Reduction of electricity consumption peaks and optimization problems induced on the demand side

Chloe Desdouits

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE la Communauté UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Automatique Productique**

Arrêté ministériel : 7 Août 2006

Présentée par

Chloé Desdouits

Thèse dirigée par **Mazen Alamir**
et codirigée par **Rodolphe Giroudeau**

préparée au sein **Gipsa-lab, LIRMM**
et de **EEATS**

Réduction des pics de consommation d'électricité et problèmes d'optimisation induits pour les consommateurs

Thèse soutenue publiquement le **10 février 2017**,
devant le jury composé de :

M Christophe Picouleau
Professeur du CNAM - CEDRIC, Président
M Didier Dumur
Professeur de CentraleSupélec, Rapporteur
M Christian Artigues
Directeur de recherche du LAAS, Rapporteur
M Claude Le Pape
VP Optimization and Analytics, Schneider Electric, Examinateur
M Mazen Alamir
Directeur de recherche au Gipsa-lab, Directeur de thèse
M Rodolphe Giroudeau
Maître de Conférences au LIRMM, Co-Directeur de thèse

GRENOBLE ALPES UNIVERSITY

DOCTORAL THESIS IN AUTOMATIC CONTROL

Reduction of electricity consumption peaks and optimization problems induced on the demand side

Author:
Chloé DESDOUITS

Supervisors:
* Dr. Mazen ALAMIR
† Dr. Rodolphe GIROUDEAU
§ Dr. Claude LE PAPE

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in

§ Schneider Electric
Global Technology
AA& P
Grenoble, France

* GIPSA-lab
Control Systems Department
SysCo
Saint Martin d'Hères, France

† LIRMM
Computer Science Department
MAORE
Montpellier, France

Defended on **February 10, 2017**

Jury members

President
Prof. Christophe PICOULEAU

Referee
Prof. Didier DUMUR

Referee
Dr. Christian ARTIGUES

"Bien qu'on ait du cœur à l'ouvrage, l'art est long et le temps est court."

Charles Baudelaire

GRENOBLE ALPES UNIVERSITY

Abstract

EEATS doctoral school

Doctor of Philosophy

Reduction of electricity consumption peaks and optimization problems induced on the demand side

by Chloé DESDOUITS

Keywords: Energy optimization, Real-time control, Scheduling, Linear Programming, Software architecture.

While concerns about global warming have never been so important, one of its first causes: global electricity consumption, is still growing. One way to stem the phenomenon could be to better balance demand and production, in order to switch on less big production groups and to allow the integration of more renewable production sources. The new paradigm of electricity market incites customers to reduce their electricity consumption peak and to shift their consumption when the demand is lower, by introducing economical incentives. Thus, new optimization algorithms and methodologies are needed at the customers side to optimize power usage over time. Schneider Electric has proposed, in the context of the Arrowhead European project, to study three application use-cases: an elevator with multiple electricity sources, a manufacturing plant, and a drinking water network. For each of these use-cases, a methodology to optimize power consumption peaks (sometimes through an electricity cost function) is given, as well as optimization algorithms. For the multisource elevator case, two coupled controllers are proposed: one at the strategic level solving a linear problem, the other one rule-based at the tactical level. For the manufacturing plant, the methodology we used to monitor, build energy models, and finally optimize is explained. Furthermore, three linear formulations are given, as well as a simple local search procedure and a naive constraint satisfaction formulation to handle the NP-hard scheduling problem. For the water network use-case, a quadratically constrained formulation is used to compare optimized pumping plans with the Business As Usual tactic. The methods proposed bring between 10.9% to 114% savings on the energy bill, depending on the context. Moreover, they allow electricity consumers to participate in the demand-response market. Finally, the knowledge extracted from our three use-cases is summarized, and guidelines are given to optimize the electricity bill of any electricity consumer system.

