nous en déduisons que $\varphi(\vec{x})$ est également une instance de $\mathrm{DCSP}(S)$.

Maintenant prenons une formule $\varphi(\vec{x})$ étant une instance de $\mathrm{DCSP}(S)$. Cette instance peut être transformée par une réduction logspace en une instance $\varphi'(\vec{x})$ de $\mathrm{DCSP}(f(S))$ en remplaçant chaque littéral construit à partir d'une relation $R$ par un littéral construit à partir de $f(R)$. De plus, grâce au fait que $R$ est close par $f$, nous avons $f(R) \subseteq R$ et nous en déduisons que toutes les solutions de $\varphi'(\vec{x})$ sont également des solutions de $\varphi(\vec{x})$. Réciproquement, si $h$ est une solution de $\varphi(\vec{x})$ alors $f(h)$ est une solution de $\varphi'(\vec{x})$. Ainsi nous avons une logspace équivalence entre $\mathrm{DCSP}(f(S))$ et $\mathrm{DCSP}(S)$. □

Avant d'exposer les propositions nécessaires à la preuve du théorème dichotomique, nous introduisons deux propositions aussi simples qu'essentielles.

**Proposition 4.2** *Soit $S$ un ensemble de relations. $S$ est d-valide si et seulement si $\langle S \rangle$ est d-valide.*

*Preuve :* Le sens $\langle S \rangle$ $d$-valide implique $S$ $d$-valide est trivial du fait que nous avons $S \subseteq \langle S \rangle$, pour tout ensemble $S$.

Pour prouver le sens $S$ $d$-valide implique $\langle S \rangle$ $d$-valide, il est suffisant de vérifier si les quatre opérations de conjonction, disjonction, identification de variables et la projection gardent la propriété de $d$-validité.

La conjonction est en fait un simple produit cartésien. Il est clair que le produit cartésien de deux relations $d$-valides est également $d$-valide.

La disjonction est l'union de deux produits cartésien particuliers, *i.e.* l'union d'un produit cartésien entre une $R_1$ et chaque $m$-tuple possible dans $D^m$ - en particulier le $m$-tuple $(d, d, \ldots, d)$ - et un produit cartésien entre chaque $n$-tuples possible dans $D^n$ - en particulier le $n$-tuple $(d, d, \ldots, d)$ - et une relation $R_2$, telles que l'arité de $R_1$ et $R_2$ sont respectivement $n$ et $m$. Ainsi, il est encore une fois clair que la disjonction est $d$-valide si $R_1$ et $R_2$ sont tous deux $d$-valide.

Enfin, il est également clair par définition que l'identification de variables et la projection sont $d$-valides si nous les appliquons sur des relations $d$-valides. □

La proposition suivante présente le lien entre un ensemble $d$-valide $S$ et l'existence d'une fonction constante dans $F$ l'ensemble des polymorphismes de $S$.

**Proposition 4.3** *Soit $S$ un ensemble de relations et $F$ l'ensemble des polymorphismes de $S$. $S$ est d-valide si et seulement s'il existe une fonction constante $f$ dans $F$ telle que pour tout $a \in D$, on ait $f(a) = d$.*

*Preuve :* Soit $S$ un ensemble de relations étant $d$-valide. Alors la fonction constante $f$ évaluant chaque élément $a \in D$ à $d$ (c'est-à-dire que pour tout $a \in D$, on a $f(a) = d$) est un polymorphisme de $S$, autrement dit $f$ appartient à $F$.

Soit maintenant $f$ une fonction constante dans $F$ évaluant chaque entrée à $d$. Puisque $c$ est un polymorphisme de $S$, nous devons avoir $f(R) \in R$, pour chaque relation $R$ dans $S$. Cependant, nous avons $f(R) = (d, d, \ldots, d)$ pour toute relation $R$. Alors, chaque relation dans $S$ contient un $d$-vecteur, signifiant que $S$ est $d$-valide. □

Nous étudierons les problèmes $\mathrm{DCSP}(S)$ à travers les ensembles de fonctions $F$ satisfaisant l'identité $\mathrm{Inv}\, F = S$. Les deux propositions suivantes démontrent que $\mathrm{DCSP}(S)$ est polynomial si $S$ est $d$-valide, et est NP-complet sinon.

**Proposition 4.4** *Soit $S$ un ensemble de relations. Si $S$ est d-valide alors* $\mathrm{DCSP}(S)$ *est décidable en un temps polynomial.*

*Preuve :* Si $S$ est $d$-valide, ceci signifie qu'il existe une valeur $d \in D$ telle que chaque relation $R \in S$ contient un $d$-vecteur, *i.e.* une application de chaque élément du domaine vers la valeur $d$. Ainsi chaque instance de $\mathrm{DCSP}(S)$ est safisfaisable par un $d$-vecteur. □

Avant d'étudier le cas où $S$ n'est pas $d$-valide, nous observons la proposition suivante.

