Répertoire de modules logiques

pour la

Programmation Linéaire en Nombres Entiers

Christian Rivière

Programmation linéaire en nombres entiers

Utiliser un solveur et des techniques de programmation linéaire n'est pas toujours la méthode la plus appropriée pour résoudre un problème linéaire entièrement en variables entières, encore moins quand ces variables sont booléennes. Toutefois, de nombreux problèmes de programmation linéaire nécessitent localement l'utilisation de variables discrètes, entières ou même booléennes (condition, ensemble, ordre, liste, arbre, …). Bien que d'expression mathématique simple, l'insertion de ces variables booléennes dans un jeu de contraintes linéaires est parfois délicate. De plus, la mise au point du problème général devient tributaire d'un problème logique supplémentaire.

Le catalogue de formulations proposé ici, permet de s'affranchir de la complexité inhérente à l'expression des ces briques élémentaires.

Lineara programado per entjeroj

Uzi solvilon kaj teknikon de lineara programado ne ĉiam estas la plej adekvata metodo por solvi problemon kiu koncernas nur entjerajn variablojn, des malpli kiam tiuj variabloj estas binaraj. Tamen, multaj problemoj de lineara programado necesigas lokan uzon de diskretaj, entjeraj aŭ eĉ binaraj variabloj (kondiĉo, aro, ordo, listo, arbo, ...). Kvankam facile esprimebla matematike, enkonduko de binaraj variabloj en sistemon de linearaj trud-asertoj foje riveliĝas delikata. Krome, la funkciigo de la ĝenerala problemo estas tiuokaze pli komplika ĉar ĝi dependas de plia logika problemo aldone al la strikte reel-nombra problemo.

La formulado-katalogo ĉi-tie proponata, ebligas almenaŭ seniĝi je la komplekseco devena de tiuj logikaj problemoj.

Linear integer programming

Making use of automatic solvers and linear programming techniques is not always the most appropriate method for solving a linear problem solely involving integer variables, and even less so for logical variables. However, many linear programming problems require the use of local discrete variables, integer or even logical (condition, ensemble, order, list, tree...) Despite being easily expressible mathematically, the introduction of logical variables in a system of linear constraints can sometimes be a delicate procedure. Furthermore, the implementation of the general problem is complicated in such cases, as it depends on an additional logic problem on top of the strictly numerical one.

The online catalogue of formulations proposed here eliminates the complications due to such logic problems.

Programmazione lineare con numeri interi

Usare risolutori automatici e tecniche di programmazione lineare non è sempre il metodo più adeguato per risolvere un problema che contiene solamente variabili intere, tanto più se queste variabili sono binarie. Tuttavia, molti problemi di programmazione lineare rendono necessario l'uso locale di variabili discrete, intere o perfino binarie (condizione, insieme, ordine, lista, albero, ecc..). Sebbene facilmente esprimibile matematicamente, l'introduzione di variabili binarie in un sistema di vincoli lineari è spesso delicata. A parte questo, la messa a punto del problema generale è in tal caso più complessa poiché dipende da un ulteriore problema di logica oltre che dallo stretto problema numerico.

Il catalogo di formulazioni proposto, rende possibile almeno il superamento della complessità dovuta a tali problemi logici.

Table des matières

Pages

Avant-propos

En programmation linéaire, les contraintes s'expriment par des expressions du type : $A \geq B$ (i.e. A=B)

Le solveur utilise des moyens informatiques => le type utilisé est le flottant. Les variables ne sont donc pas à strictement parler réelles (espace infini, continu). En corollaire, les intervalles ouverts n'ont pas de sens, pas plus que A>B (i.e. $A \neq B$)

Dans la pratique, il est suffisant pour ce faire, d'utiliser des constantes du type :

ε (valeur arbitrairement petite)

∞ (valeur arbitrairement grande)

En programmation linéaire, les variables sont en principe positives ou nulles. Le traitement de variables négatives est possible après déclaration explicite. En cas d'impossibilité, on peut utiliser après translation :

ω (valeur moyennement grande)

Même si le type entier est proposé, les calculs sont effectués en flottant. Le type entier ne fait que réduire l'espace des solutions a posteriori. Il est donc illusoire de penser que la moyenne de deux entiers (valant resp. 1 et 2) soit entier sous le prétexte que le résultat est mis dans un entier $N = 0.5N_1 + 0.5N_2$ Le résultat fait 1.5, et donc le problème sera sans solution (car il n'existe pas d'entier valant 1.5)

Même si le problème est décrit au moyen d'un langage procédural (séquentiel), la programmation linéaire est déclarative (factuelle). Des instructions de type "*while* condition *do* …", "*for* (; ;) … " ou "*if* condition *then* … *else* …" n'ont aucun sens séquentiel, mais uniquement factuel. Il s'ensuit :

. "*while*" ou "*for*" signifient plus ou moins : "pour tout" ($\forall$) ;

. "*if* condition *then* … *else* …" signifie que, selon la condition, on impose une contrainte ou une autre ;

. les formes (cond_1 *et/ou* cond_2) sont commutatives et expriment une table logique, et non pas une succession de tests.

A ce titre, exprimer : *si* cond *alors* contrainte_1 *sinon* contrainte_2 est équivalent au système :

si contrainte_1 n'est pas satisfaite *alors* cond ne l'est pas

si contrainte_2 n'est pas satisfaite *alors* cond l'est

En fait, ce n'est pas à proprement parler "cond" qui, selon sa valeur, impose une contrainte ou l'autre, ni a contrario l'état satisfait d'une des deux contraintes qui imposerait la valeur d'un booléen, mais un système de **trois contraintes** satisfaites ou non selon une table logique. En final, c'est ce système qui doit être satisfait dans sa totalité.

A noter dans l'exemple ci-dessus que, si une contrainte n'est pas satisfaite l'autre doit l'être, ou alors c'est que le système est sans solution.

Concernant la fonction de coût, généralement les solveurs n'admettent pas de signe négatif, du type $\min(\alpha A - \beta B)$, même si l'expression est à coup sûr positive.

Dans ce qui suit :

Les majuscules sont les variables du système de contraintes
- *X ou Y* représentent des variables binaires
- *N* représente des variables entières
- *K* représente des variables "réelles"
- *A, B, C* et *D* représentent des membres quelconques de contraintes
- U_j, V_j représentent des éléments de séries de variables (n-uplets)

Les minuscules (*b, q, k, t,* …) sont des constantes ou des variables hors-problème (non déterminées par le système de contraintes)
- *b* comme "base de numération"
- *q* comme "quantité"
- *k, f(k)* représentent des "variables de fonction"
- *t* comme "tolérance"

Notations :

^	ou exclusif
$\overline{X}$	booléen complément : $\overline{X} = (1 - X)$
X = (condition)	si la condition est vérifiée $X = 1$ sinon $X = 0$
K = (condition) ? K_1 : K_2	si la condition est vérifiée $K = K_1$ sinon $K = K_2$