UNIVERSITÉ GRENOBLE ALPES

Résumé

École doctorale EEATS

Docteur en Philosophie

Réduction des pics de consommation d'électricité et problèmes d'optimisation induits pour les consommateurs

par Chloé DESDOUITS

Mots clés : Optimisation énergétique, Contrôle en temps réel, Ordonnancement, Programmation linéaire, Architecture logicielle.

Tandis que le réchauffement climatique devient de plus en plus préoccupant, l'une de ses causes principales : la consommation d'électricité, continue à croître. Une des manières d'endiguer le phénomène pourrait être de mieux équilibrer la consommation et la production, afin d'allumer moins de gros groupes de production, et de permettre l'intégration de plus de sources renouvelables. Le nouveau paradigme du marché de l'électricité incite les consommateurs à réduire leur pic de consommation, et à différer leur consommation quand la demande est moindre, à l'aide d'incitations tarifaires. De nouveaux algorithmes d'optimisation, et de nouvelles méthodologies sont donc nécessaires pour optimiser la puissance d'électrique utilisée par les consommateurs. Schneider Electric a proposé, à travers le projet européen Arrowhead, d'étudier trois cas applicatifs : un ascenseur avec plusieurs sources énergétiques, une usine manufacturière et un réseau d'eau potable. Pour chacune de ces trois applications, une méthodologie pour optimiser les pics de puissance consommée (parfois à l'aide d'une fonction de coût de l'électricité) est donnée, ainsi que des algorithmes d'optimisation. Dans le cas de l'ascenseur multi-sources, deux contrôleurs couplés sont proposés : l'un au niveau stratégique résolvant un problème linéaire, et l'autre à base de règles au niveau tactique. Dans le cas de l'usine, la méthodologie que nous avons utilisée pour faire des mesures, construire des modèles énergétiques, et finalement optimiser est présentée. De plus, trois formulations linéaires, ainsi qu'une procédure de recherche locale, et une formulation naïve de programmation par contraintes sont données afin de résoudre le problème d'ordonnancement classifié NP-difficile. Dans le cas du réseau d'eau, une formulation à contraintes quadratiques est utilisée pour comparer des plans de pompage optimisés avec la tactique utilisée habituellement dans le réseau. Les méthodes proposées entraînent des gains sur la facture énergétique de 10.9% à 114%, dépendamment du contexte. De plus, elles permettent aux consommateurs de participer au nouveau marché de l'énergie. Finalement, les connaissances retirées de l'étude de ces trois pilotes sont résumées, et des lignes directrices sont données pour l'optimisation de la facture énergétique d'un système quelconque consommant de l'énergie.

Résumé long

Réduction des pics de consommation d'électricité et problèmes d'optimisation induits pour les consommateurs

par Chloé DESDOUITS

Tandis que le réchauffement climatique devient de plus en plus préoccupant, l'une de ses causes principales : la consommation d'électricité, continue à croître. Une des manières d'endiguer le phénomène pourrait être de mieux équilibrer la consommation et la production, afin d'allumer moins de gros groupes de production, et de permettre l'intégration de plus de sources renouvelables. Le nouveau paradigme du marché de l'électricité incite les consommateurs à réduire leur pic de consommation, et à différer leur consommation quand la demande est moindre, à l'aide d'incitations tarifaires. Cependant, seuls les gros consommateurs peuvent participer au marché de l'énergie. Afin d'ouvrir cette possibilité aux petits consommateurs, de nouveaux acteurs émergent : les aggrégateurs. Comme leur nom l'indique, ils agrègent la flexibilité des nombreux petits consommateurs pour former des gros blocs d'énergie qu'ils peuvent ensuite revendre. De nouveaux algorithmes d'optimisation, et de nouvelles méthodologies sont donc nécessaires pour optimiser la puissance d'électrique utilisée par les consommateurs dans le temps. Schneider Electric propose, à travers le projet européen Arrowhead, d'étudier trois cas applicatifs : un ascenseur avec plusieurs sources énergétiques, une usine manufacturière et un réseau d'eau potable. Le but de cette thèse est d'étudier différents moyens de résoudre ces problèmes et de comparer les méthodes en terme d'efficacité et de généricité.

