Génération rapide de scénarios géophysiques par satisfaction de contraintes pour la localisation des séismes

Jean-Philippe Poli[1] **David Mercier**[1] **Anthony Larue**[1]
Carole Maillard[2] **Jocelyn Guilbert**[2]

[1]CEA, LIST, Laboratoire Intelligence Multicapteurs et Apprentissage
F-91191 Gif-sur-Yvette, France
[2]CEA, DAM, DIF, F-91297 Arpajon, France.

{prenom.nom}@cea.fr

Résumé

Le but du travail que nous menons est d'améliorer un système déjà existant dont le but est la localisation automatique des séismes. La localisation d'un événement à l'aide de ce système se déroule en trois temps. Dans un premier temps, les différents signaux collectés par les stations sismiques sont utilisés pour détecter les différentes heures d'arrivées des phases sismiques. Ces phases sont ensuite classées en cinq classes en fonction de la distribution temps-fréquence de leur signal. Enfin, la position de l'événement sismique est déduite de cet étiquetage des phases. Malheureusement, les heures d'arrivées des phases, appelées *pointés*, peuvent être imprécises ce qui a pour effet de perturber l'étiquetage des phases. Ces erreurs affectent la localisation du séisme du fait qu'elles mènent à des scénarios géophysiquement impossibles.

Afin d'améliorer le système existant, nous proposons d'insérer une phase supplémentaire qui consiste à déterminer l'étiquetage des phases le plus vraisemblable possible à partir des sorties des classifieurs. L'idée sous-jacente est d'étiqueter chacune des phases en considérant toutes les autres phases plutôt que d'étiqueter chacune des phases indépendamment les unes des autres, afin de s'assurer de la cohérence de l'ensemble des étiquettes. Pour cela, nous utilisons plusieurs solveurs de contraintes afin de prédire tous les étiquetages géophysiquement possibles de l'ensemble du réseau de stations. La principale difficulté d'une telle approche est en fait la taille de l'espace de recherche. En effet, pour un séisme moyen, le pointeur automatique fournit 4 pointés pour une vingtaine de stations. Puisque chaque pointé appartient à une des 5 classes, le nombre de scénarios à examiner est de l'ordre de $5^{4\times 20} \approx 8.3 \times 10^{55}$. Le cadre des CSP va nous permettre de diminuer considérablement cet espace de recherche en ne tenant plus compte des scénarios géophysiquement impossibles en réduisant sa taille à 10^{40} environs.

Dans cet article, nous décrivons la méthode retenue en insistant sur une contrainte globale qui est proche de la subsumption en logique et notre choix de contraintes nous permettant de générer une représentation condensée de l'ensemble des solutions en 25 secondes dans le pire des cas.

Abstract

The goal of our work is to improve an existing automatic location system for seismic surveillance. The present system is a three-step workflow. Firstly, it uses the signal from several stations which collect the waves of the event in order to determine the arrival times of the different phases. It then classifies the different phases into five classes, depending on the time-frequency distribution of their signal. Finally, it deduces the position of the seismic event from these labelled phases. Unfortunately, the automatic arrival times may be inaccurate and the classifier may misclassify some phases. All these mistakes strongly affect the location of the event because they can lead to impossible geophysical scenarios.

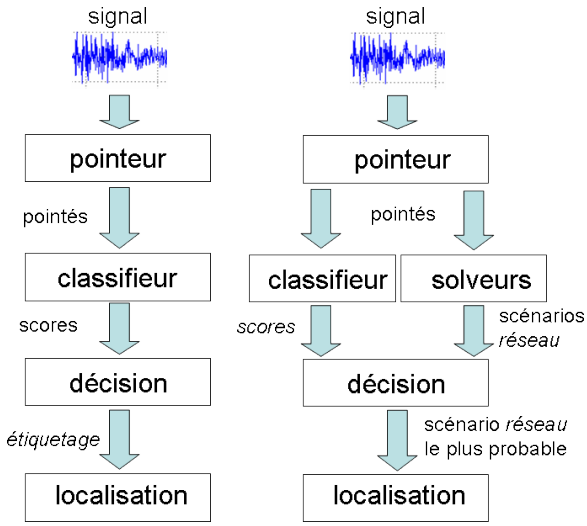
In order to improve the current system, we propose to insert another step which consists in determining the most probable phase labelling from the classifier outputs. This idea basically consists in considering the labelling of all the phases of the network at the same time instead of considering each phase individually, in order to ensure the geophysical consistency of the overall labelling. The main difficulty is the huge number of scenarios which

must be examined : for an average seismic event, the automatic picker can find 4 arrival-times for 20 stations. Each arrival-time matches with a phase which can take 5 values. The number of scenarios to be examined is thus $5^{4\times 20} \approx 8.3 \times 10^{55}$. However, constraint programming helps us to fastly generate geophysically coherent scenarios. We have split the problem into several models which work on different variable scales.

We describe in this article a subsumption-like global constraint and argue the choices of constraints which allow to browse such search-spaces in 25 seconds in the worst encountered case.

1 Introduction

1.1 Contexte



(a) Processus de localisation actuel (b) Notre proposition

FIG. 1 – Comparaison du processus de localisation automatique et de notre proposition.

Le DASE (Département d'analyse et de surveillance de l'environnement) du CEA (Commissariat à l'Energie Atomique) est en charge des alertes sismiques en France. Dans le cadre de leur mission, les agents du DASE procèdent à une automatisation de certaines tâches qu'il est nécessaire d'accomplir avant de pouvoir localiser un séisme. Afin de rester au niveau de l'état-de-l'art et d'améliorer le système automatique en termes de précision et de rapidité, le DASE procède régulièrement à une mise à jour de certaines parties du système.

