# Conception d'une place de marché pour la vente et la distribution d'énergie dans les *smart grids*

Jesús Cerquides[†] cerquide@iiia.csic.es
Gauthier Picard[⋆] gauthier.picard@emse.fr
Juan A. Rodríguez-Aguilar[†] jar@iiia.csic.es

[†]IIIA-CSIC, Campus UAB, 08193 Cerdanyola, Catalonia, Spain

[⋆]Laboratoire Hubert Curien UMR CNRS 5516 + Institut Henri Fayol, MINES Saint-Etienne, France

***Résumé***

*Cet article introduit un nouveau mécanisme de marché qui permet aux prosommateurs d'échanger de l'électricité tout en satisfaisant les contraintes physiques du réseau. La règle d'allocation de notre marché est mise en œuvre au moyen de* RadPro*, un algorithme efficace de programmation dynamique qui évalue en temps polynomial combien d'énergie chaque prosommateur échange ainsi que comment l'énergie doit être distribuée au travers du réseau. Nos résultats empiriques montrent que* RadPro *surclasse de manière significative CPLEX et Gurobi en temps lors du calcul de l'allocation optimale dans des réseaux acycliques. De plus, la gestion par envoi de messages de* RadPro *offre la possibilité d'exécuter notre marché d'une manière décentralisée (pair-à-pair).*

***Mots-clés :*** *smart grid, marché de l'énergie, prosommateurs,* RadPro

***Abstract***

*This paper introduces a novel market that allows prosumers to trade electricity while satisfying the constraints of the grid. Our market's allocation rule is implemented by means of the so-called* RadPro*, an efficient dynamic programming algorithm that assesses in polynomial time how much energy each prosumer trades as well as how energy must be distributed throughout the grid. Our empirical results show that* RadPro *significantly outperforms both CPLEX and Gurobi in solving time when computing the optimal allocation over acyclic networks. Furthermore, the message-passing nature of* RadPro *offers the possibility of running our market in a decentralized (peer-to-peer) manner.*

***Keywords:*** *smart grid, energy market, prosumers,* RadPro

## 1 Introduction

Notre modèle centralisé de production et de transmission de l'énergie est une source d'énormes gâchis [6]. Comme les stations de production sont généralement éloignées des centres de demande, la plupart de la chaleur produite est non utilisée, mais ventilée dans des cheminées ou rejetée dans des rivières. Des pertes additionnelles surviennent lors du transport de l'électricité le long des câbles des systèmes de transmission et de distribution [6, 23]. Favoriser une génération décentralisée de l'énergie devrait réduire les inefficacités de génération et de distribution et faciliter des contributions accrues des énergies renouvelables [23]. Ce nouveau modèle est destiné à être pris en charge par les *smart grids* (ou *SG*).

Une SG est un réseau d'électricité qui peut intégrer les actions de tous les utilisateurs connectés —générateurs, consommateurs— afin de livrer efficacement (en terme d'économie, de sécurité) de l'électricité. Dans une SG, le consommateur peut être un individu ou un foyer, mais aussi une communauté ou une PME. Dans sa forme la plus générale, une SG est peuplée de *prosommateurs* capables à la fois de production et de consommation d'énergie. Ainsi, les SG jouent un rôle central dans l'intégration de tous ces usagers au moyen de l'adoption d'un système qui satisfait un certain nombre d'objectifs sociétaux. Parmi ces objectifs, il y a celui de la fixation des prix du marché de l'électricité en tenant compte des contraintes structurelles de la SG. Un défi majeur de la recherche au cœur de plusieurs feuilles de route pour les SG [3, 4] est donc la conception de marchés pour prosommateurs qui prennent en compte des contraintes du réseau de distribution, ce qui permettra aux prosommateurs de commercer directement sur la SG [8]. D'après [17], les opérations de marché impliqueront un grand nombre de prosommateurs hétérogènes, distribués à travers le réseau (plus près du point d'utilisation de l'électricité), et un commerce d'énergie négociée en quantité plus faible qu'aujourd'hui. La distribution de l'électricité emploie l'un des trois types de topologies de réseaux : radial, en anneau, et interconnecté [5, 7, 21]. Les réseaux radiaux sont acycliques, mais comme on l'observe dans [11], bien que les anneaux et les réseaux interconnectés contiennent des cycles, ils sont confi-

gurés comme des réseaux acycliques au moyen de commutateurs [7, 21].

La vision des SG a stimulé de nombreuses recherches sur la conception de marchés et d'agents commerciaux. L'état de l'art a surtout considéré l'emploi de différents types de ventes aux enchères. Ainsi, l'échange d'énergie basé sur le marché est généralement traité dans la littérature par un ensemble de prosommateurs qui participent à une double vente aux enchères où l'énergie est négociée sur une base journalière [8, 9, 10, 12, 14, 19]. Les exceptions à cette approche commune envisagent des ventes aux enchères multi-unitaires adaptées [22] et des enchères inversées combinatoires simultanées [15] pour faire correspondre la demande à l'offre. Dans la limite de nos connaissances, aucun des mécanismes de marché utilisés dans la littérature à ce jour ne tient compte des contraintes physique de la SG. Ainsi, la compensation du marché se produit sans prendre en compte, par exemple, le fait que la transmission d'énergie est réalisée le long des réseaux de distribution aux capacités limitées [21]. Par conséquent, le commerce et la distribution sont considérés comme des activités découplées.

L'objectif de cet article est donc de concevoir un nouveau marché qui permet aux prosommateurs de faire le commerce de l'électricité dans une SG tout en satisfaisant les contraintes de distribution de la SG. Plus précisément, (i) nous formalisons le problème d'allocation d'énergie (EAP) comme le problème de la décision de la quantité d'énergie échangée entre chaque prosommateur de telle sorte que le bénéfice global soit maximisé tout en respectant les contraintes physiques et les préférences des utilisateurs. Résoudre l'EAP revient à compenser (ou équilibrer) notre marché de prosommation. (ii) Nous proposons un nouvel algorithme de programmation dynamique, par envoi de messages, pour résoudre des instances acycliques d'EAP, appelé Radial Energy Network Algorithm for Prosumer Market (RadPro).

Cet article est structurée comme suit. La section 2 présente l'algorithme Acyclic-Solving sur lequel est basé RadPro. La section 3 définit formellement la règle d'allocation que nous proposons pour équilibrer les marchés d'énergie pour prosommateurs. Ensuite, la section 4 introduit RadPro. La section 5 analyse empiriquement RadPro et la section 6 conclut et trace les perspectives pour nos travaux à venir.

## 2 Fondements algorithmiques

Cette section introduit la notion de problèmes d'optimisation sous contraintes puis présente Acyclic-Solving, l'algorithme de programmation dynamique à la base de RadPro.