Contraindre que $K \in \cup\{k_j\}$

Variables supplémentaires

n variables binaires X_j

Formulation

$$\sum_j X_j = 1$$

$$K = \sum_j k_j X_j$$

Contraindre que $K \in \cup\lfloor k_j, k_{j'} \rfloor \cup \{0\}$

Variables supplémentaires

n variables binaires X_j

n variables réelles K_j

Formulation

$$K = \sum_j K_j$$

$$\sum_j X_j \leq 1$$

$$\forall j \;\; K_j \geq k_j X_j$$

$$\forall j \;\; K_j \leq k'_j X_j$$

Nota

cette méthode ne peut être utilisée pour $K \in \cup\lfloor K_j, K'_j \rfloor \cup \{0\}$ (variables)

Contraindre que $K \in \cup\lfloor k_j, k_{j'} \rfloor$

Variables supplémentaires

n variables binaires X_j

n variables réelles K_j

Formulation

$$K = \sum_j K_j$$

$$\sum_j X_j = 1$$

$$\forall j \;\; K_j \geq k_j X_j$$

$$\forall j \;\; K_j \leq k'_j X_j$$

Nota

cette méthode ne peut être utilisée pour $K \in \cup\lfloor K_j, K'_j \rfloor$ (variables)

tant qu'il n'y a pas plus de deux intervalles, on peut faire $X_1 = 1 - X_2$

Contraindre que $K \neq k$

Méthode

selon la formulation précédente, K_1 entre 0 et k-ε , K_2 entre k+ε et ∞

Variables supplémentaires

variable binaire X variables réelles K_1 K_2

Formulation

$K = K_1 + K_2$

$K_1 \leq (k - \varepsilon)X$

$K_2 \geq (k + \varepsilon)(1 - X)$

$K_2 \leq (1 - X).(\infty)$

Nota

pour des nombres entiers $N \neq n$, ε peut valoir 1

cette méthode ne peut être utilisée pour $K_1 \neq K_2$ (deux variables)

Contraindre que $K_1 > K_2$

Méthode

$K_1 \geq K_2$ et $K_1 \neq K_2$

Formulation

$K_1 \geq K_2 + \varepsilon$

Nota

pour des nombres entiers $N_1 > N_2$, ε peut valoir 1

Contraindre que $K_1 \neq K_2$

Méthode

$K_1 - K_2 \geq \varepsilon$ ou $K_2 - K_1 \geq \varepsilon$

Variables supplémentaires

variable binaire X

Formulation

$K_1 - K_2 \geq X.(-\infty) + \varepsilon$

$K_2 - K_1 \geq (1 - X).(-\infty) + \varepsilon$

Nota

pour des nombres entiers $N_1 \neq N_2$, ε peut valoir 1

Contraindre que $K_1 \approx K_2$ (tolérance t)

Formulation

$K_1 - K_2 \leq t$

$K_2 - K_1 \leq t$

Contraindre que $N = E(K)$, soit la partie entière (arrondi au nombre inférieur ; fonction *floor*)

Méthode

$N \leq K$, $N+1 \geq K$ et $N+1 \neq K$

Variables supplémentaires

variable binaire X

Formulation

$N \leq K$

$N+1 \geq K+\varepsilon$

Contraindre que $N = ceil(K)$, soit l'arrondi au nombre supérieur

Méthode

$N \geq K$, $N-1 \leq K$ et $N-1 \neq K$

Variables supplémentaires

variable binaire X

Formulation

$N \geq K$

$N-1 \leq K-\varepsilon$

Contraindre que $K_1 = |K_2|$

Méthode

$K_3 = K_2 + \omega$ (éventuellement hors problème soumis au solveur)

si $K_3 \geq \omega$ alors $K_1 = K_3 - \omega$ sinon $K_1 = \omega - K_3$

la formulation d'une condition est vue plus loin

Variable supplémentaire

variable binaire X

Formulation

$K_3 - \omega \geq (1-X).(-\infty)$

$\omega - K_3 \geq X.(-\infty)$

$K_1 - K_3 + \omega \geq (1-X).(-\infty)$

$K_3 - \omega - K_1 \geq (1-X).(-\infty)$

$K_1 - \omega + K_3 \geq X.(-\infty)$

$\omega - K_3 - K_1 \geq X.(-\infty)$

Nota

si les variables peuvent être négatives, prendre directement $\omega = 0$, donc $K_3 = K_2$

Contraindre que $K = \sup(K_1, K_2)$

Méthode

si $K_1 \geq K_2$ alors $K = K_1$ sinon $K = K_2$

la formulation d'une condition est vue plus loin

Variable supplémentaire

variable binaire X

Formulation

$K_1 - K_2 \geq (1 - X).(-\infty)$

$K_2 - K_1 - \varepsilon \geq X.(-\infty)$

$K_1 - K \geq (1 - X).(-\infty)$

$K - K_1 \geq (1 - X).(-\infty)$

$K_2 - K \geq X.(-\infty)$

$K - K_2 \geq X.(-\infty)$

Nota

pour des nombres entiers $N = \sup(N_1, N_2)$, ε peut valoir 1

Contraindre que $K = \sup(K_1, K_2)$ (variante simplifiée)

Méthode

K est plus grand que K_1 et que K_2, et plus petit qu'au moins un des deux

la formulation de *"qu'au moins un des deux"* est vue plus loin

Variable supplémentaire

variables binaires X_1 X_2

Formulation

$K \geq K_1$

$K \geq K_2$

$K_1 \geq K + X_1.(-\infty)$

$K_2 \geq K + X_2.(-\infty)$

$X_1 + X_2 \leq 1$

Contraindre que $K = \sup(K_j)$

Méthode

K est plus grand que tous les K_j et plus petit qu'au moins un d'entre eux

la formulation de *" qu'au moins un d'entre eux "* est vue plus loin

Variable supplémentaire

n variables binaires X_j

Formulation

$\forall j \in [1, n]\ K \geq K_j$

$\forall j \in [1, n]\ K_j \geq K + X_j.(-\infty)$

$\sum_j (1 - X_j) \geq 1$

Traduire si $(A > B)$ alors $X = 1$

Formulation

$B - A \geq X.(-\infty)$

Traduire si $(A \neq B)$ alors $X = 1$

Formulation

$B - A \geq X.(-\infty)$

$A - B \geq X.(-\infty)$

Traduire si $(A \geq B)$ alors $X = 1$

Formulation

$B - A - \varepsilon \geq X.(-\infty)$

Nota

pour des nombres entiers : si $(N_1 \geq N_2)$ alors $X = 1$, ε peut valoir 1

Traduire si $(A = B)$ alors $X = 1$

Méthode

si $(A \geq B)$ alors $X_1 = 1$, si $(B \geq A)$ alors $X_2 = 1$, puis $X = 1$ si X_1 *et* X_2 valent 1

Variables supplémentaires

variables binaires X_1 X_2

Formulation

$B - A - \varepsilon \geq X_1.(-\infty)$

$A - B - \varepsilon \geq X_2.(-\infty)$

$X_1 + X_2 \leq X + 1$

Nota

pour des nombres entiers : si $(N_1 = N_2)$ alors $X = 1$, ε peut valoir 1

Traduire si $(A \approx B)$ (tolérance t) alors $X = 1$

Méthode

si $(A \geq B - t)$ alors $X_1 = 1$, si $(B \geq A - t)$ alors $X_2 = 1$, puis $X = 1$ si X_1 *et* X_2 valent 1