**Lemme 4.5** *Soit $S$ un ensemble de relations tel que $F$ soit l'ensemble des polymorphismes de $S$ et le clone $[F]$ ne contient aucune fonction constante. Soit $f \in [F]$ une fonction avec la plus petite cardinalité de* $\mathrm{ran}\, f$. *Alors* $\mathrm{End}\, f(S)$ *contient seulement des permutations.*

*Preuve :* Supposons qu'il existe une fonction $g \in \mathrm{End}\, f(S)$ qui ne soit pas une permutation. Nécessairement, $g$ n'est pas injective, *i.e.* l'inclusion $\mathrm{ran}\, g \subsetneq \mathrm{ran}\, f$ est vérifiée. Ceci est une contradiction avec le fait que la cardinalité de $\mathrm{ran}\, f$ est la plus petite parmi toutes les fonctions dans $[F]$. □

Nous sommes maintenant en mesure de nous occuper du cas où $S$ n'est pas $d$-valide.

**Proposition 4.6** *Soit $S$ un ensemble de relations. Si $S$ n'est pas d-valide alors* DCSP($S$) *est* NP*-complet.*

*Preuve :* Ici nous adaptons la preuve de la proposition 5.6 de l'article de Jeavons [6]. Soit $S$ un ensemble de relations qui n'est pas $d$-valide, et $F$ l'ensemble des polymorphismes de $S$. Soit $f \in [F]$ une fonction dont la cardinalité de ran $f$ est la plus petite parmi les fonctions dans $F$. Par le lemme 4.5, nous savons que l'ensemble End $f(S)$ contient uniquement des permutations.

Puisque $S$ n'est pas $d$-valide, nous savons par la proposition 4.2 que $\langle S \rangle$ n'est pas $d$-valide. Ainsi il est clair que $[F]$ ne contient pas de fonction constante par la proposition 4.3. Nous savons également que la cardinalité de ran $f$ satisfait la condition $|\text{ran}\, f| \geq 2$. Nous traitons les deux cas suivants séparément :

Si $|\text{ran}\, f| = 2$, alors nous posons sans perte de généralité que $\text{ran}\, f = \{0, 1\}$. L'ensemble End $f(S)$ contient seulement des permutations sur $\{0, 1\}$. Soit $R_{NAE}$ la relation $\{0, 1\}^3 \setminus \{000, 111\}$. Il est clair que cette relation $R_{NAE}$ est close par n'importe quelle permutation sur $\{0, 1\}$. Ainsi nous avons End $f(S) \subseteq$ End $R_{NAE}$. La relation $R_{NAE}$ correspond au problème NOT-ALL-EQUAL-3SAT, connu pour être NP-complet. Nous en concluons que DCSP($f(S)$) est NP-complet.

Soit maintenant $|\text{ran}\, f| \geq 3$. L'ensemble de relations Inv End($f(S)$) est clos par les permutations sur ran $f$. En particulier, il contient l'ensemble des relations $Q \subseteq D^2$ où $Q = \{a_1, a_2, \ldots, a_k\}^2 \setminus \{(a_1, a_1), (a_2, a_2), \ldots, (a_k, a_k)\}$, tel que les éléments $a_1, \ldots, a_k$ présents dans la relation satisfassent $|\{a_1, a_2, \ldots, a_k\}| = |\text{ran}\, f|$. Les relations dans $Q$ sont les valuations valides pour toutes instances du problème de $|\text{ran}\, f|$-coloriage. Ce problème est connu pour être NP-complet puisque $|\text{ran}\, f| \geq 3$. Nous en concluons que DCSP($f(S)$) est également NP-complet dans ce cas présent.

Nous avons vu dans la proposition 4.1 que le problème DCSP($f(S)$) est logspace équivalent à DCSP($S$). Nous en déduisons que DCSP($S$) est NP-complet. □

De ces deux dernières propositions nous dérivons le principal théorème de cet article.

**Théorème 4.7** *Le problème* DCSP($S$) *est polynomial si $S$ est d-valide. Autrement, il est* NP*-complet.*

*Preuve :* Immédiat par les propositions 4.4 et 4.6. □

Nous avons présenté un critère très simple de dichotomie concernant la complexité du problème DCSP($S$) sur un domaine $D$ fixé de taille finie. Pour décider de la complexité de DCSP($S$), il suffit de vérifier si l'ensemble $S$ est $d$-valide ou non.