Soit $X = \langle x_1, \ldots, x_n \rangle$ une séquence de variables, avec chaque variable $x_j$ prenant ses valeurs dans un ensemble fini $\mathcal{D}_j$, son *domaine*. Le domaine joint $\mathcal{D}_X$ est le produit cartésien des domaines de toutes les variables. Nous notons $\mathbf{x}_j$ un état possible de $x_j$, c'est-à-dire $\mathbf{x}_j \in \mathcal{D}_j$. Etant donné un ensemble de variables $Y \subseteq X$, un tuple $\mathbf{X}_Y$ affecte un état possible à chaque variable de $Y$, c'est-à-dire $\mathbf{X}_Y \in \mathcal{D}_Y$. Une fonction d'utilité $f$ de portée $Y$ est une fonction $f : \mathcal{D}_Y \to \mathbb{R}$. Nous notons $s(f)$ la portée de $f$. La somme de deux fonctions d'utilité $f$ et $f'$ de portée $Y$ et $Z$ respectivement est une nouvelle fonction d'utilité $h = f + f'$ de portée $Y \cup Z$, de telle sorte que $h(\mathbf{T}) = f(\mathbf{T}^{\downarrow Y}) + f'(\mathbf{T}^{\downarrow Z})$ où $\mathbf{T}^{\downarrow Y}$ est la projection du tuple $\mathbf{T}$ à la portée $Y$. La projection d'une fonction d'utilité $f$ de portée $Y$ à la portée $Z$, est $f^{\downarrow Z}(\mathbf{Z}) = \max\{f(\mathbf{Y}) \mid \mathbf{Y} \in \mathcal{D}_Y, \mathbf{Y}^{\downarrow Z} = \mathbf{Z}\}$.

Formellement, un COP (*Constraint Optimisation Problem*) est un problème d'optimisation dont l'entrée est un tuple $\langle X, \mathcal{D}, F \rangle$, où $F$ est un ensemble de fonctions d'utilité et dont l'objectif est de trouver le tuple $\mathbf{X}^*$ qui maximise la somme $g = \sum_{f \in F} f$ des fonctions d'utilité, c'est-à-dire de trouver $\mathbf{X}^* = \operatorname{argmax}_{\mathbf{X}} g(\mathbf{X})$. Etant donné un COP, le *réseau de contraintes* est défini comme le graphe ayant un nœud pour chaque contrainte et un arc entre deux nœud si leur portée partage au moins une variable. Lorsque le réseau de contraintes est acyclique, l'algorithme Acyclic-Solving peut résoudre le COP efficacement [2, Chapitre 9, page 248], si toutefois la portée des fonctions d'utilité est limitée. Acyclic-Solving sélectionne un nœud comme étant la racine. Chaque nœud $j$ (non racine) se voit attribué un parent $p_j$ et une fonction d'utilité $f_j$ (représentant sa propre vue du problème). La portée d'un nœud $j$ est la portée de $f_j$. Le séparateur $s_j$ entre $j$ et $p_j$ est l'intersection de leurs portées (l'ensemble des variables partagées par ces deux agents).

Acyclic-Solving, présenté dans l'algorithme 1, s'exécute en deux phases : (1) les coûts sont envoyés des feuilles jusqu'à la racine ; (2) les affectations optimales sont décidées et communiquée de manière descendante. En fait, il détermine la solution optimale en utilisant le principe suivant. Premièrement, chaque feuille commence par envoyer sa propre utilité à son parent. Lorsque qu'un nœud a reçu les messages de tous ses enfants, il les combine avec sa propre utilité pour produire une utilité agrégée de son propre sous-arbre, puis l'envoie à son parent. Une fois que la racine a reçu les messages de tous ses enfants, il évalue l'utilité agrégée du problème entier, puis décide de la meilleure affectation (tuple de coût maximal) pour ses propres va-

**Algorithme 1 :** ACYCLIC-SOLVING (adapté de [2])

```
// chaque nœud j de l'arbre exécute
1 pour chaque enfant k faire recevoir un message μ_{k→j}
2 si j n'est pas racine alors
3     évaluer le message μ_{j→p_j} = (f_j + Σ_k μ_{k→j})^{↓s_j}
4     envoyer le message μ_{j→p_j} à son parent p_j
5     recevoir le message X*_{s_j} de son parent p_j
6 évaluer la meilleure affectation
  X*_j = argmax_{X_j}(f_j + Σ_k μ_{k→j})(X*_{s_j}, X_j)
7 pour chaque enfant k faire envoyer un X*^{↓s_k} à k
```

riables. Enfin, il envoie cette affectation à ses enfants, qui évaluent à leur tour la meilleure affectation et l'envoient dans leur sous-arbre. Après l'exécution de l'algorithme, chaque agent connaît les valeurs optimales pour les variables de sa portée.

Comme ACYCLIC-SOLVING peut être exprimé comme un algorithme par envoi de messages, il peut aisément être appliqué à des COP distribués [1], où il peut être vu comme un cas particulier de DPOP [16] ou d'Action-GDL [18]. La complexité calculatoire de chaque agent dans ACYCLIC-SOLVING est exponentielle selon le nombre de variables dans sa portée —ce qui est fondamentalement dû au calcul des messages de la ligne 3. Habituellement, une technique de *bookkeeping* est utilisée lors de ce processus pour rendre l'évaluation de la meilleure affectation de la ligne 6 peu coûteuse. La complexité en temps de chaque agent $j$ est exponentielle selon le nombre de variables partagées avec son parent (taille du séparateur $s_j$). Comme le nombres de messages est linéaire, la complexité globale est alors exponentielle.

# 3 Problème d'allocation d'énergie

Cette section fournit un modèle mathématique simple pour le marché énergétique dans le réseau de prosommateurs, et la règle d'allocation proposée pour ce marché. Nous illustrons le modèle des prosommateurs et le modèle du réseau énergétique que nous considérons à l'aide d'un exemple de scénario d'échange d'énergie. Ensuite, nous présentons la règle d'affectation pour ce marché comme la solution d'un problème d'optimisation : le problème d'allocation d'énergie (EAP, *Energy Allocation Problem*).

## 3.1 Scénario d'échange d'énergie

La figure 1 montre un exemple d'un scénario d'échange d'énergie impliquant quatre prosommateurs (représentés par des cercles). Chaque arc connectant deux prosommateurs signifie qu'ils sont physiquement connectés. De plus, chaque lien est étiqueté avec sa capacité à transporter de l'énergie. Par exemple, le prosommateur 1 est connecté au prosommateur 2, et leur lien peut transporter jusqu'à 2 unités d'énergie (e.g. $2kW$). Chaque prosommateur peut offrir d'acheter, de vendre ou bien de transmettre de l'énergie. L'offre de chaque prosommateur est représentée par une table à côté de chaque prosommateur, où chaque entrée est une paire (unités,prix). Par convention, une offre de vente est exprimée par un nombre négatif d'unités et un prix négatif, alors qu'une offre d'achat est traduite par un nombre positif d'unité et de prix. Par exemple, le prosommateur 4 offre (entre autre) : d'acheter 2 unités d'énergie et de payer 1.75c€, de vendre 3 unités pour 11c€, et de transmettre de l'énergie gratuitement (0 unités pour 0c€). Dans la figure 1, nous pouvons observer que le prosommateur 1 ne fait que vendre de l'énergie, et le prosommateur 2 ne fait qu'en vendre, alors que les prosommateurs 3 et 4 peuvent soit vendre soit acheter.

Remarquons que les entrées d'une table représentent des offres mutuellement exclusives. Ainsi, par exemple, l'offre du prosommateur 4 indique qu'il peut soit acheter 2 unités, soit vendre 3 unités, mais pas les deux.