Dans [5], les auteurs proposent une première amélioration de ce système de localisation automatique avec un nouveau processus en trois phases, représenté par la figure 1(a). Dans un premier temps, le système détermine les pointés, c'est-à-dire l'horodatage des phases.

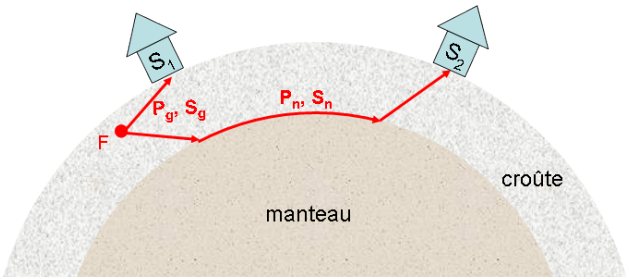


FIG. 2 – Comparaison des ondes de type n et de type g. S_1 et S_2 représentent deux stations et F le foyer du séisme.

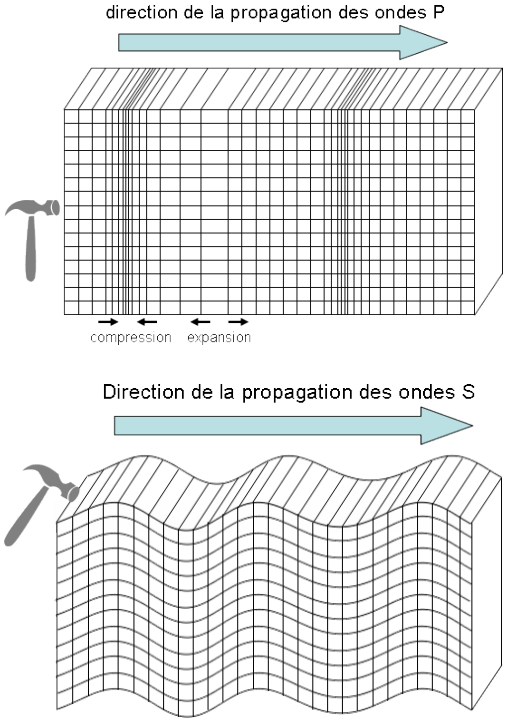


FIG. 3 – Comparaison de la propagation des ondes P et S.

En France, plus d'une quarantaine de stations ont pour but de collecter ces différentes phases. Un pointeur automatique est utilisé pour déterminer à partir de ces signaux les différents pointés qui caractérisent un séisme. Des classifieurs automatiques utilisent la portion de signal autour de ces pointés afin de leur attribuer un type de phase.

Le système proposé distingue cinq types de phases : Pg, Pn, Sg, Sn. Une classe *rejet* est aussi considérée afin de prendre en compte les erreurs éventuelles du pointeur automatique. Les quatre premières classes sont définies en fonction du mode de propagation de l'onde sismique (P ou S) et du chemin emprunté par celle-ci (n ou g), comme illustré dans la figure 3. Les

phases g sont des ondes directes tandis que les phases n sont réfléchies par l'interface croûte-manteau (figure 2). Les classifieurs fournissent ainsi un score pour chacune des classes envisagées. La décision finale consiste en un vote majoritaire pour chacune des phases. L'étiquetage des pointés d'une station est appelé un scénario *station* et l'étiquetage des pointés de l'ensemble du réseau de stations est appelé scénario *réseau*. Finalement, le scénario *réseau* ainsi obtenu permet de procéder à la localisation de l'événement. Il apparaît nettement que la précision de la localisation dépend fortement de l'exactitude de l'étiquetage des phases et de la précision du pointeur automatique, c'est-à-dire de l'exactitude des scénarios *station* et des scénarios *réseau*.

Ce système de localisation automatique peut mener à des scénarios *station* ou des scénarios *réseau* géophysiquement impossibles car aucune propriété géophysique n'est utilisée pendant la prise décision. De plus, le système souffre du fait que les phases sont étiquetées les unes indépendamment des autres. Nous proposons dans cet article une solution à ce problème.

1.2 Position du problème

Le but de notre travail est de sélectionner le scénario *réseau* le plus vraisemblable parmi l'ensemble de scénarios *réseau* possibles. Pour cela, nous souhaitons estimer la probabilité des différents scénarios en considérant les sorties des classifieurs comme des probabilités. Si l'on suppose que toutes les phases sont indépendantes les unes des autres, la probabilité d'un scénario *station* peut être estimée par le produit des probabilités des phases. Si l'on suppose aussi que toutes les stations sont indépendantes, alors la probabilité d'un scénario *réseau* peut être estimée par le produit des probabilités des scénarios *station* qui le composent. Etant données ces hypothèses, la sélection du scénario *réseau* le plus vraisemblable et géophysiquement cohérent peut être effectué en calculant les probabilités de tous les scénarios *réseau* corrects (figure 1(b)). Contrairement au processus d'étiquetage précédent, l'étiquetage d'une phase se fait à présent respectueusement de toutes les autres phases : la cohérence de l'étiquetage est dans un premier temps vérifié au niveau des stations, puis au niveau du réseau global. Dans la mesure où nous nous insérons dans un système opérationnel existant, le temps de calcul doit être une préoccupation importante et ne doit pas excéder la demi-minute.

Afin de résoudre le problème de génération des scénarios *réseau* géophysiquement possibles, nous avons choisi d'adopter la programmation par contraintes [7]. Dans un autre cadre, des réseaux de contraintes sont souvent utilisés pour reconnaître des scénarios décrits par des contraintes temporelles, comme celles proposées par Allen [1]. Par exemple, dans le domaine du multimédia, les solveurs permettent de reconnaître le genre d'un document audiovisuel [6, 2] en utilisant des contraintes pour décrire un modèle pour chaque genre considéré. Dans ces cas là, il s'agit de tester une solution potentielle pour déterminer si elle satisfait l'ensemble des contraintes, ce qui se résout en un temps polynômial.