Variables supplémentaires

variables binaires X_1 X_2

Formulation

$B - t - A - \varepsilon \geq X_1.(-\infty)$

$A - t - B - \varepsilon \geq X_2.(-\infty)$

$X_1 + X_2 \leq X + 1$

Nota

pour des nombres entiers : si $(N_1 \approx N_2)$ alors $X = 1$, ε peut valoir 1

Traduire la condition $X = (A > B)?$

Méthode

si $A > B$ alors $X = 1$ si $A \leq B$ alors $X = 0$

Formulation

$A - B - \varepsilon \geq (1 - X).(-\infty)$

$B - A \geq X.(-\infty)$

Nota

pour des nombres entiers $X = (N_1 > N_2)?$, ε peut valoir 1

Traduire la condition $X = (A \neq B)?$

Méthode

$X_1 = (A > B)?$, $X_2 = (B > A)?$ puis $X = X_1 \; ou \; X_2$

Variables supplémentaires

variables binaires X_1 X_2

Formulation

$A - B - \varepsilon \geq (1 - X_1).(-\infty)$

$B - A \geq X_1.(-\infty)$

$B - A - \varepsilon \geq (1 - X_2).(-\infty)$

$A - B \geq X_2.(-\infty)$

$X \leq X_1 + X_2$

$2X \geq X_1 + X_2$

Nota

pour des nombres entiers $X = (N_1 \neq N_2)?$, ε peut valoir 1

Traduire la condition $X = (A \geq B)?$

Méthode

si $A \geq B$ alors $X = 1$ si $A < B$ alors $X = 0$

Formulation

$A - B \geq (1 - X).(-\infty)$

$B - A - \varepsilon \geq X.(-\infty)$

Nota

pour des nombres entiers $X = (N_1 \geq N_2)?$, ε peut valoir 1

Traduire la condition $X = (A = B)?$

Méthode

$X_1 = (A \geq B)?$, $X_2 = (B \geq A)?$ puis $X = X_1 \; et \; X_2$

Variables supplémentaires

variables binaires X_1 X_2

Formulation

$A - B \geq (1 - X_1).(-\infty)$

$B - A - \varepsilon \geq X_1.(-\infty)$

$B - A \geq (1 - X_2).(-\infty)$

$A - B - \varepsilon \geq X_2.(-\infty)$

$2X \leq X_1 + X_2$

$X + 1 \geq X_1 + X_2$

Nota

pour des nombres entiers $X = (N_1 = N_2)?$, ε peut valoir 1

Traduire la condition $X = (A \approx B)?$ (tolérance t)

Méthode

$X_1 = (A \geq B - t)?$, $X_2 = (B \geq A - t)?$ puis $X = X_1 \; et \; X_2$

Variables supplémentaires

variables binaires X_1 X_2

Formulation

$A - B + t \geq (1 - X_1).(-\infty)$

$B - A - t - \varepsilon \geq X_1.(-\infty)$

$B - A + t \geq (1 - X_2).(-\infty)$

$A - B - t - \varepsilon \geq X_2.(-\infty)$

$2X \leq X_1 + X_2$

$X + 1 \geq X_1 + X_2$

Nota

pour des nombres entiers $X = (N_1 \approx N_2)?$, ε peut valoir 1

Quand une contrainte $A_j \geq B_j$ ne concerne pas un sous-ensemble S (dont les éléments vérifient $Y_j=1$)

Enoncé

$$Y_j = 0 \Rightarrow A_j \geq B_j$$

Formulation

$$\forall j \quad A_j \geq B_j + Y_j.(-\infty)$$

Quand une contrainte $A_j = B_j$ ne concerne pas un sous-ensemble S (dont les éléments vérifient $Y_j=1$)

Enoncé

$$Y_j = 0 \Rightarrow A_j = B_j$$

Formulation

$$\forall j \quad A_j \geq B_j + Y_j.(-\infty)$$

$$\forall j \quad B_j \geq A_j + Y_j.(-\infty)$$

Quand une contrainte $A_j \neq B_j$ ne concerne pas un sous-ensemble S (dont les éléments vérifient $Y_j=1$)

Enoncé

$$Y_j = 0 \Rightarrow A_j \neq B_j$$

Variable supplémentaire

variables binaires X_j

Formulation

$$\forall j \quad A_j \geq B_j + X_j.(-\infty) + \varepsilon + Y_j.(-\infty)$$

$$\forall j \quad B_j \geq A_j + (1 - X_j).(-\infty) + \varepsilon + Y_j.(-\infty)$$

Contraindre que parmi n contraintes $A_j \geq B_j$, au moins m soient vérifiées

Enoncé

$$\sum_j (A_j \geq B_j) \geq m$$

Méthode

si la contrainte n'est pas vérifiée alors Y_j est forcé à 1

Variables supplémentaires

n variables binaires Y_j

Formulation

$$\forall j \in [1,n] \;\; A_j \geq B_j + Y_j.(-\infty)$$

$$\sum_j (1 - Y_j) \geq m$$

Pénalités P pour celles non vérifiées $(Y_j = 1)$:

$$\min(co\hat{u}t + \sum_j P.Y_j)$$

Contraindre que parmi n contraintes $A_j = B_j$, au moins m soient vérifiées

Enoncé

$$\sum_j (A_j = B_j) \geq m$$

Variables supplémentaires

n variables binaires Y_j

Formulation

$$\forall j \in [1,n] \;\; A_j \geq B_j + Y_j.(-\infty)$$

$$\forall j \in [1,n] \;\; B_j \geq A_j + Y_j.(-\infty)$$

$$\sum_j (1 - Y_j) \geq m$$

Contraindre que parmi n contraintes $A_j \geq B_j$, au plus m soient vérifiées

Enoncé

$$\sum_j (A_j \geq B_j) \leq m$$

Méthode

si la contrainte inverse n'est pas vérifiée alors Y_j est forcé à 1

Variables supplémentaires

n variables binaires Y_j

Formulation

$$\forall j \in [1,n] \;\; B_j \geq A_j + \varepsilon + Y_j.(-\infty)$$

$$\sum_j (1 - Y_j) \geq n - m$$

Contraindre que parmi n contraintes $A_j = B_j$, au plus m soient vérifiées

Enoncé

$$\sum_j (A_j = B_j) \le m$$

Variables supplémentaires

n variables binaires X_j

n variables binaires Y_j

Formulation

$$\forall j \in [1,n] \;\; A_j \ge B_j + \varepsilon + X_j.(-\infty) + Y_j.(-\infty)$$

$$\forall j \in [1,n] \;\; B_j \ge A_j + \varepsilon + (1 - X_j).(-\infty) + Y_j.(-\infty)$$