## 3.2 Définition du problème

Le problème auquel les prosommateurs font face dans la figure 1 est de décider combien d'énergie échanger et avec qui de telle sorte que le bénéfice global soit maximisé et que les contraintes de capacité du réseau soient respectées. Ceci signifie que : (i) chaque prosommateur doit choisir une seule offre dans sa table (quelle quantité) ; et (ii) chaque paire de prosommateurs connectés par un lien doivent se mettre d'accord sur la quantité à transférer et la direction de ce transfert (avec qui). Dans ce qui suit, nous exprimons ce problème comme un problème d'optimisation.

D'après la figure 1, le réseau connectant un ensemble de prosommateurs $P$ peut être modélisé comme un graphe $(P, E)$, où les nœuds représentent les prosommateurs et les arcs de $E$ représentent les paires de prosommateurs. Un arc $\{i, j\} \in E$ signifie que le prosommateurs $i$ et $j$ sont physiquement connectés pour échanger de l'énergie. Lorsque $i < j$ on dit que $i$ est un voisin entrant (de l'ensemble $in(j)$) de $j$ et que $j$ est un voisin sortant (de l'ensemble $out(j)$) de $i$.

Chaque prosommateur $j$ exprime ses offres d'achat et de vente au moyen d'une *fonction d'offre* $o_j : \mathbb{Z} \to \mathbb{R} \cup \{-\infty\}$. Par exemple, $o_j(3) = 2$ indique que le prosommateur $j$ souhaite acheter 3 unités à 2c€, alors que $o_j(-4) = -2$ si-

1. où chaque agent est affecté à une seule contrainte et le réseau de contraintes est acyclique.

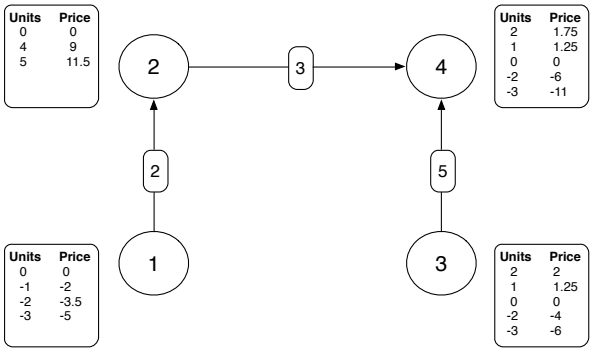



FIGURE 1 – Un scénario d'échange d'énergie : quatres prosommateurs (cercles) publient des offres (tables) dans un réseau aux capacités limitées (arcs).

gnifie qu'il souhaite vendre 4 unités pour 2c€. De plus, nous permettons à un prosommateur d'exprimer qu'il souhaite laisser l'énergie transiter par son nœud en vendant autant d'unités qu'il achète. Nous supposons que $o_j(0) \neq -\infty$ pour tout $j \in P$. Remarquons que les fonctions d'offres capturent les contraintes des prosommateurs. Pour communiquer ses offres, chaque prosommateur envoie une table comme celle de la figure 1 rendant explicite ses états possibles et leurs valeurs. Si une offre pour $k$ unités n'apparaît pas dans la table, ceci signifie qu'un tel état est non admissible pour le prosommateur et qu'ainsi sa valeur $o_j(k)$ est $-\infty$. Formellement, une fonction d'offre est une *valuation*.

**Définition 1.** *Une* valuation $\alpha$ *est une fonction* $\alpha : \mathbb{Z} \rightarrow \mathbb{R} \cup \{-\infty\}$. *Le domaine de* $\alpha$ *est décomposé en deux parties : le* domaine à valeurs finies $FVD(\alpha)$ *est le sous-ensemble de* $\mathbb{Z}$ *dans lequel* $\alpha$ *prend ses valeurs finies. La taille* $n_\alpha$ *d'une valuation est le nombre d'éléments de* $FVD(\alpha)$. *Nous définissons la valuation vide* $\diamond$ *comme étant celle qui associe* 0 *à* 0 *et* $-\infty$ *à tout autre élément.*

Au-delà des offres, nous considérons également que le réseau d'énergie est physiquement contraint par la capacité des connexions entre prosommateurs. Nous noterons $c_{ij}$ la capacité du lien $\{i, j\}$, à savoir le nombre maximum d'unités échangeables entre les prosommateurs $i$ et $j$. Une allocation (ou affectation) spécifie le nombre d'unités d'énergie que chaque prosommateur échange avec ses voisins. Nous formaliserons une allocation par un ensemble de variables $Y = \{y_{ij} \mid i \in P, j \in out(i)\}$, où $\mathbf{y}_{ij}$ représente le nombre d'unité que $i$ vend à $j$ et qui est bornée par la capacité limite $c_{ij}$. Le domaine de la variable $y_{ij}$ est ainsi $D_{ij} = [-c_{ij} .. c_{ij}]$. Par conséquent, si $\mathbf{y}_{ij}$ prend la valeur $k$ strictement positive, ceci signifie que $i$ vend $k$ unités d'énergie à $j$. Sinon, si $\mathbf{y}_{ij}$ prend une valeur strictement négative $-k$, on dit que $i$ achète $k$ unités à $j$. Et donc, $\mathbf{y}_{ij}$ représente un échange du point de vue de $i$.

Maintenant, nous souhaitons évaluer la valeur d'une affectation donnée. Avant cela, nous définissons la valeur locale d'une affectation pour un unique prosommateur. Nous devons évaluer la quantité d'énergie qu'un prosommateur acquiert et vend suivant une affectation $\mathbf{Y}$. Le prosommateur $j$ considérera uniquement sa vue locale de l'affectation, représentée par $\mathbf{Y}_j = \mathbf{y}_{.j} \cup \mathbf{y}_{j.}$. Nous pouvons évaluer l'*équilibre net d'énergie* pour le prosommateur $j$ comme

$$net(\mathbf{Y}_j) = \sum_{i \in in(j)} \mathbf{y}_{ij} - \sum_{k \in out(j)} \mathbf{y}_{jk}, \qquad (1)$$

où les $\mathbf{y}_{ij}$ et $\mathbf{y}_{jk}$ sont agrégées avec différents signes car $j$ prend le rôle de vendeur dans $\mathbf{y}_{ij}$ et d'acheteur dans $\mathbf{y}_{jk}$. Ainsi, la valeur locale $v_j$ de l'affectation $\mathbf{Y}$ pour $j$ peut être évaluée comme la valeur de son équilibre d'énergie net au moyen de sa fonction d'offre

$$v_j(\mathbf{Y}_j) = o_j(net(\mathbf{Y}_j)). \qquad (2)$$

Par conséquent, la valeur d'une affectation $\mathbf{Y}$ peut être obtenue en ajoutant les valeurs locales des affectations des prosommateurs :

$$Value(\mathbf{Y}) = \sum_{i \in P} v_j(\mathbf{Y}_j). \qquad (3)$$

Maintenant, nous sommes prêts à définir le problème allocation d'énergie comme celui qui consiste à trouver la valeur maximale qui satisfait les capacités du réseau.