Nous pouvons également considérer les CSP disjonctifs en partant d'un ensemble de fonctions $F$. Étant donné un ensemble de fonctions unaires, nous considérons le problème DCSP(Inv $F$). Ainsi, par le proposition 4.3 et le précédent théorème 4.7, nous concluons que DCSP(Inv $F$) est dans P si $F$ contient une fonction constante, et est NP-complet sinon. Une question intéressante d'un point de vue complexité est de se demander si, étant donné un ensemble de fonctions unaires $F$, le clone $[F]$ contient une fonction constante. Ce méta-problème est traité dans la section suivante.

## 5 Complexité des clones

Pour déterminer si une composition de fonctions venant d'un ensemble $F$ produit une fonction constante, on doit calculer au moins une partie du clone $[F]$. Salomaa [10] appelle *class membership problem* le problème de décider, étant donné un ensemble de fonctions $F$ et une classe de fonction $C$, si le clone $[F]$ contient une fonction appartenant à $C$. Salomaa a prouvé ce problème NP-difficile. Il est même NP-complet si $C$ est la classe des fonctions constantes. Cependant, il est intéressant de noter que nous travaillons sur les CSP où la taille du domaine est fixée. Nous verrons que ceci nous permet de prouver que le *class membership problem* est fixed-parameter tractable (FPT).

Nous avons besoin tout d'abord d'un résultat de Salomaa [10] nous permettant de majorer la profondeur d'une combinaison de fonctions pour obtenir une fonction constante. Cette limite sera utile pour la preuve de l'appartenance à la classe NP par la suite.

**Proposition 5.1 ([10])** *Soit $F$ un ensemble de fonctions sans fonction constante. Soit $D$ un domaine de taille $n$. Chaque fonction constante $f_c$ sur $D$ vérifie la condition suivante*

$$\text{Depth}(f_c) \leq n^3/2 - 3n^2/2 + 2n$$

Suivant Salomaa [10], il n'est pas nécessaire d'aller au delà de cette borne polynomiale pour trouver une fonction constante dans $[F]$. La proposition suivante présente le résultat de Salomaa sur la NP-difficulté de *class membership problem.*

**Proposition 5.2 ([10])** *Class membership problem est* NP*-difficile.*

De la proposition 5.2 nous déduisons le résultat suivant.

**Proposition 5.3** *Class membership problem est* NP*-complet si $C$ est la classe des fonctions constantes.*

*Preuve :* Nous savons par la proposition 5.2 que ce problème est NP-difficile. Nous avons simplement besoin de démontrer qu'il appartient à NP. Par la proposition 5.1, nous savons que la profondeur de chaque fonction constante est limitée par $n^3/2 - 3n^2/2 + 2n$, avec $n$ la taille du domaine. Un certificat $f_c$ ne peut pas être plus long que cette limite. Nous avons à vérifier deux conditions. Premièrement, que chaque fonction composant la séquence $f_c$ est dans $F$, et deuxièmement, que $f_c$ est une fonction constante. La première étape est en $O(n^3.|F|)$ puisque la profondeur de $f_c$ est en $O(n^3)$, et la seconde demandant de calculer $f_c$ est en $O(n^4)$ puisque nous avons à calculer $n$ valeurs $n^3$ fois. Notez que $|F|$ est le nombre $k$ de fonctions dans $F$ fois la taille d'une fonction, qui est $n$. Ainsi, le certificat $f_c$ a une taille dépendant de $n$, et il peut être décidé en $O(k.n^4)$ si $f_c$ est une fonction constante. Alors Class membership problem est dans NP si $C$ est la classe des fonctions constantes, et ainsi ce problème est NP-complet. □

Nous nous concentrons maintenant sur la complexité du problème de décider si une fonction constante peut être obtenue à partir de fonctions dans $F$, *i.e.* le *class membership problem* où $C$ est la classe des fonctions constantes. Si nous posons $D$ comme ayant une taille fixée, nous pouvons considérer la version suivante du *parametric class membership problem* :

**Problème :** PARAMETRIC CLASS MEMBERSHIP PROBLEM
*Entrée :* Un ensemble $F$ de fonctions.
*Paramètre :* La taille $n$ du domaine.
*Question :* Le clone $[F]$ contient-il une fonction constante ?

Nous montrerons que la complexité de ce problème est *fixed-parameter tractable*. Nous commençons par introduire cette classe de complexité.