$$\sum_j (1 - Y_j) \ge n - m$$

Contraindre que parmi n contraintes $A_j \ge B_j$, exactement m soient vérifiées

Enoncé

$$\sum_j (A_j \ge B_j) = m$$

Méthode

si la contrainte n'est pas vérifiée alors Y_j est forcé à 1

si la contrainte inverse n'est pas vérifiée alors Y_j est forcé à 0

Variables supplémentaires

n variables binaires Y_j

Formulation

$$\forall j \in [1,n] \;\; A_j \ge B_j + Y_j.(-\infty)$$

$$\forall j \in [1,n] \;\; B_j \ge A_j + \varepsilon + (1 - Y_j).(-\infty)$$

$$\sum_j (1 - Y_j) = m$$

Contraindre que parmi n contraintes $A_j = B_j$, exactement m soient vérifiées

Enoncé

$$\sum_j (A_j = B_j) = m$$

Variables supplémentaires

n variables binaires X_j

n variables binaires Y_j

Formulation

$$\forall j \in [1,n] \;\; A_j \ge B_j + Y_j.(-\infty)$$

$$\forall j \in [1,n] \;\; B_j \ge A_j + Y_j.(-\infty)$$

$$\forall j \in [1,n] \;\; A_j \ge B_j + \varepsilon + X_j.(-\infty) + (1 - Y_j).(-\infty)$$

$$\forall j \in [1,n] \;\; B_j \ge A_j + \varepsilon + (1 - X_j).(-\infty) + (1 - Y_j).(-\infty)$$

$$\sum_j (1 - Y_j) = m$$

Contraindre que l'ensemble *(E_i)*, soit inclus dans l'ensemble *(F_j)*

Enoncé

$$\forall i \in [1,n]\ \exists j \in [1,m]\ /\ E_i \neq F_j$$

Variables supplémentaires

n.m variables binaires Y_{ij}

Formulation

$$\forall i \in [1,n]\ \forall j \in [1,m]\ \ E_i \geq F_j + (1 - Y_{ij}).(-\infty)$$

$$\forall i \in [1,n]\ \forall j \in [1,m]\ \ F_j \geq E_i + (1 - Y_{ij}).(-\infty)$$

$$\sum_i \sum_j Y_{ij} = n$$

Traduire X = (l'ensemble *(E_i)*, est inclus dans l'ensemble *(F_j)*) ?

Variables supplémentaires

n.m variables binaires X_{ij}

n.m variables binaires Y_{ij}

Formulation

$$\forall i \in [1,n]\ \forall j \in [1,m]\ \ E_i \geq F_j + (1 - Y_{ij}).(-\infty)$$

$$\forall i \in [1,n]\ \forall j \in [1,m]\ \ F_j \geq E_i + (1 - Y_{ij}).(-\infty)$$

$$\forall i \in [1,n]\ \forall j \in [1,m]\ \ E_i \geq F_j + \varepsilon + X_{ij}.(-\infty) + Y_{ij}.(-\infty)$$

$$\forall i \in [1,n]\ \forall j \in [1,m]\ \ F_j \geq E_i + \varepsilon + (1 - X_{ij}).(-\infty) + Y_{ij}.(-\infty)$$

$$1 - X \leq n - \sum_i \sum_j Y_{ij}$$

$$n(1 - X) \geq n - \sum_i \sum_j Y_{ij}$$

Contraindre que l'ensemble *(E_i)*, soit différent de l'ensemble *(F_j)*

Enoncé

$$\exists i \in [1,n]\ \forall j \in [1,n]\ /\ E_i \neq F_j$$

Variables supplémentaires

n.m variables binaires Y_{ij}

Formulation

$$\forall i \in [1,n]\ \forall j \in [1,n]\ \ E_i \geq F_j + (1 - Y_{ij}).(-\infty)$$

$$\forall i \in [1,n]\ \forall j \in [1,n]\ \ F_j \geq E_i + (1 - Y_{ij}).(-\infty)$$

$$\sum_i \sum_j Y_{ij} \leq n - 1$$

Traduire X = (l'ensemble *(E_i)*, est différent de l'ensemble *(F_j)*) ?

Variables supplémentaires

n.m variables binaires X_{ij}

n.m variables binaires Y_{ij}

Formulation

$$\forall i \in [1,n]\ \forall j \in [1,n]\ \ E_i \geq F_j + (1 - Y_{ij}).(-\infty)$$

$$\forall i \in [1,n]\ \forall j \in [1,n]\ \ F_j \geq E_i + (1 - Y_{ij}).(-\infty)$$

$$\forall i \in [1,n]\ \forall j \in [1,n]\ \ E_i \geq F_j + \varepsilon + X_{ij}.(-\infty) + Y_{ij}.(-\infty)$$

$$\forall i \in [1,n]\ \forall j \in [1,n]\ \ F_j \geq E_i + \varepsilon + (1 - X_{ij}).(-\infty) + Y_{ij}.(-\infty)$$

$$X \leq n - \sum_i \sum_j Y_{ij}$$

$$nX \geq n - \sum_i \sum_j Y_{ij}$$

Contraindre que le n-uplet (U_j), soit différent du n-uplet (V_j)

Enoncé

$$\exists j \in [1,n] \ / \ U_j \neq V_j$$

Variables supplémentaires

n variables binaires Y_j

Formulation

$$\forall j \in [1,n] \ U_j \geq V_j + (1 - Y_j).(-\infty)$$

$$\forall j \in [1,n] \ V_j \geq U_j + (1 - Y_j).(-\infty)$$

$$\sum_j Y_j \leq n - 1$$

Contraindre que le n-uplet (U_i), soit inclus dans le m-uplet (V_j)

Enoncé

$$\forall i \in [1,n] \ \exists j \in [1,m] \ / \ U_i = V_j$$

$$\forall i_1 \in [1,n] \ \forall i_2 \in [1,n] \ i_1 > i_2 \Rightarrow j_1 > j_2$$

Variables supplémentaires

$n.m$ variables binaires Y_{ij}

Formulation

$$\forall i \in [1,n] \ \forall j \in [1,m] \ U_i \geq V_j + (1 - Y_{ij}).(-\infty)$$

$$\forall i \in [1,n] \ \forall j \in [1,m] \ V_j \geq U_i + (1 - Y_{ij}).(-\infty)$$

$$\forall i \in [1,n-1] \sum_j (Y_{ij} - Y_{i+1\,j}).b^{(m-j)} \geq \varepsilon \ \text{(se } n > 1\text{) kun } b > 1 + \varepsilon$$

$$\forall i \in [1,n] \sum_j Y_{ij} = 1$$

Nota

si m n'est pas trop grand, il suffit de prendre b =2 (base de numération rapide et simple), sinon prendre une valeur de b plus petite

Contraindre que le n-uplet (U_j), soit antérieur au n-uplet (V_j) (au sens ordre "alphabétique")

Enoncé

$$\exists n' \in [1,n] \ / \ \forall j \in [1,n'] \ U_j \geq V_j \text{ et } (U_{n'} > V_{n'} \text{ ou } n'=n)$$

Variables supplémentaires

n variables binaires Y_j et n variables binaires X'_j

Formulation

$$\forall j \in [1,n] \ U_j \geq V_j + (1 - Y_j).(-\infty)$$

$$\forall j \in [1,n] \ V_j \geq U_j + \varepsilon + Y_j.(-\infty)$$

$$\forall j \in [1,n] \ V_j \geq U_j + (1 - Y'_j).(-\infty)$$

$$\forall j \in [1,n] \ U_j \geq V_j + \varepsilon + Y'_j.(-\infty)$$

$$\sum_j (Y_j - Y'_j).b^{(n-j)} \geq 0 \ \text{ kun } b > 1$$

Nota

si m n'est pas trop grand, il suffit de prendre b =2 (rapidité et simplicité), sinon prendre une valeur de b plus petite.