**Problème 1.** *Etant donnés un ensemble de prosommateurs* $P$, *leurs offres* $\{o_j | j \in P\}$, *et un graphe* $E$ *où chaque arc est étiqueté avec sa capacité* $c_{ij}$, *le problème d'allocation d'énergie (EAP) revient à trouver une affectation* $\mathbf{Y}$ *qui maximise* $Value(\mathbf{Y})$. *Lorsque le graphe* $E$ *est acyclique, nous disons que l'EAP est acyclique.*

Examinons l'exemple de la figure 1. Lors sa résolution, nous obtenons l'affectation de variables présentée dans la figure 2. La solution indique que le prosommateur 1 transfert 2 unités au prosommateur 2 ($y_{12} = 2$), le prosommateur 2 reçoit également 3 unités du prosommateur 4 ($y_{24} = -3$), et le prosommateur 3 transfert 3 unités au prosommateur 4. Dans chaque table d'offres, nous soulignons l'offre choisie par chaque prosommateur. Pour chaque prosommateur, le premier nombre souligné correspond à l'équilibre d'énergie net (équation 1), alors que le prix souligné correspond à la valeur locale de l'affectation (équation 2). Ainsi, l'affectation qui maximise l'équation 3 a la valeur 2.

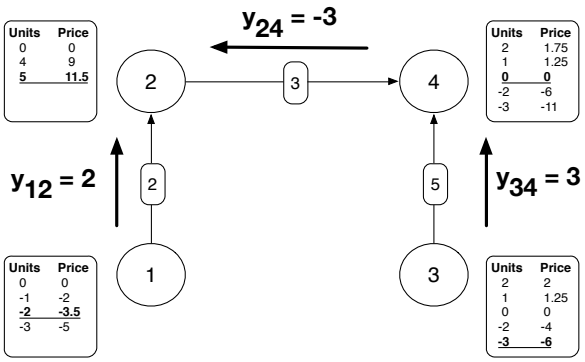



FIGURE 2 – Solution à l'EAP du scénario d'échange d'énergie.

Notons que le prosommateur 2 obtient 5 unités en agrégeant les unités reçues des prosommateurs 1 et 4. Cependant le prosommateur 4 ne vend rien au prosommateur 2. Le rôle du prosommateur 4 est de *relayer* au prosommateur 2 l'énergie transférée du prosommateur 3, qui est celui qui vend l'énergie. En général, notre modèle considère que chaque prosommateur (i) agrège l'énergie reçue de ses voisins lors d'achats ; ou bien (ii) partage et distribue de l'énergie à ses voisins lors de ventes ; ou bien (iii) relaie de l'énergie de telle sorte que les autres prosommateurs puissent satisfaire leur demande.

# 4 Envoi de messages pour EAP acyclique

Cette section présente RADPRO, un nouvel algorithme pour EAP acyclique basé sur des envois de messages, et qui peut ainsi être facilement implanté de manière distribuée. Comme expliqué dans la section 2, l'algorithme ACYCLIC-SOLVING repose sur des envois de messages pour résoudre des COP acycliques. Le but de cette section est double. Premièrement, la section 4.1 détaille comment employer ACYCLIC-SOLVING directement pour résoudre EAP. Deuxièmement, comme l'évaluation des messages dans ACYCLIC-SOLVING est de complexité exponentielle, la section 4.2 détaille la théorie mathématique qui sous-tend l'évaluation des messages d'ACYCLIC-SOLVING en complexité polynomiale. En raison de la limitation d'espace, nous formulons les lemmes et théorèmes, et laissons les preuves dans un rapport technique [1]. Enfin, la section 4.3 introduit RADPRO et compare les complexités en nombre de messages et en temps avec celles d'ACYCLIC-SOLVING.

## 4.1 Solution directe

Notons que la valeur d'une allocation, d'après l'équation 3 est la somme des fonctions utilité, une par prosommateur. Ainsi, on peut directement faire correspondre l'EAP à un COP. Lorsque l'EAP est acyclique, le réseau de contrainte l'est également et on peut ainsi directement appliquer ACYCLIC-SOLVING (voir section 2). Notons maintenant que le séparateur $s_j$ entre chaque nœud $j$ et son parent $p_j$ correspond soit avec l'unique variable $y_{ip_j}$ si $p_j$ est un voisin sortant de $j$ ou avec $y_{p_jj}$ si $p_j$ est un voisin entrant de $j$. Dans tous les cas, la taille du message $\mu_{j\to p_j}$ est le nombre d'états possibles pour le lien entre $j$ et $p_j$, et le message peut être interprété comme la communication de l'utilité de chacun de ces états au réseau composé des prosommateurs dans le sous-arbre de racine $j$. Ainsi, la quantité d'information communiquée nécessaire à l'exécution de ACYCLIC-SOLVING est très faible.

L'expression du message de $j$ à son parent peut être obtenue en particularisant l'équation générale d'ACYCLIC-SOLVING (équation **??**). Lorsque le parent $p_j$ est un voisin sortant, $j$ envoie le message suivant :

$$\mu_{j\to p_j}(\mathbf{y}_{jp_j}) = \max_{\mathbf{Y}_{j-p_j}} v_j(\mathbf{y}_{jp_j}, \mathbf{Y}_{j-p_j}) + \sum_{k\in out(j)\setminus\{p_j\}} \mu_{j\to k}(\mathbf{y}_{jk}) + \sum_{i\in in(j)} \mu_{i\to j}(\mathbf{y}_{ij}) \quad (4)$$

et lorsque le parent $p_j$ est un voisin entrant, $j$ envoie le message suivant :

$$\mu_{j\to p_j}(\mathbf{y}_{p_jj}) = \max_{\mathbf{Y}_{j-p_j}} v_j(\mathbf{y}_{p_jj}, \mathbf{Y}_{j-p_j}) + \sum_{k\in out(j)} \mu_{j\to k}(\mathbf{y}_{jk}) + \sum_{i\in in(j)\setminus\{p_j\}} \mu_{i\to j}(\mathbf{y}_{ij}) \quad (5)$$

où $\mathbf{Y}_{j-p_j}$ représente une affectation de valeurs à chacune des variables de $Y_j$ exceptée $y_{jp_j}$ (ou $y_{p_jj}$). En accord avec la section 2, l'évaluation des messages prend un temps exponentiel selon le nombre de variables dans la portée de la contrainte, qui dans ce cas correspond au nombre de voisins du prosommateur. Le coût de calcul pour un prosommateur $j$ pour évaluer le message destiné à son parent est en $\mathcal{O}((2C_j+1)^{N_j})$, où $C_j$ est la capacité du lien le plus puissant entre $j$ et un agent voisin, et $N_j$ est le nombre de voisins de $j$. Comme expliqué en section 2, il est possible d'utiliser du *bookkeeping* durant l'évaluation des messages pour réduire la complexité du calcul de la meilleure affectation (équation **??**) en temps constant. Ainsi ACYCLIC-SOLVING sera efficace dès lors que le degré des nœuds du réseau est faible. Ceci peut se

produire dans des scénarios ruraux [13], cependant dans des zones urbaines [20] il a été analysé que le degré des nœuds peut suivre une distribution géométrique avec certains atteignant des degrés supérieurs à 35. Ceci entrave l'usage direct d'Acyclic-Solving dans de tels scénarios. Des approches par programmation dynamique similaires pour la réduction des émissions de $CO_2$ dans les réseaux électriques ont été qualifiée de fonctionnelle avec des degrés maximums égaux à 4 [11].