Dans notre cas, la tâche est un peu différente : nous avons besoin d'obtenir l'ensemble des scénarios *réseau* géophysiquement possibles parmi l'ensemble des scénarios pouvant être élaborés à partir des différents pointés. Prenons le cas par exemple d'un séisme moyen : les différentes ondes issues de ce séisme sont collectées par une vingtaine de stations. En moyenne, le pointeur automatique trouve 4 pointés par station. Chacun de ces pointés peut prendre une valeur parmi les cinq considérées : Pg, Pn, Sg, Sn ou *rejet*. Il y a donc $5^4 = 625$ scénarios *station* possibles par station. Puisque une vingtaine de stations a perçu les ondes du séisme, le nombre de scénarios *réseau* est de l'ordre de $625^{20} \approx 8.3 \times 10^{55}$. Cette estimation est très pessimiste dans la mesure où les contraintes sur les scénarios *station* sont très fortes. Evidemment, la difficulté de notre travail va consister à parcourir cet espace de recherche très grand en quelques secondes. Les contraintes géophysiques, obtenues auprès des experts, ne suffisent pas à elles seules à satisfaire cette contrainte opérationnelle.

Dans la suite de cet article nous décrivons notre système composé de plusieurs modèles. Nous décrivons ensuite les contraintes utilisées et nous introduisons une nouvelle contrainte globale, comparable à la subsumption logique, que nous avons mise en place afin d'accélérer le parcours de l'espace de recherche. Nous terminerons par l'exhibition de quelques résultats expérimentaux avant de conclure.

2 Description du système

Nous avons choisi d'utiliser plusieurs modèles afin de générer la liste des scénarios *réseau* géophysiquement possibles. Ce choix a été appuyé par le fait que les différents modèles allaient utiliser des variables de types très différents. La figure 4 donne un aperçu du système.

Le premier modèle est en charge de générer les scénarios *station* possibles. Les variables à ce niveau représentent les pointés perçus par une même station par un tuple (t, l) où t est la date d'arrivée et l une étiquette parmi Pg, Pn, Sg, Sn et *rejet*. En effet, puisque les scénarios *station* ont au plus quatre phases alors que le pointeur automatique peut en trouver jusqu'à

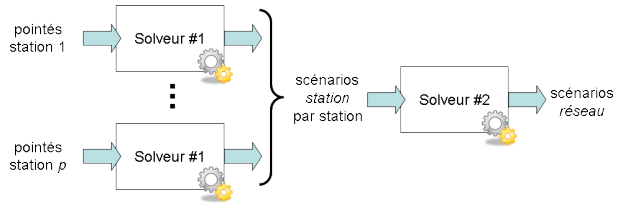


FIG. 4 – Architecture du système.

huit, la classe *rejet* nous permettra de prendre en compte des pointés issus des erreurs du pointeur automatique. Puisque t est connu pour chaque pointé, le rôle du premier modèle est d'attribuer une étiquette à chacun d'eux. Ainsi, les solutions obtenues après résolution du modèle représentent des scénarios *station* dont la cohérence est vérifiée à l'échelle de la station. Dans un soucis de performance, plusieurs instances du premier solveur sont exécutées en parallèle (une par station).

Le second modèle est composé de variables qui représentent un scnéario *station* : il y a donc autant de variables que de station ayant perçue l'événement. Ce modèle a donc pour but d'unifier les scénarios *station* en un unique scénario *réseau*. Son rôle est donc de s'assurer que l'étiquetage des phases est cohérent au niveau du réseau global. La particularité du second modèle est que les domaines des variables sont les solutions du premier modèle.

Pour chacun de ces deux niveaux de cohérences, des contraintes différentes seront utilisées. Nous décrivons ces contraintes dans la section suivante.

3 Contraintes géophysiques et parcours de l'espace de recherche

Nous avons implémenté notre propre solveur de contraintes basé sur un algorithme de *backtracking* avec filtrage [4] afin de parcourir l'espace de recherche : il s'agit d'un solveur dédié uniquement à ce problème. Si le premier modèle utilise majoritairement des contraintes globales, nous avons préféré limiter le second à des contraintes binaires ou unaires : en effet, les contraintes portent ainsi sur au plus deux stations et les instanciations partielles peuvent être plus rapidement évincées. En particulier, à chaque fois qu'une station sera ajoutée à l'instanciation partielle d'un scénario *réseau*, il suffira de la comparer à chacune des stations déjà instanciées.

Les contraintes que nous avons utilisées découlent de la principale information géophysique que nous avons, utilisé par les experts eux-mêmes : l'hodochrone (figure 5). L'hodochrone est une représentation graphique qui fournit le temps de propagation des phases en fonction

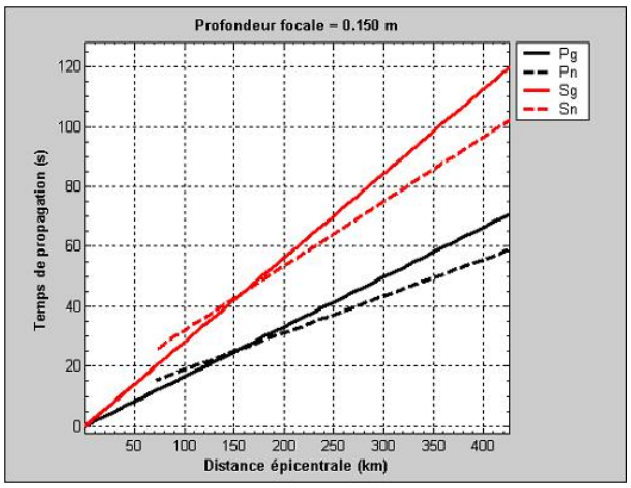


FIG. 5 – Hodochrone de référence.

de la distance entre l'événement et la station. L'hodochrone que nous utilisons n'est qu'une simplification des hodochrones réels : il s'agit d'un hodochrone de référence qui a été estimé empiriquement sur l'ensemble des séismes d'une année en France. Afin de servir de modèle, il a été simplifié. Nous supposons que les autres paramètres d'un hodochrone réel, comme par exemple la profondeur du foyer sismique, ne modifient que négligemment l'hodochrone de référence.