**Définition 5.4 ([4])** *Un problème paramétrique $P$ est* fixed-parameter tractable*, ou* FPT*, s'il existe un algorithme prenant en entrée $(I, k)$, où $I$ est la principale partie de l'entrée et $k$ le paramètre, et décidant l'appartenance $(I, k) \in P$ en temps $f(k).|I|^c$, avec $f$ une fonction arbitraire et $c$ une constante.*

**Théorème 5.5** *Le parametric class membership problem pour un ensemble de fonctions unaires $F$ est fixed-parameter tractable et peut être décidé en temps $O(n^n.|F|)$, où $n$ est la taille du domaine ainsi que le paramètre.*

*Preuve :* Il suffit de montrer un algorithme décidant de ce problème, prenant comme paramètre la taille $n$ du domaine et terminant en $O(f(n).|F|^c)$ pour une certaine constante $c$. Ainsi la preuve est relative à la complexité de l'algorithme 1.

**Algorithm 1** Parametric Class Membership Problem

```
 1: Q ← {f_i | f_i ∈ F}
 2: S ← ∅
 3: while Q ≠ ∅ do
 4:     f ← dequeue(Q)
 5:     S ← S ∪ {f}
 6:     for all f_i ∈ F do
 7:         g = f_i ∘ f
 8:         if g is a constant function then
 9:             return "YES"
10:         end if
11:         if g ∉ S then
12:             Q ← Q @ {g}
13:         end if
14:     end for
15: end while
16: return "NO"
```

Tout d'abord, montrons que l'algorithme 1 est correct et termine. On remarque que $Q$ est l'ensemble des fonctions que nous devons traiter (représenté par une file), et $S$ l'ensemble des fonctions déjà produites. Au début, $Q$ est initialisé à $F$. Pour chaque élément de $Q$, l'algorithme les compose avec chaque fonction de $F$, et empile ces combinaisons dans $Q$ si elles sont nouvelles (c'est-à-dire absente de $S$). La propriété suivante est un invariant de boucle : $Q$ ne contient jamais deux fois la même fonction. Chaque fonction possible générée par un combinaison sera explorée par l'algorithme, et testée pour savoir s'il s'agit d'une fonction constante, démontrant la correction de l'algorithme.

Puisque le nombre de fonctions unaires sur un domaine fini de taille $n$ ne peut pas être plus grand que $n^n$, la taille de $Q$ ne dépasse pas cette limite. Par la ligne 11 nous savons que la file $Q$ ne peut pas contenir deux fois la même fonction et la ligne 4 nous montre qu'un élément est retiré de $Q$ à chaque tour de la boucle while. Ainsi l'algorithme 1 termine.

Analysons la complexité de l'algorithme 1. Nous savons que $|Q| \leq n^n$, ainsi nous avons la limite du nombre de tours de la boucle while. La ligne 6 ne dépend que de la taille de $F$ et la complexité des lignes 7 à 13 ne dépendent que du choix des structures de données. Si nous choisissons d'utiliser une table de hachage pour représenter $S$, et que la longueur des listes de collisions est proportionnelle à $n$, alors ces lignes sont exécutées en $O(n)$. Il est clair que l'algorithme 1 tourne en $O(n^n.|F|)$ et nous permet de conclure que le *parametric class membership problem* est fixed-parameter tractable, où $n$ est le paramètre. □

## 6 Complexité des clones sur les domaines ternaires

Dnas la section précédente, nous avons montrer que le méta-problème est FPT, étant donné la taille du domaine comme paramètre. Nous nous interessons maintenant à la complexité du méta-problème où la taille du domaine n'est plus un paramètre, mais est connue et fixée. Nous dégageons ainsi une nouvelle technique prometteuse pour déterminer la complexité du méta-problème où la taille du domaine est fixée.

Ainsi, le méta-problème pour DCSP sur les domaines ternaires peut être traité autrement que le méta-problème général. En fait, nous n'avons pas à calculer $[F]$, même en partie, pour savoir s'il existe une combinaison de fonctions dans $F$ formant une fonction constante. Pour savoir si c'est possible de produire une telle fonction, il suffit de vérifier si les fonctions de $F$ remplissent certaines conditions. Notons tout d'abord le fait suivant.

**Remarque 6.1** *Les kernels des fonctions sur un domaine ternaire sont limités à une seule classe d'équivalence. Ceci peut être facilement vérifié par le principe de Dirichlet. De plus, la taille de ces kernels ne peut qu'être égal à 0, 2 ou 3.*

Avant tout, nous avons besoin de ce lemme fondamental, vrai pour tous domaines finis.