### 4.2 Calcul efficace des messages

L'évaluation des messages par les équations 4 et 5 prend un temps exponentiel selon le nombre de voisins du prosommateur. Ainsi, pour des réseaux denses, cela peut fortement limiter l'applicabilité de l'algorithme. Pour palier ce problème, nous introduisons une algèbre des valuations dont le prosommateur peut prendre avantage lors de l'évaluation de ses messages.

**Objets et opérations des *valuations*.** Notre objectif est de construire une algèbre nous permettant d'accélérer l'évaluation des messages. L'objet mathématique principal de l'algèbre, appelé *valuation*, a déjà été présenté dans la définition 1. A la fois les offres $o_j$ et les messages $\mu_{j\to p_j}$ peuvent directement être représentés par des valuations. Ces dernières peuvent à leur tour être implantées au moyen de tables de hachage. Dans la suite, nous introduisons les opérations nécessaires pour l'évaluation des messages. Tout d'abord considérons le cas où un prosommateur $j$ avec un seul voisin entrant $p_j$. Le message à envoyer d'après l'équation 5 sera tout simplement une copie de l'offre $o_j$. Cependant, comme les état possibles de $y_{p_j j}$ sont limités par la capacité du lien, nous avons besoin d'une opération permettant de supprimer de la valuation $o_j$ les valeurs hors de ces limites. Cette opération est appelée *restriction*.

**Définition 2.** *Etant donnés une valuation $\alpha$, et un sous-ensemble $D \subseteq \mathbb{Z}$ on définit $\alpha[D]$, la* restriction de $\alpha$ à $D$ *comme*

$$\alpha[D](k) = \begin{cases} \alpha(k) & k \in D \\ -\infty & \textit{sinon.} \end{cases}$$

Ainsi, le message d'un prosommateur $j$ à son unique voisin entrant $p_j$ peut être écrit comme $\mu_{j\to p_j} = o_j[D_{p_j j}]$.

Considérons maintenant le cas d'un prosommateur $j$ ayant un unique voisin sortant $p_j$. Ici, l'équation d'équilibre de flux pour $j$ montre que $net(\mathbf{y}_{jp_j}) = -\mathbf{y}_{jp_j}$. Ainsi, le message à envoyer, d'après l'équation 4 est la version symétrique d'$o_j$, restreinte à la capacité du lien. Nous appelons *complément* l'opération permettant d'obtenir la version symétrique d'une valuation.

**Définition 3.** *Etant donnée une valuation $\alpha$, on définit $\overline{\alpha}$, le* complément de $\alpha$ *comme*

$$\overline{\alpha}(k) = \alpha(-k)$$

Ainsi, le message de $j$ à son unique voisin entrant $p_j$ peut s'écrire comme $\mu_{j\to p_j} = \overline{o_j}[D_{p_j j}]$.

Enfin, prenons le cas d'un prosommateur $j$ ayant son parent $p_j$ comme voisin entrant et un voisin sortant $k$. Ici l'équation 5 est réduite à

$$\mu_{j\to p_j}(\mathbf{y}_{p_j j}) = \max_{\mathbf{y}_{jk}} \left( v_j(\mathbf{y}_{p_j j}, \mathbf{y}_{jk}) + \mu_{j\to k}(\mathbf{y}_{jk}) \right) = \max_{\mathbf{y}_{jk}} \left( o_j(\mathbf{y}_{p_j j} - \mathbf{y}_{jk}) + \mu_{j\to k}(\mathbf{y}_{jk}) \right) \quad (6)$$

Notons que pour certaines des affectations $\mathbf{y}_{p_j j}, \mathbf{y}_{jk}$, il peut arriver que $o_j$ soit non défini pour l'équilibre énergétique $\mathbf{y}_{p_j j} - \mathbf{y}_{jk}$. On peut éviter cette situation en introduisant une nouvelle opération sur les valuations, l'*agrégation*.

**Définition 4.** *Etant données deux valuations $\alpha$ et $\beta$, on définit $\alpha\cdot\beta$, l'*agrégation de $\alpha$ et $\beta$ *comme*

$$(\alpha \cdot \beta)(k) = \max_{\substack{i,j \\ k=i+j}} \alpha(i) + \beta(j) \quad (7)$$

Maintenant le message de $j$, lorsque son parent $p_j$ est un voisin entrant et $k$ est un voisin sortant, peut être écrit comme $\mu_{j\to p_j} = (o_j \cdot \mu_{k\to j})[D_{p_j j}]$.

**Structure de l'algèbre des *valuations*.** Maintenant que les valuations et leurs opérations ont été définies, nous présentons la structure algébrique que nous avons créée, sous la forme de deux lemmes. Concentrons-nous d'abord sur l'agrégation.

**Lemme 1.** *L'ensemble de valuations muni de l'opération d'agrégation forme un monoïde commutatif, $\diamond$ étant l'élément neutre. Ceci signifie que pour toutes valuations $\alpha$, $\beta$, et $\gamma$ nous avons*

$$\alpha \cdot \beta = \beta \cdot \alpha$$
$$(\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma)$$
$$\alpha \cdot \diamond = \alpha$$

D'un point de vue mathématique le lemme 1 nous permet d'écrire des expressions telles que $\alpha\cdot\beta\cdot\gamma$ qui n'établit aucun ordre spécifique dans lequel les agrégations doivent être effectuées. On peut ainsi étendre la définition d'agrégation de paires de valuations à des ensembles de valuations. Nous définissons cette agrégation n-aire de valuations dans $A = \{\alpha_1, \ldots, \alpha_n\}$ comme $(\prod_{i=1}^n \alpha_i)(k) = \max_{\substack{j_1,\ldots,j_n \\ \sum_{i=1}^n j_i = k}} \sum_{i=1}^n \alpha_i(j_i)$. Remarquons que nous avons $\prod_{i=1}^n \alpha_i = \alpha_1 \cdot \ldots \cdot \alpha_n$.

**Algorithme 2 :** Agrégation de deux valuations

```
γ ← ⋄
pour i ∈ FVD(α) faire
    pour j ∈ FVD(β) faire
        γ(i + j) ← max(γ(i + j), α(i) + β(j))
```

Ainsi, l'agrégation n-aire d'un ensemble fini de valuations peut être évalué en utilisant uniquement des agrégations binaires de la définition 4. Du point de vue de la complexité, le lemme 1 nous permet de grouper les opérations d'agrégation de la manière qui nous arrange. Ainsi, il est possible de suivre un ordre séquentiel, ou d'évaluer les messages de manière arborescente dans le cas où cela peut être plus efficace.

Maintenant intéressons-nous à la caractérisation des relations entre les différentes opérations.

**Lemme 2.** *Le complément définit un automorphisme involutif du monoïde des valuations, c'est-à-dire que pour toutes valuations $\alpha$ et $\beta$ nous avons*

$\overline{\alpha \cdot \beta} = \overline{\alpha} \cdot \overline{\beta}$
$\overline{\overline{\alpha}} = \alpha$
$\overline{\diamond} = \diamond$
$\bar{\cdot} : \Phi \to \Phi$ *est une association bijective*

*De plus,* $\overline{\alpha[D]} = (\overline{\alpha})[-D]$ *où* $-D = \{-x | x \in D\}$.