Dans le reste de cet article, nous appellerons une phase un pointé qui ne porte pas l'étiquette *rejet*. Nos contraintes proviennent en fait d'une comparaison entre des phases : c'est pour cela qu'elles ne sont applicables que pour les scénarios *station* à au moins deux phases. De même, les contraintes portant sur les pointés ne fonctionnent qu'à partir de deux pointés par station.

Contrainte 1 (exclusivité) Deux pointés ne peuvent avoir la même étiquette excepté dans le cas où l'étiquette est celle de la classe rejet. Il s'agit de la contrainte globale *alldifferent_except_0* (”*tous_différents_sauf_0*”) où 0 représente notre étiquette *rejet*.

Contrainte 2 (ordre des phases) Les ondes P arrivent avant les ondes S. Il s'agit cette fois d'une contrainte globale spécifique à notre problème. Si p représente le nombre de pointés perçus par la station considérée et si (t_i, l_i) représente le i^{e}pointé, alors la contrainte peut s'écrire :

$$\forall i, j \in \{1, \ldots, p\} \text{ tels que } t_i < t_j,$$
$$\nexists\, (l_i, l_j) \text{ tel que } l_i \in \{Sg, Sn\} \text{ et } l_j \in \{Pg, Pn\}. \quad (1)$$

Contrainte 3 (temps maximal) La différence entre les dates d'arrivées de deux pointés étiquetés ne

doit pas dépasser un certain seuil $\Delta_{max}(l_1, l_2)$ où l_1 et l_2 sont des étiquettes différentes de *rejet* :

$$\forall i, j \in \{1, ..., p\} \text{ tels que } t_i < t_j \text{ et } l_i, l_j \notin \{rejet\} \\ |t_i - t_j| < \Delta_{max}(l_i, l_j). \quad (2)$$

Les deux premières contraintes sont fortement restrictives et réduisent considérablement l'espace de recherche du premier modèle. Sans ces contraintes, nous avions p^5 scénarios *station* possibles, où p est le nombre de pointés. A présent, avec seulement les deux premières contraintes, le nombre n de scénarios est donné par la relation :

$$n = 1 + 4 \times p + 8 \times C_p^2 + 8 \times C_p^3 + 4 \times C_p^4 \quad (3)$$

puisqu'il n'y a que 8 scénarios à 2 ou 3 phases et 4 scénarios à 4 phases possibles, complétés ensuite par des pointés rejetés. Ainsi, avec 4 phases, nous passons de 625 à 117 scénarios *station* possibles.

Le second solveur doit comparer deux scénarios *station* afin de déterminer s'ils sont géophysiquement compatibles. Tout scénario *station* est géophysiquement compatible avec un scénario *station* à une seule phase ou avec un scénario *station* qui n'a que des pointés étiquetés en *rejet* (appelé « scénario vide »). Ce n'est qu'à partir du second solveur que nous exploitons des informations avancées telles que le temps t_0 de l'événement et la distance δ entre la station et l'événement à partir de deux phases. Soit (a_{l_i}, b_{l_i}) les paramètres de la droite de l'hodochrone correspondant au type de phase l_i. Nous considérons deux phases (t_1, l_1) et (t_2, l_2), telles que $t_1 < t_2$ qui ne sont pas nécessairement consécutives. Alors t_0 et δ peuvent s'exprimer comme suit :

$$t_0 = \frac{a_{l_1} \times (t_1 + b_{l_2}) - a_{l_2} \times (t_2 - b_{l_1})}{a_{l_1} - a_{l_2}} \quad (4)$$

$$\delta = \frac{t_1 - t_2 - b_{l_1} + b_{l_2}}{a_{l_1} - a_{l_2}} \quad (5)$$

Par la nature de l'hodochrone et une estimation rapide de l'erreur, au plus les phases choisies seront éloignées, au plus t_0 et δ seront précis. Les contraintes du second modèle sont soit binaires soit unaires et prennent des scénarios *station* en argument :

Contrainte 4 (dates de l'événement similaires) La différence entre les dates de l'événement estimées de façon indépendante pour chacune des stations doit être inférieure à un seuil Δ_{t_0}. Soit t_0^1 and t_0^2 l'estimation du t_0 à partir du premier et du second scénario *station* :

$$\left|t_0^1 - t_0^2\right| < \Delta_{t_0}. \quad (6)$$

Contrainte 5 (antériorité des t_0) Les dates de l'événement estimées de façon indépendante pour chacune des stations doivent être antérieures à la première phase de chacune des deux stations. Si (t_1, l_1) représente la première phase de la station, alors $t_0 < t_1$ doit être vérifié.