**Lemme 6.2** *Soit $F$ un ensemble de fonctions sans fonction constante. Le clone $[F]$ contient une fonction constante $f_c$ si et seulement s'il existe deux fonctions $f_a, f_b \in [F]$ telles que $f_c = f_a \circ f_b$ et $\operatorname{ran} f_b \subseteq \ker f_a$.*

*Preuve :*
$\boxed{\Leftarrow}$ Trivial.

$\boxed{\Rightarrow}$ Soit $f_c \in [F]$ une fonction constante. Alors $f_c$ doit être une composition de deux fonctions $f_a$ et $f_b$ puisque $F$ ne contient pas de fonction constante. Sans perte de généralité, posons $f_a$ et $f_b$ comme étant deux fonctions non constantes telles que $f_c = f_a \circ f_b$.

Supposons que pour chaque classe d'équivalence $[d]_{f_a} \in \ker f_a$, nous avons $\operatorname{ran} f_b \nsubseteq [d]_{f_a}$. Soit $x, y \in \operatorname{ran} f_b$ tel que, pour chaque $[d]_{f_a}$ on a $\{x, y\} \nsubseteq [d]_{f_a}$. Alors $f_a(x) \neq f_a(y)$, mais puisque $x, y \in \operatorname{ran} f_b$, il doit exister $x', y' \in D$ tels que $f_b(x') = x$ et $f_b(y') = y$. Ainsi, $(f_a \circ f_b)(x') \neq (f_a \circ f_b)(y')$, *i.e.* $f_c(x') \neq f_c(y')$. Ceci est une contradiction avec le fait que $f_c$ soit une fonction constante. □

De ce lemme nous déduisons le corollaire suivant.

**Corollaire 6.3** *S'il existe deux fonctions $f_a$ et $f_b$ telles que $\operatorname{ran} f_b = \ker f_a$, alors la composition $f_a \circ f_b$ est une fonction constante.*

En plus du lemme 6.2, nous montrons quelques résultats utiles sur les images et les kernels des fonctions.

**Lemme 6.4** *Soient $f, g \in F$.*

*(i)* $\operatorname{ran}(f \circ g) \subseteq \operatorname{ran} f$ *et* $\ker g \subseteq \ker(f \circ g)$ *;*

*(ii) si* $\operatorname{ran} g \nsubseteq \ker f$ *et* $|\ker f| = 2$ *alors* $\operatorname{ran}(f \circ g) = \operatorname{ran} f$ *;*

*(iii) si* $\operatorname{ran} g \nsubseteq \ker f$ *et* $|\ker g| = 2$ *alors* $\ker g = \ker(f \circ g)$.

*Preuve :* Chaque fonction unaire $f$ est monotone, *i.e.*, si $A \subseteq B$ alors on a $f(A) \subseteq f(B)$ pour tous sous-ensembles $A, B$ de $D$. Puisque $\operatorname{ran} g \subseteq D$ et $f$ est monotone, nous avons $f(\operatorname{ran} g) \subseteq f(D)$. De plus, $f(\operatorname{ran} g)$ est $\operatorname{ran}(f \circ g)$ et $f(D)$ est $\operatorname{ran} f$. Ainsi on a $\operatorname{ran}(f \circ g) \subseteq \operatorname{ran} f$.

Soit $x, y \in \ker g$. On a $g(x) = g(y)$, donc $(f \circ g)(x) = (f \circ g)(y)$, *i.e.* $x, y \in \ker(f \circ g)$.

Soit $\operatorname{ran} g \nsubseteq \ker f$ et $|\ker f| = 2$. Nous devons montrer que $\operatorname{ran} f \subseteq \operatorname{ran}(f \circ g)$. Soit $y \in \operatorname{ran} f$. Supposons que pour tout $x \in \operatorname{ran} g$, on ait $f(x) \neq y$. Soit $z \in \operatorname{ran} f$, avec $z \neq y$. Donc pour tout $x \in \operatorname{ran} g$, on a $f(x) = z$ puisque $|\ker f| = 2$, ce qui implique $\operatorname{ran} g \subseteq \ker f$ : contradiction avec notre supposition.

Soit $\operatorname{ran} g \nsubseteq \ker f$. Nous devons montrer que $\ker(f \circ g) \subseteq \ker g$. Supposons qu'il existe $x, y \in \ker(f \circ g)$ tels que $g(x) \neq g(y)$. Puisque $|\ker g| = 2$, on a $\{g(x), g(y)\} = \operatorname{ran} g$. Cependant on a $(f \circ g)(x) = (f \circ g)(y)$, donc $\operatorname{ran} g \subseteq \ker f$, ce qui est une contradiction. Ainsi, pour tout $x, y \in \ker(f \circ g)$, on a $x, y \in \ker g$. □

Une fois encore, nous déduisons le résultat suivant du lemme 6.4.