Ces travaux se sont déroulés au sein de l'entreprise Schneider Electric, de deux laboratoires de recherche : le Gispa-lab et le LIRMM, et dans le cadre du projet européen Arrowhead. Claude Le Pape, docteur en informatique et expert en ordonnancement, supervise cette thèse au sein de Schneider Electric, le spécialiste mondial de la gestion de l'énergie et de l'automatisation. Le directeur de cette thèse, Mazen Alamir, directeur de recherche et spécialiste du contrôle à base des modèles non-linéaires, fait partie d'un laboratoire internationalement reconnu pour ses recherches en contrôle et traitement du signal : le Gipsa-lab. Rodolphe Giroudeau, Maître de Conférences HDR, spécialiste de l'approximabilité et de la complexité des problèmes d'ordonnancement au sein du LIRMM, est co-directeur. Enfin, cette thèse est partiellement financée par le projet Arrowhead, qui fédère plusieurs dizaines de partenaires industriels et académiques, et vise à permettre l'automatisation collaborative par des dispositifs embarqués communicants.

Dans le cadre de ce projet, trois pilotes devaient être développés. Le premier vise à l'optimisation de la facture énergétique d'un ascenseur multi-sources, sous contrainte de puissance maximum. Le second concerne l'optimisation énergétique de la production dans une usine d'armoires électriques. Le troisième et dernier a pour but la génération de plans de pompage optimaux pour un réseau d'eau potable. Plusieurs façon de résoudre ces problèmes peuvent être envisagées. La première consiste à oublier les détails physiques des processus industriels pour se ramener à un problème d'ordonnancement sous contrainte de ressources (incluant l'énergie). De tels problèmes sont connus pour être parmi les plus difficiles des problèmes NP-difficiles. De nombreux travaux de recherches traitent de la résolution des problèmes d'ordonnancement sous contrainte de ressources, souvent à l'aide de méthodes de résolution mixtes. Cependant la considération de l'énergie comme une contrainte de ressource, ou encore comme un objectif économique est un axe de recherche plus récent, et moins étudié. La dimension multi-critères du problème d'ordonnancement s'en trouve renforcée. Il peut donc s'avérer plus efficace de décomposer le problème global en sous-problèmes à résoudre séparément. D'autre part, coupler des modules d'optimisation et de simulation, ou encore utiliser du contrôle à base de modèles (où le problème est résolu de nouveau à intervalles de temps réguliers) peut être envisagé.

En ce qui concerne le troisième problème de plans de pompage pour un réseau d'eau potable, une autre thèse a été lancée sur le sujet (suite au stage de Gratien Bonvin chez Schneider Electric). Par conséquent, un premier modèle d'optimisation quadratique a été développé mais aucun chapitre de ce manuscrit ne lui est consacré. Les premiers résultats prédisent une diminution de 15% de la facture énergétique.

Dans les bâtiments accueillant du public, plusieurs équipements doivent éviter de manquer d'énergie en cas de coupure de courant (éclairage d'urgence, ascenseurs dédiés à évacuer les handicapés, ...). Dans ce contexte, des batteries doivent être installées dans les bâtiments et peuvent être exploitées pour compenser une partie de la facture énergétique. En effet, elles peuvent être connectés à des sources d'énergie du bâtiment pour être rechargées et peuvent founir de l'énergie quand l'électricité est chère. Dans ce cas, un système électronique est nécessaire pour connecter toutes les sources d'énergie aux consommateurs et assurer que la tension nominale de chaque équipement soit respectée. Nous étudions un système où six équipements sont connectés au même « *concentrateur d'énergie* » :

- Un ascenseur qui peut consommer ou produire de l'énergie, i.e. : qui s'il monte plein ou descend vide, et produit dans le cas contraire.
- Une batterie qui a une grosse capacité énergétique mais une petite capacité en puissance.
- Une supercapacité qui a une petite capacité en énergie mais une grosse capacité en puissance. Elle peut ainsi quasiment absorber les plus gros pics de puissance de l'ascenseur.
- Le réseau électrique, à qui nous considérons pouvoir acheter et revendre de l'énergie.
- Des panneaux solaires partiellement dédiés à l'ascenseur.
- Une résistance permettant de dissiper l'énergie en surplus, par exemple lorsque l'ascenseur produit alors que la batterie est pleine et la revente impossible.