Certaines de ces propriétés vont se montrer très pratiques pour travailler sur des expressions impliquant l'agrégation, le complément et la restriction.

**Efficacité de l'agrégation.** Intéressons-nous maintenant au coût de calcul de l'agrégation d'un ensemble de valuations. L'algorithme 2 peut être utilisé pour implanter l'agrégation, avec le résultat suivant :

**Lemme 3.** *L'agrégation de deux valuations $\alpha$ et $\beta$ peut être effectuée en temps $\mathcal{O}(n_\alpha \times n_\beta)$. De plus, $n_{\alpha\cdot\beta} \leq n_\alpha \times n_\beta$.*

De là, nous pouvons montrer que l'évaluation de l'agrégation de $M$ valuations de taille $N$ peut être effectuée en temps $O(N^M)$.

Notre objectif était d'améliorer les calculs d'agrégation des messages. Cependant, nous pouvons observer que dans le cas le plus général, l'agrégation de $M$ messages prend un temps exponentiel. Prenons maintenant avantage d'une des particularités des messages à agréger pour nous permettre de calculer l'agrégation en temps polynomial. Remarquons que le FVD d'un message $\mu_{i\to j}$ est toujours inclus dans un intervalle réduit $D_{ij} = [-c_{ij}..c_{ij}]$ autour de 0. Nous disons qu'une valuation $\alpha$ est *de capacité restreinte* $C$ si $FVD(\alpha) \subseteq [-C..C]$. Pour des valuations de capacité restreinte nous obtenons le résultat suivant.

**Lemme 4.** *L'agrégation de deux valuations $\alpha$ et $\beta$ de capacité restreinte $C$ peut être faite en temps $\mathcal{O}(C^2)$. De plus, $\alpha \cdot \beta$ est une valuation de capacité restreinte $2C$.*

Remarquons que, tandis que pour les valuations en général la taille de la valuation agrégée grandit de manière quadratique, pour les valuations à capacité restreinte, elle grandit de manière linéaire. Ceci devient fondamental pour prouver le résultat suivant, qui nous dit qu'il est possible d'agréger des valuations en temps polynomial dès lors qu'elles sont de capacité restreinte.

**Lemme 5.** *Etant donné un ensemble $A$ de $M$ valuations à capacité restreinte $C$ on peut évaluer son agrégation $\prod_{\alpha\in A} \alpha$ en temps $\mathcal{O}(M^2C^2)$.*

La preuve utilise la propriété d'associativité pour diviser l'agrégation en deux agrégations plus petite, chacune ayant la moitié des valuations, pour évaluer ces agrégations plus réduites et finalement agréger les résultats.

**Usage de l'algèbre pour l'évaluation des messages.** Le théorème suivant montre qu'il est possible de fournir une expression des messages d'Acyclic-Solving des équations 4 et 5 en termes d'opérations de l'algèbre des valuations et ceci en temps polynômial.

**Théorème 1.** *Le message de $j$ à son parent et voisin entrant $p_j$ défini dans l'équation 5 peut être évalué comme*

$$\mu_{j\to p_j} = \left( o_j \cdot \prod_{k\in out(j)} \mu_{j\to k} \cdot \prod_{i\in in(j)\setminus\{p_j\}} \overline{\mu_{i\to j}} \right)[D_{p_j j}] \tag{8}$$

*et le message de $j$ à son parent et voisin sortant $p_j$ défini dans l'équation 4 peut être évalué comme*

$$\mu_{j\to p_j} = \left( \overline{o_j} \cdot \prod_{k\in out(j)\setminus\{p_j\}} \overline{\mu_{j\to k}} \cdot \prod_{i\in in(j)} \mu_{i\to j} \right)[-D_{jp_j}]. \tag{9}$$

*De plus la complexité en temps de l'évaluation de chaque message est en $\mathcal{O}(N_j C_j n_{o_j} + N_j^2 C_j^2)$ où $N_j$ est le nombre de voisins de $j$ et $C_j$ est la plus grande capacité de tous les liens connectés à $j$.*

La preuve repose sur la transformation des expressions des équations 4 et 5 en une agrégation n-aire qui est ensuite transformée en un ensemble d'agrégations binaires manipulées à l'aide du lemme 2. Le résultat de complexité repose sur le lemme 5.

| | Acyclic-Solving | RadPro |
|---|---|---|
| Messages | $\mathcal{O}(nC_{max})$ | $\mathcal{O}(nC_{max})$ |
| Temps | $\mathcal{O}(n(2C_{max}+1)^{N_{max}})$ | $\mathcal{O}(nN_{max}^2C_{max}^2)$ |

Table 1 – Complexités théoriques comparées

### 4.3 Envoi de message dans RadPro

Le théorème 1 montre que les messages d'Acyclic-Solving peuvent être évalués en temps polynomial pour EAP. Nous appelons RadPro le nouvel algorithme résultant de l'exécution d'Acyclic-Solving comme décrit dans l'algorithme 1 mais en évaluant les messages en utilisant les équations 8 et 9. Notons que nous ne modifions que l'algorithme utilisé pour évaluer les messages. Comme les messages sont échangés de la même manière pour les deux algorithmes, le nombre et la taille des messages restent les mêmes : $2n$ messages envoyés, avec $n$ le nombre de prosommateurs, pour une taille de messages bornée par $2C_{max}+1$, où $C_{max}$ est la plus grande capacité du réseau. Ainsi, le nombre de valeurs échangées par les deux algorithmes est en $\mathcal{O}(nC_{max})$.

Concernant la complexité en temps, comme précisé dans la section 4.1, dans Acyclic-Solving la complexité pour un prosommateur $j$ d'évaluer le message à son parent est en $\mathcal{O}((2C_j+1)^{N_j})$ où $C_j$ est la plus grande capacité des liens connectés à $j$ et $N_j$ est le nombre de voisins de $j$. Ainsi, la complexité globale est bornée par $\mathcal{O}(n(2C_{max}+1)^{N_{max}})$ où $N_{max}$ est le nombre de voisins du prosommateur le plus connecté et $n$ est le nombre total de prosommateurs. D'autre part, la complexité en temps pour évaluer un message dans RadPro, comme vu dans le théorème 1 est en $\mathcal{O}(N_jC_jn_{o_j}+N_j^2C_j^2)$. Comme le nombre de messages à évaluer est égal au nombre de prosommateurs dans le réseau, la complexité globale est bornée par $\mathcal{O}(n(N_{max}C_{max}n_{max}+N_{max}^2C_{max}^2))$ où $n_{max}$ est la taille de l'offre la plus large. En supposant que dans le cas le plus commun $n_{max} < N_{max}C_{max}$, on peut simplifier l'expression en une complexité finale en $\mathcal{O}(nN_{max}^2C_{max}^2)$.

## 5 Expérimentation et évaluation

Dans cette section nous évaluons le temps d'exécution de RadPro et le comparons avec celui d'Acyclic-Solving et ceux de solveurs MIP (*Mixed Integer Programming*) classiques, après avoir transformé EAP en programme linéaire, suivant le modèle de [1].