Traduire X = (le n-uplet (U_i) est inclus dans le m-uplet (V_j)) ?

Variables supplémentaires

$n.m$ variables binaires Y_{ij}

n variables binaires X_i

Formulation

$\forall i \in [1,n]\ \forall j \in [1,m]\ \ U_i \geq V_j + (1-Y_{ij}).(-\infty)$

$\forall i \in [1,n]\ \forall j \in [1,m]\ \ V_j \geq U_i + (1-Y_{ij}).(-\infty)$

$\forall i \in [1,n-1] \sum_j (Y_{ij} - (b-1).\sum_{k=1}^{n-i} Y_{i+k\ j}).b^{(m-j)} \geq \varepsilon + X_i.(-\infty)$ (si $n > 1$) avec $b > 1$

$\forall i \in [1,n] \sum_j Y_{ij} \leq 1$

$\forall i \in [1,n] \sum_j Y_{ij} \geq 1 + X_i.(-\infty)$

$\forall i \in [1,n]\ 1 - \sum_j Y_{ij} - \varepsilon \geq (1 - X_i).(-\infty)$

$\min(\sum X_i)$

$1 - X \leq \sum X_i$

$n\ (1 - X) \geq \sum X_i$

Nota

si m n'est pas trop grand, il suffit de prendre b =2 (rapidité et simplicité) ; le terme (b-1) peut alors disparaître, sinon prendre une valeur de b plus petite.

Déterminer le nombre minimum de transformations pour passer du *n*-uplet *(U_i)* au *m*-uplet *(V_j)*

Variables supplémentaires

n.m variables binaires Y_{ij}

n variables binaires X_i

Formulation

$$\forall i \in [1,n]\ \forall j \in [1,m]\ \ U_i \geq V_j + (1 - Y_{ij}).(-\infty)$$

$$\forall i \in [1,n]\ \forall j \in [1,m]\ \ V_j \geq U_i + (1 - Y_{ij}).(-\infty)$$

$$\forall i \in [1,n-1]\ \sum_j (Y_{ij} - (b-1).\sum_{k=1}^{n-i} Y_{i+k\ j}).b^{(m-j)} \geq \varepsilon + X_i.(-\infty)\ \ (\text{si } n > 1) \text{ avec } b > 1$$

$$\forall i \in [1,n]\ \sum_j Y_{ij} \leq 1$$

$$\forall i \in [1,n]\ \sum_j Y_{ij} \geq 1 + X_i.(-\infty)$$

$$\forall i \in [1,n]\ 1 - \sum_j Y_{ij} - \varepsilon \geq (1 - X_i).(-\infty)$$

$$\min(2.\sum X_i + m - n)$$

Nota

si *m* n'est pas trop grand, il suffit de prendre *b* =2 (rapidité et simplicité) ; le terme (*b*-1) peut alors disparaître, sinon prendre une valeur de *b* plus petite.

$2.\sum X_i + m - n$ représente la distance de Levenshtein entre *(U_i)* et *(V_j)*

Par exemple : d("NICHE", "CHIENS")=5 (=2x2+6-5)

A noter que : d("CHIENS", "NICHE")=5 (=2x3+5-6)

Bien entendu, pour des n-uplets connus, soit d(*(u_i)*, *(v_j)*), il suffit de poser :

$$\forall i \in [1,n]\ \forall j \in [1,m]\ u_i \neq v_j \Rightarrow X_{ij} = 0$$

Contraindre que $X = X_1 \; ou \; X_2$

Formulation

$$X \leq X_1 + X_2$$
$$2X \geq X_1 + X_2$$

Contraindre que $X = X_1 \; ou \; X_2 \; ou \; \ldots X_n$

Formulation

$$X \leq \sum X_i$$
$$nX \geq \sum X_i$$

Contraindre que $X = X_1 \; et \; X_2$

Formulation

$$2X \leq X_1 + X_2$$
$$X + 1 \geq X_1 + X_2$$

Contraindre que $X = X_1 \; et \; X_2 \; et \; \ldots et \; X_n$

Formulation

$$nX \leq \sum X_i$$
$$X + n - 1 \geq \sum X_i$$

Contraindre que $X = X_1 \wedge X_2$

Méthode

$$X = ((X_1 \; et \; \overline{X_2}) \; ou \; (\overline{X_1} \; et \; X_2))$$

Variables supplémentaires

variables binaires X_3 X_4

Formulation

$$2X_3 \leq X_1 - X_2 + 1$$
$$X_3 + 1 \geq X_1 - X_2 + 1$$
$$2X_4 \leq X_2 - X_1 + 1$$
$$X_4 + 1 \geq X_2 - X_1 + 1$$
$$X \leq X_3 + X_4$$
$$2X \geq X_3 + X_4$$

Nota

la formulation n'est pas facilement généralisable à n booléens

Contraindre que $X = X_1 \wedge X_2$ (variante simplifiée)

Formulation

$$X \leq X_1 + X_2$$
$$X \leq 2 - X_1 - X_2$$
$$X \geq X_1 - X_2$$
$$X \geq X_2 - X_1$$

Nota

la variante de formulation est donc plus simple dans le cas du "ou exclusif"

la formulation n'est pas facilement généralisable à n booléens

Contraindre que $X = X_1 \wedge X_2 \wedge \ldots \wedge X_n$

Méthode

un et un seul X_i vaut 1 parmi n (voir plus loin)

Variables supplémentaires

variables binaires Y_1 Y_2

Formulation

$$\sum X_i \leq nY_1$$
$$\sum X_i \geq Y_1$$
$$\sum X_i \geq 2 - 2Y_2$$
$$\sum X_i \leq n - (n-1)Y_2$$
$$2X \leq Y_1 + Y_2$$
$$X + 1 \geq Y_1 + Y_2$$

Nota

on retrouve des formules similaires en faisant $X = (\sum X_i = 1)$?