Contrainte 6 (croissance de l'hodochrone) Si la première phase de la première station est antérieure à la première phase de la seconde station et qu'elles ont la même étiquette, alors la seconde phase de la première station doit aussi être antérieure à la seconde phase de la seconde station si elles ont la même étiquette. Supposons que le premier scénario *station* S_1 soit constitué de n pointés et que le second scénario *station* S_2 soit constitué de m pointés, alors :

$$\forall i, j \in \{1, ..., n\} \text{ et } \forall i\prime, j\prime \in \{1, ..., m\} \\ \text{tels que } t_i < t_j, t_{i\prime} < t_{j\prime}, l_i, l_j, l_{i\prime}, l_{j\prime} \notin \{\text{rejet}\}$$

$$(l_i = l_{i\prime}) \wedge (l_j = l_{j\prime}) \wedge (t_i < t_{i\prime}) \Rightarrow (t_j < t_{j\prime}). \quad (7)$$

Contrainte 7 (inégalité triangulaire) Il s'agit d'une règle géométrique contraignant la distance estimée des stations par rapport à l'épicentre de l'événement. Soit deux stations éloignées de l'épicentre des distances δ_1 et δ_2 et éloignées entre elles de la distance δ_{12}. Ces trois distances sont liées par les relations suivantes :

$$|\delta_1 - \delta_2| / \tau \leq \delta_{12} \leq (\delta_1 + \delta_2) \times \tau \quad (8)$$

où τ est un facteur d'assouplissement de la contrainte géométrique.

Contrainte 8 (absence des phases $\mathbf{P_n}$ et $\mathbf{S_n}$) En dessous d'une certaine distance $\delta_{P_nS_n}$ avec l'épicentre, une station ne peut percevoir de phase P_n ou S_n. $\delta_{P_nS_n}$ est appelée "distance critique". Si S représente le scénario *station* courant et si on suppose que nous avons un prédicat *Contient(S)* dont la valeur est vraie si S contient une phase Pn ou Sn, alors la contrainte peut s'écrire :

$$Contient(S) \Rightarrow (\delta > \delta_{PnSn}). \quad (9)$$

Contrainte 9 (respect de l'hodochrone) Les différents pointés étiquetés sont replacés sur l'hodochrone afin de vérifier s'ils sont cohérents entre eux. Les deux pointés les plus éloignés sont utilisés pour déduire la distance entre l'épicentre et la station. A partir de cette distance, l'hodochrone nous permet de prédire le délai entre les autres phases. La différence entre le délai prédit et le délai observé doit être inférieure à un seuil $\Delta_{hodochrone}$. Soit $t_0(i, j)$ et $\delta(i, j)$ la date de

l'événement et sa distance à la station estimés à partir de la i^{e}et de la j^{e}phase, alors :

$$\forall i,j,k \in \{1,..,n\}$$
$$\text{tels que } t_i \neq t_j \neq t_k \text{ et } l_i, l_j, l_k \notin \{rejet\}$$

$$t_k - t_0(i,j) - a_{l_k} \times \delta(i,j) + b_{l_k} \leq \Delta_{hodochrone} (10)$$

Afin de relâcher un peu les contraintes, compte tenu qu'elles ne sont déduites que d'un hodochrone de référence, nous avons introduit différents seuils. Ces seuils ont une signification géophysique et peuvent être paramétrés avec l'aide des experts. Dans le reste de cet article, nous dirons que deux scénarios *station* sont incompatibles si au moins une des contraintes ci-dessus est violée.

Les contraintes unaires du second modèle semblent adéquates au premier modèle. Cependant, par un soucis de temps de calcul, nous ne calculons t_0 et δ qu'une seule fois durant le processus, et les contraintes du second modèle utilisent plus souvent ces valeurs. Notons aussi que les contraintes 4 et 7 sont redondantes (sans être redondantes au sens de la propagation [3]) puisque t_0 et δ sont tirés de la même équation.

Dans le cas de petits événements sismiques, c'est-à-dire avec un nombre de pointés limités, les solutions sont générées en quelques secondes. Malheureusement, dans le cas de séismes moyens ou forts, l'objectif de la demi-minute que nous nous sommes fixés est largement dépassé. Dans les cas les plus complexes, il faut jusqu'à une dizaine de minutes. Cela vient du fait que certains scénarios ne sont filtrés par aucune contrainte du second solveur, comme par exemple les scénarios à une seule phase ou les scénarios vides, ce qui a pour effet d'augmenter considérablement le nombre de combinaisons à tester. Afin d'éviter ce problème, nous avons introduit une contrainte globale qui décuple la vitesse de parcours de l'espace de recherche.

4 Contrainte globale

Afin d'accélérer le parcours de l'espace de recherche, nous avons introduit une contrainte globale proche de la subsumption en logique. La subsumption est une relation entre deux concepts A et B. Si A possède certaines propriétés et que A est plus général que B, alors B possède ces mêmes propriétés. On dit alors que A subsume B. Nous nous proposons d'avoir ce genre de relation entre deux scénarios *station* afin d'éliminer de nombreuses valeurs des domaines des variables du second modèle.

Prenons par exemple deux stations S_1 et S_2 qui perçoivent trois phases chacune. Notons $\times$ les pointés rejetés. Si le scénario $sc_1 = \mathrm{P}_g \times \mathrm{S}_g$ de S_1 n'est pas compatible avec le scénario $sc_2 = \mathrm{P}_g \times \mathrm{S}_n$ de S_2, alors sc_1 est aussi incompatible avec $sc_2\prime = \mathrm{P}_g\ \mathrm{P}_n\ \mathrm{S}_n$ de S_2. Plus généralement, si sc_1 et sc_2 sont incompatibles, alors tout scénario de S_1 construit à partir de sc_1 sera incompatible avec tous les scénarios de S_2 construits à partir de sc_2. La construction d'un scénario à partir d'un autre se fait en remplaçant des pointés rejetés par des phases (tout en respectant les contraintes 1 à 3). En revanche la réciproque n'est pas vraie : si sc_1 est compatible avec sc_2, les scénarios dérivés ne sont pas forcément compatibles entre eux. A cause de cela, nous acceptons de procéder à un filtrage sous-optimal : nous allons considérer que la réciproque est vraie afin de diminuer la complexité du parcours de l'espace de recherche, en acceptant en contrepartie de filtrer moins de scénarios. Notre filtrage aurait été optimal sans la considération de cette hypothèse de travail.