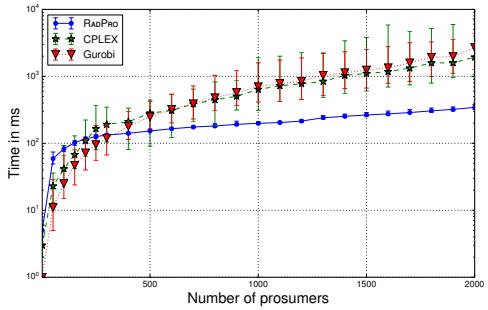
**Comparaison avec des solveurs MIP**. Ici nous comparons les performances de RadPro à celles de solveurs commerciaux (IBM CPLEX et Gurobi). Pour ce faire, nous générons des instances d'EAP acycliques dont la structure simule la topologie de réseaux radiaux. Conformément à [20], une bonne approximation de la distribution empirique du degré des nœuds dans les SG réelles est obtenue grâce à une distribution géométrique. Ainsi, nous générons des arbres dont les degrés des nœuds suivent une distribution géométrique, avec $p = 0.5$, pour simuler la structure de larges réseaux de distribution. Chaque nœud de l'arbre représente un prosommateur, qui est déterminé de manière aléatoire comme étant soit un producteur (10% des prosommateurs) soit un consommateur (90% restants). Pour chaque consommateur $j$, l'offre $o_j$ est :

$$o_j(t) = \begin{cases} t \cdot price_j & t \in [min_j..max_j] \\ 0 & t = 0 \\ -\infty & sinon \end{cases}$$

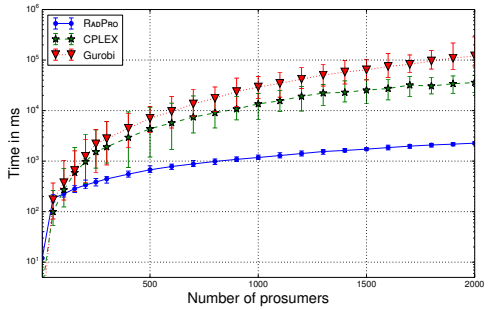
où $min_j$ et $max_j$ sont les nombres minimum et maximum d'unités permises pour le prosommateur $j$, et $price_j$ est le prix par unité d'énergie pour ce prosommateur. Le nombre maximum d'unités $max_j$ est déterminé suivant une distribution normale $\mathcal{N}(\kappa, \frac{\kappa}{2})$ où $\kappa$ est un paramètre de capacité. Le nombre minimum d'unités $min_j$ est ensuite déterminé suivant une distribution uniforme $\mathcal{U}(1, max_j)$. Le prix $price_j$ est déterminé suivant une loi normale $\mathcal{N}(1, 0.5)$. Les producteurs sont générés suivant la même procédure, à l'exception de l'offre qui est définie dans l'intervalle $[-max_j.. - min_j]$. La capacité d'un lien entre deux nœuds est fixée au nombre maximum d'unités produites ou requises par chacun des prosommateurs. Les expérimentations ont été conduites sur un unique Intel Core i7 2.66GHz, avec 8GB RAM. RadPro a été développé en Java.

Nous analysons deux capacités différentes $\kappa = 10$ et $\kappa = 100$. Le nombre d'agents (prosommateurs) $n$ varie par pas de 100 jusqu'à 2000. Pour chaque $\kappa$ et $n$, nous avons généré 100 instances différentes. La figure 3a montre la valeur médiane et l'écart interquartile 5% – 95% du temps d'exécution pour $\kappa = 10$, et la figure 3b pour $\kappa = 100$. Dans les deux cas, CPLEX et Gurobi sont plus rapides quand le nombre de prosommateurs est faible. Cependant, RadPro passe mieux à l'échelle et pour $n = 2000$ il est plus rapide de plus d'un ordre de grandeur que CPLEX, qui lui surpasse Gurobi. Dans le scénario le plus extrême ($n = 2000, \kappa = 100$), RadPro affiche un temps médian inférieur à 2.3 secondes alors que CPLEX a besoin de plus de 35.8 secondes et Gurobi plus de 2 minutes.

**Efficacité du calcul des messages.** Pour souligner le bénéfice de l'évaluation efficace des messages avec une taille de voisinage grandissante, nous exécutons RadPro sur des instances




(a) $\kappa = 10$




(b) $\kappa = 100$

FIGURE 3 – Temps d'exécution de RADPRO comparé à CPLEX et Gurobi, pour des réseaux à distribution géométrique.

plus simples basées sur des réseaux en étoile (un nœud central connecté à plusieurs voisins). les offres sont générées comme précédemment, sauf que $min_j$ et $max_j$ sont maintenant fixées à $min_j = 1$ et $max_j = \kappa$. Comme les prosommateurs font des propositions pour chaque valeur de leurs capacités, ceci représente le pire scénario possible. De plus, ceci résulte dans une capacité fixée pour les liens (tous les liens ont la même capacité $\kappa$), facilitant ainsi l'analyse des résultats. La figure 4 montre l'accélération –i.e. le rapport $\frac{time(\text{ACYCLIC-SOLVING})}{time(\text{RADPRO})}$. Par exemple, en taille $n = 7$ (un agent central ayant 6 voisins), pour $\kappa = 50$, RADPRO est 5000 fois plus rapide qu'ACYCLIC-SOLVING. Ceci souligne comment le calcul efficace des messages que nous proposons permet à RADPRO de passer à l'échelle polynomialement, alors que le temps d'exécution d'ACYCLIC-SOLVING grandit exponentiellement : à $n = 7$ et $\kappa = 50$, RADPRO prend environ 1 seconde alors qu'ACYCLIC-SOLVING prend environ 83 minutes pour résoudre une instance. Notons que ceci est également équivalent à comparer RADPRO à d'autres approches multiagents par programmation dynamique comme DPOP [16], Action-GDL [18] ou D–DYDOP [11]. La figure 5 illustre comment le temps de RADPRO est affecté par la capacité pour des réseaux plus larges (ces scénarios sont inaccessibles pour ACYCLIC-SOLVING). Ici, nous voyons que RADPRO résout un problème de capacité 100 avec 100 voisins en moins de 1 minute. Ces résultats font de RADPRO un très bon candidat pour s'attaquer à de larges EAP ayant un facteur de branchement élevé —ce qui est le cas dans les configurations urbaines.

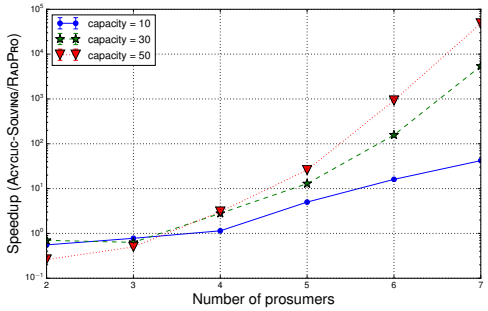



FIGURE 4 – Accélération de RADPRO par rapport à ACYCLIC-SOLVING) dans des réseaux en étoile.