Traduire "si X alors $A \geq B$ sinon $C \geq D$"

Formulation

$$A - B \geq (1 - X).(-\infty)$$
$$C - D \geq X.(-\infty)$$

Traduire "si X alors $A = B$ sinon $C = D$"

Formulation

$$A - B \geq (1 - X).(-\infty)$$
$$B - A \geq (1 - X).(-\infty)$$
$$C - D \geq X.(-\infty)$$
$$D - C \geq X.(-\infty)$$

Traduire "si X_1 *et* X_2 alors $A \geq B$ sinon $C \geq D$"

Formulation

$$A - B \geq (2 - \mathrm{X}_1 - X_2).(-\infty)$$
$$C - D \geq (1 + X_1 - X_2).(-\infty)$$
$$C - D \geq (1 + X_2 - X_1).(-\infty)$$
$$C - D \geq (X_1 + X_2).(-\infty)$$

Traduire "si X_1 *ou* X_2 alors $A \geq B$ sinon $C \geq D$"

Formulation

$$A - B \geq (2 - \mathrm{X}_1 - X_2).(-\infty)$$
$$A - B \geq (1 + X_1 - X_2).(-\infty)$$
$$A - B \geq (1 + X_2 - X_1).(-\infty)$$
$$C - D \geq (X_1 + X_2).(-\infty)$$

Traduire "si $X_1 \wedge X_2$ alors $A \geq B$ sinon $C \geq D$"

Formulation

$$C - D \geq (2 - \mathrm{X}_1 - X_2).(-\infty)$$
$$A - B \geq (1 + X_1 - X_2).(-\infty)$$
$$A - B \geq (1 + X_2 - X_1).(-\infty)$$
$$C - D \geq (X_1 + X_2).(-\infty)$$

$X=1$ si au moins m des n éléments X_i d'un ensemble sont pris ($X_i=1$)

Enoncé

$$\sum_{i=1}^{n} X_i \geq m \Rightarrow X = 1$$

Formulation

$$\sum X_i \leq (n-m+1)X + m - 1$$

$X=1$ que si au moins m des n éléments X_i d'un ensemble sont pris ($X_i=1$)

Enoncé

$$\sum_{i=1}^{n} X_i \geq m \Leftrightarrow X = 1$$

Formulation

$$\sum X_i \leq (n-m+1)X + m - 1$$

$$\sum X_i \geq mX$$

$X=1$ si au plus m des n éléments X_i d'un ensemble sont pris ($X_i=1$)

Enoncé

$$\sum_{i=1}^{n} X_i \leq m \Rightarrow X = 1$$

Formulation

$$\sum X_i \geq m + 1 - (m+1)X$$

$X=1$ que si au plus m des n éléments X_i d'un ensemble sont pris ($X_i=1$)

Enoncé

$$\sum_{i=1}^{n} X_i \leq m \Leftrightarrow X = 1$$

Formulation

$$\sum X_i \geq m + 1 - (m+1)X$$

$$\sum X_i \leq n - (n-m)X$$

$X=1$ si exactement m des n éléments X_i d'un ensemble sont pris ($X_i=1$)

Enoncé

$$\sum_{i=1}^{n} X_i = m \Rightarrow X = 1$$

Variables supplémentaires

variables binaires Y_1 Y_2

Formulation

$$\sum X_i \leq (n-m+1)Y_1 + m - 1$$
$$\sum X_i \geq m + 1 - (m+1)Y_2$$
$$2X \leq Y_1 + Y_2$$
$$X + 1 \geq Y_1 + Y_2$$

$X=1$ que si exactement m des n éléments X_i d'un ensemble sont pris ($X_i=1$)

Enoncé

$$\sum_{i=1}^{n} X_i = m \Leftrightarrow X = 1$$

Variables supplémentaires

variables binaires Y_1 Y_2

Formulation

$$\sum X_i \leq (n-m+1)Y_1 + m - 1$$
$$\sum X_i \geq mY_1$$
$$\sum X_i \geq m + 1 - (m+1)Y_2$$
$$\sum X_i \leq n - (n-m)Y_2$$
$$2X \leq Y_1 + Y_2$$
$$X + 1 \geq Y_1 + Y_2$$

Contraindre qu'un sous-ensemble S' (dont les éléments vérifient $X_i=1$) d'un ensemble E, soit inclus dans un sous-ensemble S

Enoncé

$$S'_{X_i=1} \subset S$$

Formulation

$$\sum_{E-S} X_i = 0$$

Contraindre qu'un sous-ensemble S' (dont les éléments vérifient $X_i=1$) d'un ensemble E, ne soit pas inclus dans un sous-ensemble S

Enoncé

$$S'_{X_i=1} \not\subset S$$

Formulation

$$\sum_{E-S} X_i \geq 1$$

Contraindre qu'un sous-ensemble S' (dont les éléments vérifient $X_i=1$) d'un ensemble E, soit un sur-ensemble d'un sous-ensemble S

Enoncé

$$S \subset S'_{X_i=1}$$

Formulation

$$\sum_{S} X_i = card(S)$$

Contraindre qu'un sous-ensemble S' (dont les éléments vérifient $X_i=1$) d'un ensemble E, ne soit pas un sur-ensemble d'un sous-ensemble S

Enoncé

$$S \not\subset S'_{X_i=1}$$

Formulation

$$\sum_{S} X_i \leq card(S) - 1$$

Contraindre qu'un sous-ensemble S' (dont les éléments vérifient $X_i=1$) d'un ensemble E, soit différent d'un sous-ensemble S donné

Enoncé

$$S'_{X_i=1} \neq S$$

Formulation

$$\sum_{S} X_i \leq \sum_{E-S} X_i + card(S) - 1$$

Trouver un sous-ensemble d'un ensemble E dont la somme des quantités q_i associées à ses éléments s'approche d'une valeur q_{cible} moyennant une tolérance t

Variable supplémentaire

variable réelle Q_{ecart}

Formulation

$$\sum_E X_i.q_i - Q_{ecart} \leq q_{cible} + t$$

$$\sum_E X_i.q_i + Q_{ecart} \geq q_{cible} - t$$

objectif : $\min(Q_{ecart})$

Trouver une configuration de n sous-ensembles disjoints et complémentaires d'un ensemble E dont les sommes des quantités q_i associées aux éléments aient des valeurs aussi proches que possible entre elles

Variable supplémentaire

variable réelle Q_{ecart}

Formulation

$\forall j \in [1, n]$

$$\sum_i X_{ij}.q_i - Q_{ecart} \leq q_{moyen}$$

$$\sum_i X_{ij}.q_i + Q_{ecart} \geq q_{moyen}$$

$\forall i \leq card(E)$ $\sum_j X_{ij} = 1$ (disjoints et complémentaires)

objectif : $\min(Q_{ecart})$

Nota

q_{moyen} peut être remplacé par q_j (valeur cible de chaque sous-ensemble),

avec $\sum_j q_j = \sum_i q_i$

Décrire une fonction (*k→f(k)*) continue linéaire par morceaux

Enoncé

$$\forall i \in [1,n]\ k \in [k_i, k_{i+1}] \Rightarrow f(k) = a_i k + b_i \text{ avec } k \in [k_1, k_{n+1}]$$

Variables supplémentaires

n variables réelles K_i

n +1 variables binaires X_i

Formulation

$$\sum_i K_i = k$$

$$\forall i \in [1,n]\ X_{i+1} \leq X_i$$

$$\forall i \in [1,n]\ 0 \leq K_i \leq (k_{i+1} - k_i).X_i$$

$$\forall i \in [1,n]\ (k_{i+1} - k_i).X_{i+1} \leq K_i$$

$$f(k) = \sum_i a_i.K_i + a_1 k_1 + b_1$$

Nota 1

Si la fonction est non continue linéaire par morceaux :

$$f(k) = \sum_i a_i.K_i + \sum_i ((a_i - a_{i-1}).k_i + b_i - b_{i-1}).X_i$$, avec $a_0 = b_0 = 0$