Dans l'exemple précédent, les scénarios sc_1 et sc_2 sont appelés "scénarios générateurs" tandis que le scénario $sc_2\prime$ est appelé "scénario subsumé". Un scénario générateur et l'ensemble de ses subsumés sont appelés "une famille". Le tableau 1 montre deux exemples de familles de scénarios *station*.

générateurs	$\times$	$\times$	P_g	$\times$	$\times$	S_n
subsumés	P_g	$\times$	P_g	P_n	$\times$	S_n
	P_n	$\times$	P_g	P_n	S_g	S_n
	S_g	$\times$	P_g	S_g	$\times$	S_n
	S_n	$\times$	P_g	$\times$	P_n	S_n
	$\times$	P_g	P_g	$\times$	S_g	S_n
	$\times$	P_n				
	$\times$	S_g				
	$\times$	S_n				

TAB. 1 – Exemples de scénarios *station* et de leurs subsumés. Le symbole $\times$ représente la classe rejet.

Ainsi, nous considérons que le scénario *station* vide subsume les scénarios *station* à une phase et que les scénarios *station* à deux phases subsument les scénarios à trois ou quatre phases. En effet, les cas des scénarios *station* vides et ceux à deux phases doivent être séparés dans la mesure où les contraintes 5 à 9 découlent principalement de l'hodochrone que nous ne pouvons exploiter qu'à partir de deux phases.

La subsumption diminue ainsi la complexité du problème. En effet, si le pointeur automatique détecte 6 pointés pour la station S_1 , 365 scénarios *station* sont à envisager dont 120 à deux phases. De même, si le pointeur automatique détecte 8 pointés pour la station S_2 , 985 scénarios *station* sont à envisager dont 224 à deux phases. Plaçons nous dans le pire des cas, c'est-à-dire celui dans lequel le premier solveur n'a filtré aucun scénario. Pour construire l'ensemble des scénarios *réseau* possibles compte tenu des stations S_1 et S_2, il faudra tester chacun des scénarios de S_1 avec chacun des scénarios de S_2. Sans la subsumption, cela revient à effectuer $365 \times 985 = 359525$ comparaisons,

alors qu'avec la subsumption, ce nombre de comparaisons est réduit à $121 \times 225 = 27225$, soit plus de 13 fois moins. En effet, 121 et 225 sont le nombre de scénarios *station* générateurs de S_1 et S_2, c'est-à-dire le nombre de scénarios *station* à deux phases auxquels s'ajoute le scénario *station* vide. En considérant un nombre de stations plus grand, l'apport de la subsumption est pleinement révélé.

L'introduction de cette contrainte globale a un impact sur l'architecture du système (figure 6). D'un côté, puisque seuls les scénarios *station* vides ou à deux phases sont nécessaires pour générer les scénarios *réseau*, nous avons bridé le premier solveur afin qu'il ne fournisse que ce genre de solutions, à savoir des scénarios générateurs. La production de ces scénarios est ainsi plus rapide à tel point que la parallélisation n'est plus nécessaire. D'un autre côté, le second solveur peut rester inchangé : il combinera les scénarios *station* générateurs entre eux pour obtenir les scénarios *réseau*. Cependant, en ne considérant que les scénarios générateurs, le second solveur ne fournit qu'un sous-ensemble des solutions possibles que nous appelons « familles de scénarios *station* ». Nous complétons donc ces familles en calculant les subsumés de chacun des scénarios générateurs à l'aide d'un troisième solveur.

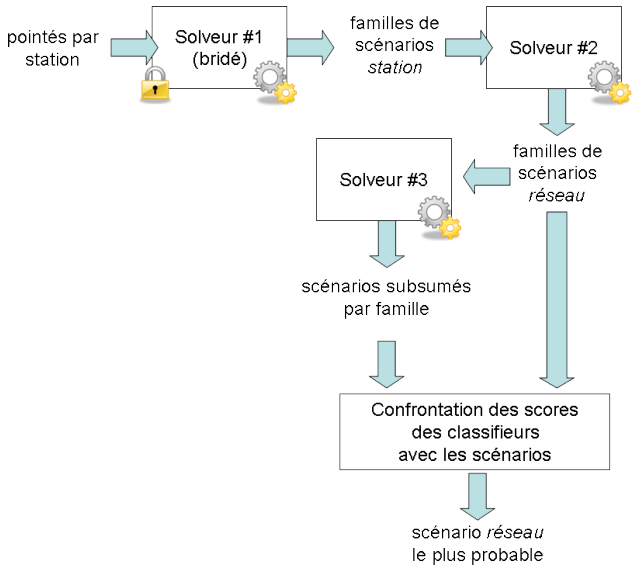


FIG. 6 – Nouvelle architecture du système.

Le troisième solveur doit uniquement construire les subsumés à partir des scénarios générateurs impliqués dans au moins une famille de scénarios *réseau*. Par souci d'efficacité de la contrainte globale, les contraintes utilisées par le troisième solveur vérifient juste la compatibilité des scénarios subsumés avec le scénario générateur. Le tableau 2 montre la nouvelle répartition des contraintes dans les trois solveurs.