Nous appelons « *prosumers* » les entités qui consomment et/ou produisent de l'énergie. Un concentrateur d'énergie permet à chaque prosumer de consommer la puissance produite par les autres prosumers au même instant. Étant donnés : un graphe en étoile de prosumers (enraciné en le concentrateur d'énergie), une période de temps, un horizon de temps et un ensemble de fonctions de coût de l'énergie, nous définissons le problème des sources d'énergie comme suit. Quelle puissance le concentrateur doit-il attribuer à chaque prosumer, pour chaque période temps dans l'horizon, de telle manière que les pics de puissance depuis le réseau restent bornés par une constante et que la facture énergétique soit minimale ?

Ce problème a été assez peu étudié dans la littérature pour le cas des ascenseurs multisources. Il existe deux catégories de méthodes : celles à base de règles qui traitent le problème en temps réel et les méthodes exactes (parfois probabilistes) qui calculent un plan minimisant la facture énergétique. Les premières sont nécessaires pour assurer la gestion en puissance du système mais ne peuvent pas prendre en compte d'objectif économique de manière optimale. Les secondes optimisent la facture énergétique mais ne peuvent pas prendre en compte les contraintes temps-réel et modifier leur solution en cas d'imprévu. Par conséquent, nous avons choisi de proposer le couplage d'un optimiseur stratégique et d'un contrôleur local pour résoudre notre problème. Par ailleurs, un algorithme randomisé est utilisé pour trouver la meilleure configuration du système (taille des batteries par exemple) et donner des garanties probabilistes de gain économique.

Afin de gérer l'enjeu temps-réel, nous proposons un algorithme centralisé à base de règles pour le concentrateur d'énergie, le « *Contrôleur Local* ». Il met à jour sa tactique toutes les secondes. De plus, une stratégie pour choisir les sources d'énergie est calculée par un « *Optimiseur Stratégique* », qui envoie des instructions stratégiques au Contrôleur Local régulièrement. Cet optimiseur considère des périodes d'un quart d'heure et un horizon de vingt-quatre heures. Il utilise des prédictions de consommation de l'ascenseur et de production des panneaux solaires. Le problème d'optimisation stratégique est modélisé en programmation linéaire avec comme variables de décision la quantité d'énergie injectée dans (ou consommée depuis) le concentrateur. Une contrainte d'état de charge minimum des batteries est prise en compte pour parer à l'éventualité d'une coupure du réseau électrique. De plus, la somme des quantités d'énergie produites et consommée par les prosumers doit être nulle à chaque instant. Les capacités en énergie et en puissance de tous les prosumers doivent être respectées. Enfin, la quantité d'énergie achetée et vendue au réseau électrique doit être lissée au court du temps si possible. L'objectif est ensuite de minimiser la facture énergétique, ainsi que le coût de vieillissement des batteries. Une fois ce problème résolu, une instruction stratégique contenant un état de charge cible pour les batteries et une puissance moyenne à acheter (ou revendre) au réseau, est envoyée au Contrôleur Local. Le Contrôleur Local, quant à lui, établit une tactique en fonction de : la dernière instruction stratégique reçue, un ordre de priorité sur les prosumers, et les flexibilités des prosumers à l'instant considéré. Un arbre de recherche est construit avec un étage par prosumer et autant de nœuds par étage que de flexibilités des prosumers. Une recherche en profondeur dans cet arbre donne une tactique. Plusieurs paramétrisations possibles sont données pour le Contrôleur Local : l'une fait en sorte de minimiser les pics de puissance achetée au réseau, une autre (dite opportuniste) de minimiser la quantité d'énergie dissipée, et la dernière d'utiliser le moins possible les batteries, quitte à perdre de l'énergie.