Nota 2

$$f'(k) = \sum_i a_i.(X_i - X_{i+1})$$

Décomposer n en nombres premiers (avec connaissance préalable des nombres premiers inférieurs)

Méthode

$$n = \prod p_i^{N_i} \text{ , d'où } \log(n) = \sum N_i . \log(p_i)$$

Formulation

$$\log(n) - \sum N_i . \log(p_i) \leq \varepsilon$$

$$\sum N_i . \log(p_i) - \log(n) \leq \varepsilon$$

Nota

s'il n'y a pas de solution, n est un nouveau nombre premier ou un multiple d'un nouveau nombre premier. Il est très simple de compléter les contraintes, pour positionner une variable binaire X donnant ce résultat

ε est destiné à palier d'éventuels arrondis de la fonction log. Il doit être inférieur à log*(n+1)*-log*(n)*=log*(1+1/n),* soit *1/((n+1) Log2).* Prendre ε *=(0,5 . 1,442695)/(n+1)*

Décomposer n en nombres premiers (sans connaissance préalable des nombres premiers inférieurs)

Méthode

plus la décomposition est fine (nombres premiers) plus la somme des exposants est élevée

Formulation

$$\log(n) - \sum N_i . \log(n_i) \leq \varepsilon$$

$$\sum N_i . \log(n_i) - \log(n) \leq \varepsilon$$

$$\max(\sum N_i)$$

Nota

les nombres n_i proposés sont tous les nombres de 2 à n

par souci d'efficacité, on peut ôter de la liste tous les multiples de 2,3,5 à l'exception de 2,3,5 et de n lui-même

Le dessous des cartes

Enoncé

Trois cartes de couleurs différentes : un *5*, un *dix*, un *roi*.
Un *5* est à droite d'un *roi*.
Un *trèfle* est à gauche d'un *pique*.
Un *10* est à gauche d'un *cœur*.
Un *coeur* est à gauche d'un *pique*.
Dans quel ordre se trouvent les têtes et les couleurs ?

Variables (toutes entre 0 et 2)

X_5, X_{10}, X_{roi}
$X_{trèfle}, X_{pique}, X_{coeur}$

Formulation

$$X_5 \geq X_{roi} + 1$$
$$X_{trèfle} \leq X_{pique} - 1$$
$$X_{10} \leq X_{coeur} - 1$$
$$X_{coeur} \leq X_{pique} - 1$$

Nota

Il n'est pas ici nécessaire d'exprimer que les trois figures (resp. les trois couleurs) sont toutes différentes. Mais en général, cela est nécessaire, au moins partiellement. Par exemple, ne sont pas explicitement esprimés que *10* et *5,* resp. *coeur* et *trèfle, roi* et *10* sont à des places différentes.
Proposition : Un seul est inférieur à 0 et la somme vaut 3. Ainsi, les trois valeurs sont 0, 1 et 2.

Résultat

$X_{10} = 0$, $X_{roi} = 1$, $X_5 = 2$
$X_{trèfle} = 0$, $X_{coeur} = 1$, $X_{pique} = 2$

Le dessous des cartes

Enoncé

Quatre cartes de couleurs différentes : un *valet*, une *dame*, un *roi*, un *as*.
Un *pique* est à côté d'un *carreau*.
Un *trèfle* est immédiatement à gauche du *roi* ou de la *dame*.
Le *roi* est immédiatement à droite d'une carte rouge.
La carte la plus à droite n'est pas un *cœur*.
L'une des deux cartes du milieu est le *valet*.
Le *roi* et la *dame* ne sont pas l'un à côté de l'autre.
Dans quel ordre se trouvent les têtes et les couleurs ?

Variables (toutes entre 0 et 3)

X_{valet}, X_{dame}, X_{roi}, X_{as}
$X_{trèfle}$, X_{pique}, $X_{carreau}$, X_{coeur}

Variables supplémentaires

X_1, X_2,
X_5, X_6,
X_9

Formulation

Un pique est à côté d'un carreau

$$X_{pique} \leq X_{carreau} + 1$$
$$X_{carreau} \leq X_{pique} + 1$$

Un trèfle est immédiatement à gauche du roi ou de la dame

$$X_{trèfle} \geq X_{roi} - 1 + X_1.(-\infty)$$
$$X_{roi} - 1 \geq X_{trèfle} + X_1.(-\infty)$$
$$X_{trèfle} \geq X_{dame} - 1 + X_2.(-\infty)$$
$$X_{dame} - 1 \geq X_{trèfle} + X_2.(-\infty)$$
$$X_1 + X_2 = 1$$

Le roi est immédiatement à droite d'une carte rouge

$$X_{roi} \geq X_{coeur} + 1 + X_5.(-\infty)$$
$$X_{coeur} + 1 \geq X_{roi} + X_5.(-\infty)$$
$$X_{roi} \geq X_{carreau} + 1 + X_6.(-\infty)$$
$$X_{carreau} + 1 \geq X_{roi} + X_6.(-\infty)$$
$$X_5 + X_6 = 1$$

La carte la plus à droite n'est pas un cœur

$$X_{coeur} \leq 2$$

L'une des deux cartes du milieu est le valet

$$X_{valet} \leq 2$$
$$X_{valet} \geq 1$$

Le roi et la dame ne sont pas l'un à côté de l'autre

$$X_{roi} \geq X_{dame} + 2 + X_9.(-\infty)$$
$$X_{dame} \geq X_{roi} + 2 + (1 - X_9).(-\infty)$$

Nota

Il est nécessaire d'exprimer que les quatre figures (resp. les quatre couleurs) sont toutes différentes (voir le problème précédent). En particulier X_{as} qui n'apparaît dans aucune contrainte. Cela n'est pas décrit ci-avant (sauf pour X_{dame} et X_{roi}).

Proposition : un seul est inférieur à 0, au moins un est supérieur 3 et la somme vaut 6. Ainsi, les quatre valeurs sont 0, 1, 2 et 3.

Résultat

$X_{as} = 0, X_{dame} = 1, X_{valet} = 2, X_{roi} = 3$
$X_{trèfle} = 0, X_{coeur} = 1, X_{carreau} = 2, X_{pique} = 3$
et :
$X_1 = 1, X_2 = 0$
$X_5 = 1, X_6 = 0$
$X_9 = 0$

Fioles à partager

Enoncé

21 fioles : 7 pleines, 7 à moitié vides, 7 vides
3 personnes doivent recevoir chacune 7 fioles et la même quantité de liquide

Variables

P1 , P2, P3 (pleines)
M1, M2, M3 (moitié pleines)
V1, V2, V3 (vides)

Formulation

$P1 + M1 + V1 = 7$
$P2 + M2 + V2 = 7$
$P3 + M3 + V3 = 7$

$P1 + 0.5\ M1 = 3.5$
$P2 + 0.5\ M2 = 3.5$
$P3 + 0.5\ M3 = 3.5$

$P1 + P2 + P3 = 7$
$M1 + M2 + M3 = 7$
$V1 + V2 + V3 = 7$

Résultat

$P1=1\ ;\ M1=5\ ;\ V1=1$
$P2=3\ ;\ M2=1\ ;\ V2=3$
$P3=3\ ;\ M3=1\ ;\ V3=3$

Traversée de pont

Enoncé

Quatre personnes (A, B, C, D) doivent traverser une rivière
Il mettent les temps suivants : A (1 mn), B (2 mn), C (5 mn), D (10 mn)
Deux personnes (i et j) au plus sur la barque ($t=\sup(t_i,t_j)$)
La barque doit contenir au moins une personne (en particulier pour revenir)
Quel est le temps minimal pour faire passer tout le monde ?