**Corollaire 6.5** *Soit $\{f, g\} = F$ tel que $\operatorname{ran} g \nsubseteq \ker f$, $\operatorname{ran} f \nsubseteq \ker g$ et $|\ker f| = |\ker g| = 2$. Si $\operatorname{ran} f = \operatorname{ran} g$ alors pour chaque fonction $h \in [F]$ nous avons $\operatorname{ran} h = \operatorname{ran} f$. Si $\ker f = \ker g$ alors pour chaque fonction $h \in [F]$, on a $\ker h = \ker f$.*

*Preuve :* Immédiat par le lemme 6.4. □

Enfin, nous avons besoin des notions de *permutation circulaire* et *swap* sur un domaine ternaire pour présenter le principal résultat de cette section.

**Définition 6.6** *Une **permutation circulaire** $c$ sur $D = \{0, 1, 2\}$ est une permutation satisfaisant la condition $c(x) = (x + k) \mod |D|$ avec $k \in D$, pour tout $x \in D$.*

*Une **permutation swap** $s$ sur $D = \{0, 1, 2\}$ est une permutation satisfaisant les conditions $s(x) = y$, $s(y) = x$ et $s(z) = z$, pour $x, y, z \in D$ distincts. L'ensemble $\{x, y\}$ est appelé* $\operatorname{swap} s$.

Nous pouvons maintenant introduire notre principal résultat, divisé en deux parties.

**Proposition 6.7** *Soit $F$ un ensemble de fonctions. Si $F$ satisfait l'une des conditions suivantes alors il existe une fonction constante dans $[F]$.*

*(i) il existe une fonction constante $f \in F$.*

*(ii) il existe $f, g \in F$ (pas nécessairement distinctes) telles que* $\operatorname{ran} f = \ker g$.

*(iii) il existe $f, c \in F$ telles que* $|\ker f| = 2$ *et $c$ soit une permutation circulaire.*

*(iv) il existe $f, s \in F$ telles que* $|\ker f| = 2$ *et $s$ soit une permutation swap où* $\operatorname{swap} s \neq \operatorname{ran} f$ *et* $\operatorname{swap} s \neq \ker f$.

*(v) il existe $f, s_1, s_2 \in F$ telles que* $|\ker f| = 2$ *et $s_1, s_2$ sont des permutations swap satisfaisant* $\operatorname{swap} s_1 \neq \operatorname{swap} s_2$.

*Preuve :* Le cas (i) est trivial. Le cas (ii) est direct par la corollaire 6.3. Notons ici que l'existence de $f$ et $g$ satisfaisant $\operatorname{ran} f \subseteq \ker g$ implique soit $\operatorname{ran} f = \ker g$ soit $|\ker g| = 3$, *i.e* $g$ est une constante. Le cas (iii) est immédiat par (ii) si $\operatorname{ran} f = \ker f$. Sinon, soit $\operatorname{ran} f = \{x, y\}$ et $\ker f = \{y, z\}$ avec $x, y, z \in D$ tous différents. Notons que $\ker(c^2 \circ f) = \ker(c \circ f) = \ker f$. Il y a deux cas possibles :

- $\operatorname{ran}(c \circ f) = \{y, z\}$, donc par (ii) $c \circ f$ est une fonction constante.
- $\operatorname{ran}(c \circ f) = \{x, z\}$. Il est facile de voir que $\operatorname{ran}(c^2 \circ f) = \{y, z\}$. Par (ii), $c^2 \circ f$ est une fonction constante.

Le cas (iv) est lui aussi immédiat par (ii) si $\operatorname{ran} f = \ker f$. Sinon, on a $s(\operatorname{ran} f) = \ker f$. Ainsi pour tout $x \in D$ on a $(s \circ f)(x) = \ker f$, *i.e.* $\operatorname{ran}(s \circ f) = \ker f$, et nous concluons par (ii) que t $(f \circ s \circ f)$ est une fonction constante. Le cas (v) est immédiat par (ii) si $\operatorname{ran} f = \ker f$. Sinon, puisque l'on a $\operatorname{swap} s_1 \neq \operatorname{swap} s_2$, la composition $s_1 \circ s_2$ produit nécessairement une permutation circulaire. Nous pouvons alors conclure par (iii). □

Nous allons voir maintenant que les conditions de la proposition 6.7 sont nécessaires pour obtenir une fonction constante dans $[F]$.