Comme l'efficacité de la stratégie dépend de la précision des prédictions, une étude de la robustesse de notre méthode envers les incertitudes doit être conduite. Et ce, afin d'identifier les situations où utiliser notre solution plutôt complexe offre de meilleures résultats qu'utiliser un simple contrôleur à base de règles. Pour cela, deux cas concrets sont considérés : un premier dans lequel un client souscrit à un tarif de l'électricité qui lui interdit de dépasser une certaine limite de puissance et veut dimensionner ses batteries pour respecter cette limite ; un deuxième dans lequel la paire (contrôleur, tarif) la plus rentable doit être indiquée à un client pour son propre système, ainsi qu'un gain économique garanti. Afin de pouvoir garantir les résultats avec une probabilité donnée, un algorithme randomisé est utilisé, ainsi qu'un modèle statistique (à base de loi Gaussienne) des trajets de l'ascenseur. Différentes conceptions du système (par exemple différents dimensionnements de batteries) sont comparées, et celle qui offre la meilleure garantie de performances est conservée. L'ensemble des incertitudes considérées comprend la consommation énergétique de l'ascenseur, la production solaire et l'état initial des batteries. L'algorithme randomisé consiste à tirer un nombre N de scénarios aléatoires parmi cet ensemble d'incertitudes, N étant calculé en fonction de niveaux de garantie que l'on souhaite obtenir et du nombre de conceptions envisagées. Puis la stratégie et la tactique sont simulées pour toutes les conceptions, sur tous les scénarios, et nous vérifions que le résultat respecte bien les contraintes que l'on souhaite. Parmi les conceptions respectant toujours les contraintes, celle qui offre la meilleure garantie de performances sur le coût global est conservée.

Des expérimentations ont été menées pour évaluer les performances de la méthode de contrôle proposée, ainsi que sa robustesse aux incertitudes. Dans un premier temps, un exemple de stratégie sur un jour typique est commentée. Le tarif considéré étant un tarif heures creuses / heures pleines, la batterie est chargée la nuit avec de l'électricité bon marché et déchargée le jour quand l'électricité est chère. L'expérience suivante concerne la perte d'information induite par le fait que l'Optimiseur Stratégique considère l'énergie agrégée sur des périodes de quinze minutes. Ce choix permet de lisser l'incertitude liée aux heures de trajet de l'ascenseur mais induit le fait que l'Optimiseur Stratégique ne prévoit pas les pics de puissance. De ce fait, les batteries doivent être utilisées beaucoup plus que prévu par la stratégie. Pour pallier ce problème, la prédiction de consommation prise en compte par l'ascenseur doit elle-même tenir compte du rendement des batteries. En effet, en considérant que toute l'énergie consommée par l'ascenseur provient des batteries, nous obtenons un comportement légèrement pessimiste de l'Optimiseur Stratégique qui ne fausse pas trop sa vision de la quantité d'énergie nécessaire sur une journée. L'attention est alors portée sur la différence de performances entre les contrôleurs locaux proposés. Comme attendu, le seul parvenant à limiter les pics de puissance achetée au réseau est le Contrôleur Local prévu à cet effet, couplé à l'Optimiseur Stratégique. Par contre, celui-ci ne permet, dans les conditions considérées, d'améliorer la facture énergétique qu'à détriment égal du coût de vieillissement des batteries. Le Contrôleur Local opportuniste, quant à lui, permet d'améliorer un petit peu le coût total. Le dernier contrôleur n'utilise pas du tout les batteries et sert de référence. Finalement, les deux dernières expériences visent à appliquer l'algorithme randomisé présenté plus haut, et à déterminer le meilleur dimensionnement des batteries, puis le meilleur couple contrôleur, tarif. Les premiers résultats indiquent qu'une supercapacité avec une capacité énergétique de 60 Wh est suffisante pour absorber tous les pics de puissance depuis le réseau. De plus, une batterie de 3 kWh est suffisante pour profiter d'un tarif heures creuses / heures pleines, et n'a pas un coût d'investissement trop haut. Gardant ces choix de dimensionnement, sont ensuite comparés les contrôleurs proposés et différents tarifs de l'électricité (constant, heures creuses / heures pleines, indexé sur spot). L'expérience montre que si la revente est autorisée, le tarif indexé sur spot ainsi que le Contrôleur Local minimisant les pics couplé à l'Optimiseur Stratégique est le meilleur choix. En effet, cette combinaison permet de tirer parti de la grande variabilité du tarif. Des gains supplémentaires (uniquement liés au contrôle, pour une conception du système donnée) allant jusqu'à 0.67 € par jour peuvent être garantis dans ce cas.