Variables

X_{ij} : personne j au trajet i (j de 1 à 4 ; i de 1 à 5)
$i = 1, 3$ ou 5 : aller
$i = 2$ ou 4 : retour
D_1, D_2, D_3, D_4, D_5 : durées des trajets

Variables supplémentaires

$Y_{1j}, Y_{3j}, Y_{5j},$

Formulation

$$\forall i \in [1,5] \sum_j X_{ij} \leq 2$$
$$\forall i \in \{2,4\} \sum_j X_{ij} \geq 1$$
$$\forall i \in \{1,3,5\}\ \forall j \in [1,4]\ X_{ij}.t_j \leq D_i$$
$$\forall i \in \{1,3,5\}\ \forall j \in [1,4]\ X_{ij}.t_j \geq D_i - Y_{ij}.(-\infty)$$
$$\forall i \in \{1,3,5\} \sum_j Y_{ij} = 1$$
$$\forall i \in \{2,4\} \sum_j X_{ij}.t_j = D_i$$
$$\forall j \in [1,4]\ X_{1j} - X_{2j} \geq 0$$
$$\forall j \in [1,4]\ X_{1j} + X_{3j} - X_{2j} - X_{4j} \geq 0$$
$$\forall j \in [1,4]\ X_{1j} + X_{3j} + X_{5j} - X_{2j} - X_{4j} = 1$$

min ($D_1+D_2+D_3+D_4+D_5$)

Résultat

$D_1 = 2, D_2 = 1, D_3 = 10, D_4 = 2, D_5 = 2$
Temps total : 17 mn
$X_{11} = X_{12} = 1$ (A et B traversent)
$X_{21} = 1$ (A revient)
$X_{33} = X_{34} = 1$ (C et D traversent)
$X_{42} = 1$ (B revient)
$X_{51} = X_{52} = 1$ (A et B traversent)

Derrière la porte

Enoncé

Un homme doit choisir entre deux portes à ouvrir, sachant que derrière ces portes il y a un dragon ou une princesse. Mais il peut y avoir deux princesses, ou deux dragons... S'il choisit la princesse, il peut l'épouser, mais s'il choisit le dragon, il sera grillé ... Sur chaque porte, il y a une affiche, et on sait que soit les affiches disent toutes les deux la vérité, soit elles mentent toutes les deux.

Sur la porte 1 est inscrit : il y a un dragon dans cette cellule ou une princesse dans l'autre.

Sur la porte 2 est inscrit : il y a une princesse dans l'autre cellule.

Que contient la première cellule ? Et la seconde ?

Variables

Xp_1 : porte 1 : 0 si dragon, 1 si princesse
Xp_2 : porte 2 : 0 si dragon, 1 si princesse
Xv_1 : assertion 1 : 0 si fausse, 1 si vraie
Xv_2 : assertion 2 : 0 si fausse, 1 si vraie

Variables supplémentaires

X_1, X_2, X_3, X_4
X_5, X_6

Formulation

$(Xv_1 \,\&\, Xv_2) \mid (!Xv_1 \,\&\, !Xv_2) = \text{vrai}$
$(Xv_1 \,\&\, (!Xp_1 \mid Xp_2)) \mid (!Xv_1 \,\&\, (Xp_1 \,\&\, !Xp_2)) = \text{vrai}$
$(Xv_2 \,\&\, Xp_1) \mid (!Xv_2 \,\&\, !Xp_1) = \text{vrai}$

soit en plus simple :

$Xv_1 + Xv_2 \mathrel{!}= 1$
$Xv_1 + ((1\text{-}Xp_1) \mid Xp_2) \mathrel{!}= 1$
$Xv_2 + Xp_1 \mathrel{!}= 1$

soit en plus simple :

$Xv_1 + Xv_2 \mathrel{!}= 1$
$Xv_1 + sup\,((1\text{-}Xp_1), Xp_2) \mathrel{!}= 1$
$Xv_2 + Xp_1 \mathrel{!}= 1$

soit en formulation linéaire :

$$Xv_1 + Xv_2 \geq 1 + 1 + X_1.(-\infty)$$
$$Xv_1 + Xv_2 \leq 1 - 1 + (1 - X_1).(-\infty)$$

$$X_2 \geq 1 - Xp_1$$
$$X_2 \geq Xp_2$$
$$1 - Xp_1 \geq X_2 + X_5.(-\infty)$$
$$Xp_2 \geq X_2 + X_6.(-\infty)$$
$$X_5 + X_6 \leq 1$$
$$Xv_1 + X_2 \geq 1 + 1 + X_3.(-\infty)$$
$$Xv_1 + X_2 \leq 1 - 1 + (1 - X_3).(-\infty)$$

$$Xv_2 + Xp_1 \geq 1 + 1 + X_4.(-\infty)$$
$$Xv_2 + Xp_1 \leq 1 - 1 + (1 - X_4).(-\infty)$$

Résultat

Il y a une princesse dans chacune des deux cellules. Les deux assertions sont vraies.

$Xp_1 = Xv_1 = Xp_2 = Xv_2 = 1$

$X_1 = X_2 = X_3 = X_4 = X_5 = 1$; $X_6 = 0$

Neuf + deux = onze (cryptarithme)

Enoncé

Remplacer chacune des huit lettres présentes dans ces trois mots par des chiffres de telle façon que la somme des nombres soit vérifiée.

Variables (binaires)

X_{ij} lien entre la lettre i (de 1 à 8 : *d, e, f, n, o, u, x, z*) et le chiffre j (de 0 à 9)

Y_i (i de 1 à 7)

Formulation

Un seul chiffre par lettre

$$\forall i \in [1,8] \sum_j X_{ij} = 1$$

Au plus une lettre par chiffre

$$\forall j \in [0,9] \sum_i X_{ij} \leq 1$$

La somme est vérifiée

$$\sum_j (X_{7j} + X_{3j} - X_{2j} + 10.(2X_{6j} - X_{8j}) + 100.(2X_{2j} - X_{4j}) + 1000.(X_{1j} + X_{4j} - X_{5j})) = 0$$

Pour que la solution trouvée (il y en a 44) soit la plus proche de l'ordre alphabétique

$$\forall i \in [1,7] \; \forall k \in [i+1,8] \; \sum_j 2^{(7-j)}(X_{ij} - X_{kj}) \geq (1 - Y_i)(-\infty)$$

$$\max(\sum_{i=1}^{7} Y_i)$$

Résultat

6345 + 1348 = 7693 soit : d=1, e=3, f=5, n=6, o=7, u=4, x=8, z=9

Christian Rivière
première version : 2011
mise à jour : 13 août 2013
dernière mise à jour : 17 mai 2024