**Proposition 6.8** *Soit $F$ un ensemble de fonctions satisfaisant aucune des conditions de la proposition 6.7. Alors il n'existe pas de fonction constante dans $[F]$.*

*Preuve :* Si $F$ ne vérifie aucune condition de la proposition 6.7, alors nous sommes dans l'un des cas suivants :

1. $F$ ne contient que des permutations,
2. pour chaque $f \in F$ nous avons $|\ker f_i| = 2$ et pour tout $f_i, f_j \in F$ (éventuellement $f_i = f_j$), on a $\operatorname{ran} f_i \neq \ker f_j$ et $\operatorname{ran} f_j \neq \ker f_i$.
3. $F$ satisfait la condition 2 et contient également des permutations swap avec le même ensemble swap, tel que pour toute $f_i \in F$ avec $|\ker f_i| = 2$, pour toute permutation swap $s_k \in F$, on a $\operatorname{swap} s_k = \ker f_i$ ou $\operatorname{swap} s_k = \operatorname{ran} f_i$. Notons que les deux en même temps est impossible puisque nous avons $\ker f_i \neq \operatorname{ran} f_i$.

Le cas (1) est trivial. Si $F$ ne contient que des permutations, alors $[F]$ ne contient seulement que des permutations également. Par le principe de Dirichlet, le cas (2) implique $\ker f_i = \ker f_j$, ou $\operatorname{ran} f_i = \operatorname{ran} f_j$, ou les deux, pour tout $f_i, f_j \in F$. Par le corollaire 6.5, nous savons que $\operatorname{ran} f_i = \operatorname{ran} f_j$ pour tout $f_i, f_j \in F$ implique que chaque $f \in [F]$ vérifie $\operatorname{ran} f = \operatorname{ran} f_i$. Puisque $|\ker f_i| = 2$ implique $|\ker f| = 2$, alors $f$ ne peut pas être une fonction constante. Nous pouvons appliquer le même argument pour le cas où $\ker f_i = \ker f_j$ pour tout $f_i, f_j \in F$.

Comme dans le cas (2), pour tout $f_i, f_j \in F$ nous avons $\ker f_i = \ker f_j$, ou $\operatorname{ran} f_i = \operatorname{ran} f_j$, ou les deux. Nous distinguons quatre cas.

Soit $\operatorname{ran} f_i = \operatorname{ran} f_j$, pour tout $f_i, f_j \in F$, et $\operatorname{swap} s_k = \operatorname{ran} f_i$. Il est clair que $\operatorname{ran}(s_k \circ f_i) = \operatorname{ran}(f_i \circ s_k) = \operatorname{ran} f_i$. Puisque $|\ker f_i| = 2$, on a $|\ker(s_k \circ f_i)| = |\ker(f_i \circ s_k)| = 2$, donc $s_k \circ f_i$ et $f_i \circ s_k$ ne peuvent pas être des fonctions constantes.

Maintenant soit $\operatorname{swap} s_k = \ker f_i$, pour un $f_i \in F$. Nous devons avoir $\ker f_i = \ker f_j$ pour tout $f_i, f_j \in F$ parce qu'autrement il existe $f_k \in F$ tel que $\operatorname{swap} s_k = \operatorname{ran} f_k$, et ainsi $\ker f_k = \operatorname{ran} f_i$ ce qui constitue une contradiction. Alors on a $\ker(s_k \circ f_i) = \ker(f_i \circ s_k) = \ker f_i$. Puisque $|\ker f_i| = 2$, nous concluons qu'il n'existe aucune composition produisant une fonction constante.

Par les même arguments, nous voyons que nous ne pouvons avoir de fonction constante si $\ker f_i = \ker f_j$ pour tout $f_i, f_j \in F$, que l'on ait $\operatorname{swap} s_k = \operatorname{ran} f_i$ ou $\operatorname{swap} s_k = \ker f_i$. □

Par les précédentes propositions, nous en déduisons le théorème suivant.

**Théorème 6.9** *Étant donné un ensemble de fonctions $F$ sur un domaine ternaire, le problème de savoir si le clone $[F]$ contient une fonction constante peut être décidé en temps polynomial.*

*Preuve :* Par les propositions 6.7 et 6.8, nous savons que $[F]$ contient un fonction constante si et seulement si $F$ satisfait au moins l'une des conditions de la proposition 6.7. Le test de satisfaisabilité de chaque condition peut se faire en temps polynomial.