Nous nous penchons ensuite sur le cas de l'usine d'armoires électriques. La gestion actuelle de l'usine de Schneider Electric est un gestion de type « pas de stock », introduite dans les années 70 au Japon. Afin de tirer parti des tarifs variables de l'énergie, il est nécessaire de changer cette politique. Nous souhaitons vérifier s'il est possible d'optimiser la facture énergétique tout en ne détériorant pas trop les coûts de retard et de stockage.

Nous avons tout d'abord dû installer des capteurs pour obtenir des données énergétiques de l'usine. Des capteurs auto-alimentés de Schneider Electric ont été installés sur les trois lignes de production, et deux passerelles ont été utilisées pour récupérer les données, les rééchantillonner, et les envoyer dans une base de données. Une première analyse visuelle des données a été effectuée, relevant que l'estampillage est l'activité qui consomme le plus d'énergie mais que la soudure semble être celle dont la consommation varie le plus en fonction des références produites. De plus, un ordre de grandeur de la consommation talon des lignes de production a pu être extrait (environ 7,5 kW pour la ligne produisant les corps des armoires). Ces données ont ensuite pu être utilisées pour construire un modèle énergétique des activités de production. Pour ce faire, les données de production et d'énergie ont du être synchronisées, puis une régression linéaire a été appliquée. Ce modèle a ensuite pu être utilisé dans le problème d'ordonnancement.

Le problème d'ordonnancement est défini comme un problème de Flow-shop généralisé avec arbre de précédence, dates d'échéances, délais minimums, et une ressource énergétique. L'objectif et de minimiser les coûts de retard, d'électricité, et de stockage. Ce problème est plus complexe que d'autres problèmes déjà démontrés NP-difficiles dans la littérature. La variante préemptive que nous considérons également est, elle aussi, NP-difficile. De nombreux travaux de recherche ont été menées concernant les Flow-shops, le RCPSP, et même les contraintes énergétiques. Toutefois, il en existe peu concernant les fonctions de coût de l'énergie linéaires par parties. Nous avons choisi d'étudier ce genre de fonctions en raison de leur utilité dans les mécanismes d'incitation sur le marché de l'énergie.

Nous proposons trois programmes mathématiques permettant de modéliser le problème, avec dans chacun une ou plusieurs approximations. Le premier modèle est orienté événements mais ne modélise que des fonctions du coût de l'énergie linéaires. Le second modèle est orienté temps, ce qui entraine une sous-optimalité dépendant du pas de temps choisi, mais prend en compte de façon exacte les fonctions de coût linéaires par parties. Le dernier modèle est également orienté temps et produit des solutions préemptées. Il modélise les précédences comme des contraintes liées au stock de produit disponible. Ce modèle donne comme solution les pas de temps pendant lesquels sont effectués les activités, mais pas leur date de début. Par conséquent, il nécessiterait une heuristique pour construire un ordonnancement, et n'est pas exploité dans les expérimentations. Enfin, une méthode de recherche locale a été implémentée, initialisée par une solution d'un problème de satisfaction de contraintes, et avec comme opérateur la formulation MILP orientée événements.