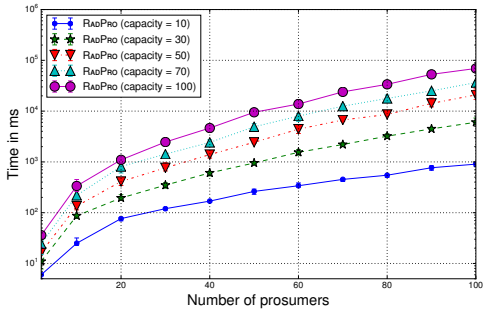



FIGURE 5 – Temps d'exécution de RADPRO dans le larges réseaux en étoile.

## 6 Conclusions et perspectives

Dans cet article nous avons étudié comment mettre en place un échange d'énergie entre prosommateurs tout en prenant en compte les limites physiques du réseau électrique. Nous avons considéré ce problème comme une problème d'optimisation, l'EAP. Pour des configurations acycliques (ce qui est un hypothèse très réaliste, compte tenu de la structure actuelle des réseaux électriques), nous avons conçu RADPRO, un nouvel algorithme de programmation dynamique pour résoudre de manière optimale et efficace l'EAP acyclique. RADPRO se base sur ACYCLIC-SOLVING car il repose sur un réseaux arborescent, qui est parfaitement adapté aux réseaux radiaux. Cependant, à cause du temps exponentiel du calcul des messages dans ACYCLIC-SOLVING, nous avons apporté à RADPRO une évaluation efficace des messages grâce à une nouvelle *algèbre des valuations*. Grâce à celle-ci, RADPRO peut efficacement calculer les messages en temps polynomial, et ainsi pallier les limitations d'ACYCLIC-SOLVING en réduisant le temps de calcul de plusieurs ordres de grandeurs. De plus, nos résultats expérimentaux montrent que RADPRO surclasse largement CPLEX et Gurobi

en temps de calcul pour trouver l'allocation optimale d'une EAP, avec une taille de marché grandissante. En général, nos expériences démontrent que les complexités en temps et en nombre de messages de RadPro le positionne clairement comme un algorithme optimal et adapté aux larges réseaux.

Voici quelques pistes de travail futur. Premièrement, la nature par envoi de messages de RadPro offre la possibilité de résoudre EAP de manière centralisée ou distribuée. Ceci est en ligne directe avec les défis dans [3], et ouvre la possibilité d'explorer les impacts économiques et environnementaux de la mise en place d'un protocole de négociation pair-à-pair. Deuxièmement, comme les futures infrastructures des SG sont amenées à évoluer et éventuellement contenir des boucles (e.g. par maillage), nous projetons d'explorer comment améliorer RadPro pour gérer efficacement les EAP cycliques. Troisièmement, la vitesse de RadPro ainsi que la convergence d'autres technologies (comme le *smart metering*) rendent la *conscience énergétique* en temps réel possible. Et ainsi, de manière alternative aux enchères centralisées journalières, RadPro permet de considérer des fenêtres de temps de l'ordre de la minute, donnant ainsi la possibilité aux prosommateurs de mettre en place des échanges en temps presque réel.

## Références


[1] J. Cerquides, G. Picard, and J. A. Rodríguez-Aguilar. Designing a marketplace for the trading and distribution of energy in the smart grid - extended version with proofs, available at http://bit.ly/18PSEb0, 2015.

[2] R. Dechter. *Constraint processing*. Morgan Kauffman, 2003.

[3] European Technology Platform. SmartGrids SRA 2035. Strategic Research Agenda. Update of the SmartGrids SRA 2007 for the needs by the year 2035, March 2012.

[4] Federation of German Industries (BDI). The Energy Industry on the Way to the Internet Age. BDI publication No.439, 2010.

[5] T. Gonen. *Electric power distribution engineering*. CRC press, 2014.

[6] Greenpeace. Decentralising power : An energy revolution for the 21st century. http://bit.ly/1xf1RCk, 2005.

[7] L. L. Grigsby. *Electric Power Generation, Transmission, and Distribution*. CRC press, 2012.

[8] D. Ilic, P. G. Da Silva, S. Karnouskos, and M. Griesemer. An energy market for trading electricity in smart grid neighbourhoods. In *6th IEEE International Conference on Digital Ecosystems Technologies (DEST), 2012*, pages 1–6. IEEE, 2012.

[9] K. Kok, B. Roossien, P. MacDougall, O. van Pruissen, G. Venekamp, R. Kamphuis, J. Laarakkers, and C. Warmer. Dynamic pricing by scalable energy management systems—field experiences and simulation results using powermatcher. In *Power and Energy Society General Meeting, 2012 IEEE*, pages 1–8. IEEE, 2012.

[10] S. Lamparter, S. Becher, and J.-G. Fischer. An agent-based market platform for smart grids. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems : Industry track*, pages 1689–1696. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[11] S. J. O. Miller. *Decentralised Coordination of Smart Distribution Networks using Message Passing*. PhD thesis, University of Southhampton, 2014.

[12] J. Mockus. On simulation of the nash equilibrium in the stock exchange contest. *Informatica*, 23(1) :77–104, 2012.

[13] B. Neagu and G. Georgescu. Optimization Possibilities for Radial Electric Energy Distribution Network Routes. *Bul. Inst. Politehnic, Iaşi, LIX (LXIII)*, (Lxiii), 2013.

[14] M. A. Olson, S. J. Rassenti, V. L. Smith, M. L. Rigdon, and M. J. Ziegler. Market design and motivated human trading behavior in electricity markets. In *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences, 1999. HICSS-32.*, pages 1–27. IEEE, 1999.

[15] Y. K. Penya and N. R. Jennings. Optimal combinatorial electricity markets. *Web Intelligence and Agent Systems*, 6(2) :123–135, 2008.

[16] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. *IJCAI International Joint Conference on Artificial Intelligence*, pages 266–271, 2005.

[17] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings. Putting the 'smarts' into the smart grid : A grand challenge for artificial intelligence. *Commun. ACM*, 55(4) :86–97, Apr. 2012.

[18] M. Vinyals, J. A. Rodríguez-Aguilar, and J. Cerquides. Constructing a unifying theory of dynamic programming DCOP algorithms via the generalized distributive law. *Autonomous Agents and Multi-Agent Systems*, 3(22) :439–464, May 2011.

[19] P. Vytelingum, S. D. Ramchurn, T. D. Voice, A. Rogers, and N. R. Jennings. Trading agents for the smart electricity grid. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems : volume 1-Volume 1*, pages 897–904. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[20] Z. Wang, A. Scaglione, and R. J. Thomas. The Node Degree Distribution in Power Grid and Its Topology Robustness under Random and Selective Node Removals. *2010 IEEE International Conference on Communications Workshops*, (1) :1–5, May 2010.

[21] B. M. Weedy, B. J. Cory, N. Jenkins, J. Ekanayake, and G. Strbac. *Electric power systems*. John Wiley & Sons, 2012.

[22] T. K. Wijaya, K. Larson, and K. Aberer. Matching demand with supply in the smart grid using agent-based multiunit auction. *2013 Fifth International Conference on Communication Systems and Networks (COMSNETS)*, pages 1–6, Jan. 2013.

[23] P. Wolfe. The implications of an increasingly decentralised energy system. *Energy policy*, 36(12) :4509–4513, 2008.