Pour le cas (i), nous avons à tester pour tout $f \in F$ si $f$ est une fonction constante. Cette condition est

vérifiée en temps $O(|F|)$. Pour le cas (ii), nous cherchons $f, g \in F$ dont nous calculons $\ker f$ et $\operatorname{ran} g$, telles que $\operatorname{ran} g \subseteq \ker f$. Cette vérification se fait en $O(|F|^2)$. Concernant le cas (iii), nous cherchons $f \in F$ tel que $|\ker f| = 2$ et une permutation circulaire $c \in F$. Cela a fait en $O(|F|)$. Pour le cas, (iv), on doit vérifier pour tout $f, s \in F$ si $|\ker f| = 2$, si $s$ est une permutation swap, et si $\operatorname{swap} s \neq \ker f$ et $\operatorname{swap} s \neq \operatorname{ran} f$. Ceci peut être fait en $O(|F|^2)$. Enfin, concernant le cas (v), nous cherchons $f \in F$ telle que $|\ker f| = 2$ et $s_1, s_2 \in F$ telles que $s_1$ et $s_2$ sont des permutations swap, telles que $\operatorname{swap} s_1 \neq \operatorname{swap} s_2$. Ceci est vérifié en $O(|F|^2)$. □

Nous avons dégagé un algorithme en temps polynomial concernant le méta-problème sur un domaine ternaire. Bien que sa complexité en $O(|F|^2)$ soit en pratique moins efficace que l'algorithme du méta-problème général en $o(n^n.|F|)$, avec comme constante $n = 3$, cet algorithme utilise une méthode permettant d'éviter de calculer, même partiellement, le clone $[F]$. Ainsi, cette méthode contitue une approche sérieuse pour obtenir des algorithmes en temps polynomial plus efficace que l'algorithme du méta-problème général.

## 7 Conclusion

Nous avons analysé la complexité algorithmique des problèmes de satisfaction de contraintes disjonctifs. Nous avons obtenu un caractérisation complète à travers un théorème dichotomique, distinguant les instances dites faciles des difficiles. Puisque l'ensemble des endomorphismes $\operatorname{End} S$ est égal au clone $[F]$ généré par un ensemble $F$ de fonctions unaires, il était également intéressant d'étudier la complexité du méta-problème, à savoir étant donné un ensemble de fonctions unaires $F$, si le clone $[F]$ contient une fonction constante. Nous avons montré que le méta-problème est NP-complet si le domaine fait parti de l'entrée, mais qu'il est fixed-parameter tractable, avec un algorithme s'exécutant en temps $O(n^n\,|F|)$, si l'on considère le domaine comme étant un paramètre. En particulier, nous avons une analyse plus fine de la complexité du méta-problème dans les domaines ternaires, nous permettant de répondre à la question de décider si un clone $[F]$ contient une fonction constante en temps polynomial, et ce sans avoir besoin de calculer, même partiellement, le clone $[F]$, alors que le méta-problème général est fixed-parameter tractable.

Nous pouvons étendre ces travaux vers plusieurs questions ouvertes. Il serait effectivement intéressant de savoir si le théorème dichotomique reste vrai si un ensemble de relations ne contient pas la relation d'égalité, et s'il reste vrai sur les domaines $\omega$ catégorique (grossièrement, sur les domaines infinis dénombrables), ou sur tous domaines infinis. De plus, trouver un algorithme polynomial pour répondre au méta-problème concernant des domaines de cardinalités supérieures est également un problème ouvert important.

## Références


[1] A. A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the Association for Computing Machinery*, 53(1) :66–120, 2006.

[2] D. Cohen, P. Jeavons, P. Jonsson, and M. Koubarakis. Building tractable disjunctive constraints. *Journal of the Association for Computing Machinery*, 47(5) :826–853, 2000.

[3] R. Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[4] R. G. Downey and M. R. Fellows. *Parametrized Complexity*. Springer-Verlag, 1999.

[5] T. Feder and M. Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction : a study through Datalog and group theory. *SIAM Journal on Computing*, 28(1) :57–104, 1998.

[6] P. Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200(1-2) :185–204, 1998.

[7] M. Krasner. Une généralisation de la notion de corps. *Journal de Mathématiques pures et appliquées*, 17 :367–385, 1938.

[8] R. Pöschel. Galois connection for operations and relations. Technical Report MATH-AL-8-2001, Technische Universität Dresden, 2001.

[9] E. L. Post. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies*, 5 :1–122, 1941.

[10] A. Salomaa. Composition sequences for functions over a finite domain. *Theoretical Computer Science*, 292(1) :263–281, 2003.

[11] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings 10th Symposium on Theory of Computing (STOC'78), San Diego (California, USA)*, pages 216–226, 1978.

[12] Yu. I. Yanov and A. A. Muchnik. On the existence of $k$-valued closed classes that have no bases. *Doklady Akademii Nauk SSSR*, 127 :44–46, 1959. In Russian.