Solveur	Rôle	Contraintes
#1	Calcul bridé des scénarios *station*	1, 2, 3
#2	Calcul des familles de scénarios *réseau*	4, 5, 6, 7, 8
#3	Calcul des scénarios subsumés *station*	1, 2, 3, 9

TAB. 2 – Répartition des contraintes

Station 1	Station 2	Station 3
Pg × Pn ×	Pg × Sn	× ×
Pg × Pn Sg ⋆	Pg Pn Sn ⋆	Pg ×
Pg × Pn Sn	Pg Sg Sn	Pn ×
Pn × × Sn	Pg × Sg ⋆	Sg ×
Pn Pg × Sn ⋆	Pg Pn Sg	Sn × ⋆
Pn × Pg Sn		× Pg
Pn × Sg Sn		× Pn
		× Sg
		× Sn

TAB. 3 – Familles de scénarios *station* pour trois stations. Les cellules grisées représentent les scénarios générateurs de chaque famille et les ceux marqués par ⋆ sont les plus probables de leur famille.

Cet arrangement en familles va rendre plus efficace le calcul des probabilités pour trouver le scénario *réseau* le plus probable : il s'agit d'une sorte de factorisation des scénarios *réseau*. En effet, pour chacune des familles de scénarios *station*, nous allons trouver le scénario le plus probable. Ensuite, pour chaque famille de scénarios *réseau*, nous allons multiplier entre elles les probabilités des ces scénarios *station* les plus probables. Ainsi, la redondance dans les calculs est éliminée. Prenons par exemple trois stations dont les familles de scénarios *station* sont répertoriées dans le tableau 3 : les scénarios les plus probables de chaque famille sont marqués du caractère ⋆ et les scénarios grisés sont les scénarios générateurs de chaque famille. Dans cet exemple, on suppose que tous les scénarios sont compatibles entre eux.

Sans l'agrégation en familles, il faudrait combiner les scénarios *station* entre eux et former ainsi 315 scénarios *réseau*, puis trouver le plus probable d'entre eux. A présent, avec le regroupement en familles, il suffit de trouver pour chaque famille de scénarios *réseau* le scénario le plus probable en prenant pour chaque famille de scéarios *station* qui la composent le scénario *station* le plus probable. Cela mène au final à la comparaison de seulement 4 scénarios *station* ce qui est plus rapide.

5 Expérimentations

Tous les paramètres de notre système (hodochrone, seuils) ont été fixés à l'aide des experts : en effet, ces paramètres ont un sens géophysiques et sont souvent interdépendants. Nous avons tenté de caractériser les performances de notre système, ce qui n'est pas facile compte tenu que les effets des contraintes du second modèle sont difficilement quantifiables.

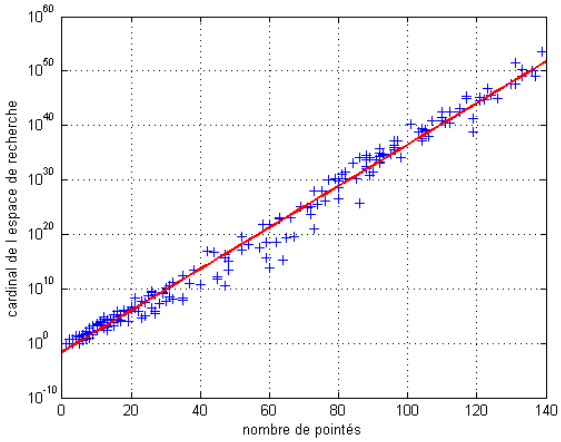


FIG. 7 – Estimation de la taille de l'espace de recherche en fonction du nombre de pointés (échelle log-log)

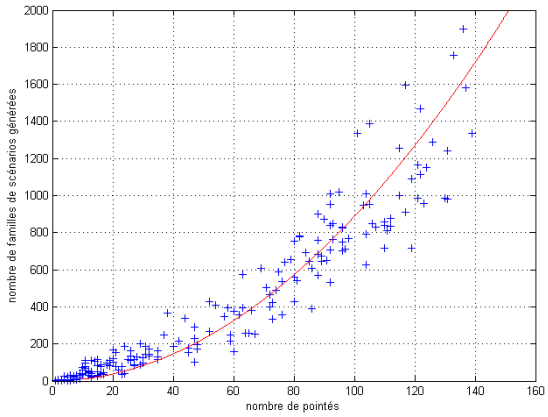


FIG. 8 – Estimation du nombre de scénarios *réseau* en fonction du nombre de pointés (échelle linéaire)

Le premier et le troisème modèle génèrent les solutions de façon presque instantanée. La génération des scénarios à 2 phases ne nécessite plus d'être vu comme un problème de satisfaction de contraintes tant la combinatoire est faible. En revanche, nous nous sommes intéressés au second modèle qui est confronté à un espace de recherche plus important. Il est possible d'estimer grossièrement la taille de son espace de recherche en multipliant les cardinaux des domaines des variables impliquées, c'est-à-dire en multipliant le nombre de scénarions *station* possibles pour chaque station entre eux. La figure 7 montre la taille de cet espace de recherche en fonction du nombre total de pointés. Après une simple régression, nous obtenons la relation suivante :

$$C \quad \approx 0.02 \times e^{0.88 \times p} \tag{11}$$

où C est la taille de l'espace de recherche et p le nombre de pointés. Cette relation montre que le cardinal de l'espace de recherche croît de façon exponentielle par rapport au nombre de pointé, même si la croissance est limitée par des coefficients assez faibles.

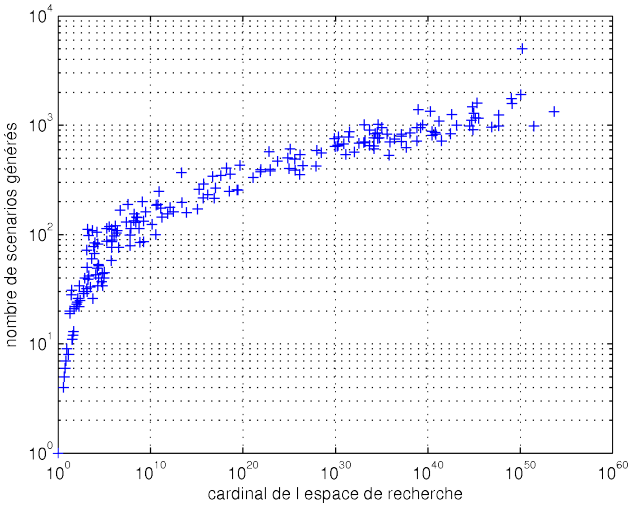


FIG. 9 – Nombre de familles de scénarios *réseau* générées en fonction du cardinal de l'espace de recherche. Echelle log-log.

Puisque le second modèle doit générer des familles de scénarios *station*, nous nous sommes intéressés au lien entre le cardinal de l'espace de recherche et le nombre familles générées. Il est important que ce nombre reste relativement faible afin de ne pas augmenter le coût des post-traitements comme la recherche du scénario le plus vraisemblable. La figure 9 montre que l'évolution suit une fonction croissante concave. Ainsi un espace de recherche plus grand ne va pas apporter nécessairement beaucoup plus de familles de scénarios *réseau* par rapport à un espace de recherche plus petit.

Enfin, nous avons tenté de caractériser le lien entre le temps de calcul total (comprenant les trois modèles) et le nombre de pointés. Cette relation est montrée dans la figure 10. Par régression, la relation s'écrit :

$$T \quad \approx 2.56 \times 10^{-7} \times p^{3.75} \tag{12}$$

où p est le nombre de pointés et T le temps de calcul en secondes. Puisqu'une alerte sismique doit être donnée sous trois minutes, nous pouvons déduire de cette dernière équation que le système pourra traiter le cas d'un séisme ayant généré 229 arrival times.

Ainsi, l'événement de 2005 qui a le plus grand espace de recherche a activé 42 stations et le pointeur auto-

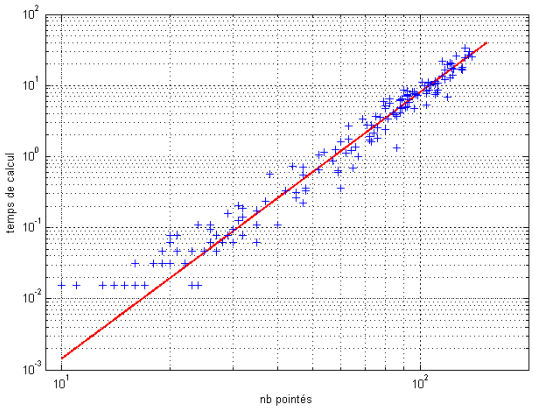


FIG. 10 – Temps de calcul (en secondes) en fonction du nombre de pointés (échelle log-log)

matique a généré 139 pointés. Pour cet événement, 1334 familles de scénarios *station* ont été générées en moins de 25 secondes. Ces familles, une fois dévéloppées, représentent plus de 1.6×10^{49} scénarios *réseau*. Ceci révèle pleinement l'avantage de la représentation en famille et la faiblesse du filtrage produit par les contraintes.

6 Conclusion

Nous avons présenté dans cet article un cas concret de l'utilisation des problèmes de satisfaction de contraintes dans le cas où l'ensemble des solutions est nécessaire. Nous avons présenté un système dont l'architecture originale s'intègre dans une chaîne de traitement existante. Le but est d'automatiser la procédure de localisation des événements sismiques et notre première tâche a été de déceler une des faiblesses du système proposé dans [5].

Notre amélioration consiste à générer tous les scénarios géophysiques possibles et d'exhiber le plus vraisemblable. La génération des scénarios se fait à l'aide d'un enchaînement de solveurs qui utilisent des contraintes fournies par les experts du DASE. Afin d'accélérer la production des scénarios, nous avons introduit une contrainte globale qui s'avère fortement efficace en diminuant l'espace de recherche durant le parcours de celui-ci. En contrepartie, pour mettre en place cette contrainte globale, nous avons accepté d'avoir un filtrage sous-optimal qui produira quelques scénarios en trop.

En pratique, le nombre de scénarios générés est parfaitement raisonable et notre système les calcule en quelques secondes. Dans le pire des cas étudiés, 25 secondes sont nécessaires pour produire 1334 familles de scénarios *réseau*.

Références

[1] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11) :832–843, 1983.

[2] J. Carrive, F. Pachet, and R. Ronfard. Clavis : a temporal reasoning system for classification of audiovisual sequences. In *Proceedings of Content-Based Multimedia Information Access (RIAO) Conference*, Paris, France, 2000.

[3] C. W. Choi, J. H.M. Lee, and P. J. Stuckey. Removing propagation redundant constraints in redundant modeling. *ACM Trans. Comput. Logic*, 8(4) :23, 2007.

[4] Vipin Kumar. Algorithms for constraint satisfaction problems : A survey. *AI Magazine*, 13 :32–44, 1992.

[5] A. Larue, D. Mercier, L. Cornez, F. Suard, J. Guilbert, and C. Maillard. Automatic classification of seismic phases for event location procedure. In *Proceedings of the 31st General Assembly of the European Seismological Commission*, 2008.

[6] A. Liret, N. Ramaux, N. Fontaine, M. Dojat, and F. Pachet. N-ary constraints for scenario recognition. Workshop on n-ary constraints, ECAI-98, 1998.

[7] Pascal Van Hentenryck. *Principles and Practice of Constraint Programming*. MIT Press, Cambridge, MA, USA, 1995.