Les solutions des trois formulations MILP sur une même (petite) instance sont comparées afin de mettre en exergue leurs différences. Puis un ensemble d'instances Benchmark de la littérature est utilisé pour comparer les performances de trois des méthodes proposées, en présence, puis absence d'objectif énergétique. Sans objectif énergétique, les trois méthodes se comportent assez bien comparées aux meilleures solution de la littérature. La plus efficace étant la formulation orienté événements, et la moins efficace la formulation orienté temps. Lorsque l'objectif énergétique est pris en compte, les résultats sont plus mitigés. Pour les plus grosses instances, la solution trouvée est souvent moins bonne que lorsque l'objectif énergétique n'est pas pris en compte. Les instances améliorées ne le sont que de moins d'un pourcent (ce qui peut tout de même représenter un gain économique important). Par ailleurs, la recherche locale améliore le plus d'instances, la formulation orientée événements donne le meilleur gain moyen sur les instances avec des précédences, et la formulation orientée temps donne le meilleur gain moyen sur les instances sans précédence. Les méthodes sont ensuite comparées sur des instances construites à partir des données de l'usine de Schneider Electric étudiée. Pour l'instance la plus grosse (une semaine de production), la recherche locale donne les meilleurs résultats : environ 4.5% du coût total est économisé. Par conséquent, si nous devions proposer une méthode d'aide à la décision pour la vraie usine, nous proposerions la suivante :

1. calculer une première solution avec le problème de satisfaction de contraintes,
2. initialiser la formulation orientée événements sans coûts énergétiques avec cette première solution réalisable pour optimiser au mieux les coûts de retard,
3. lancer en parallèle les trois méthodes de résolution proposées initialisées avec la solution précédemment calculée,
4. garder les quatre solutions résultantes et les présenter à l'utilisateur.

Le dernier chapitre de ce manuscrit est consacré aux enseignements que nous avons retiré de ces trois pilotes, au regard de l'architecture logicielle. Tout d'abord, les composants nécessaires pour optimiser la facture énergétique d'un système complexe quelconque sont : des capteurs et l'architecture remontant les données, un modèle énergétique du système considéré, un module de prédiction des données nécessaires à l'optimisation, un module d'optimisation haut niveau (ordonnancement la plupart du temps), un contrôleur bas niveau pour gérer les enjeux temps-réel. L'architecture logicielle et matérielle choisie doit refléter les besoins de chacun de ces composants. En effet, les capteurs et le contrôleur local doivent être embarqués au plus près du système alors que l'optimisation et la prédiction peuvent être exposés sous forme de web-service. L'exemple de l'architecture implémentée dans le cas de l'ascenseur multi-sources est donné.

En conclusion, nous avons proposé une ou plusieurs méthodes pour résoudre chacun des trois problèmes posés initialement, ainsi que des lignes directrices générales pour résoudre un problème d'optimisation énergétique d'un système de prosumers. Les économies obtenues vont de 10.9% à 114% et d'environ 245 € par an à 109000 € par an, selon l'application.

Acknowledgements

In the first place, I would like to thank my three supervisors: Mazen Alamir, Rodolphe Giroudeau and Claude Le Pape for the time they spent with me during the past four years. Mazen always gave me relevant, albeit pointed, comments and advice. Rodolphe became one of my supervisors at a turning point of my PhD thesis and helped me to turn the corner. Claude was a continuous support and brought a lot of ideas in a wide range of activities as fondamental research, writing and development. Thanks also to Christophe Picouleau, President of my jury, and to Didier Dumur and Christian Artigues, my referees, for having spent time to review my manuscript, and to come to my PhD defense.

The second wave of thanks logically goes to my colleagues in Schneider Electric. Working sessions on the Arrowhead pilots with Dimitri Yanculovici, Jean-Louis Bergerand, Véronique Boutin, Julie Crouzet and Sylvain Marié were very instructive, and full of good mood. I will regret them. I highly appreciated coffees, flights, pastry and debug sessions (in random order) with Laurent Battini, Alfredo Samperio, Benoît Vinot and Peter Pflaum. Many thanks to Med. Omar Fannan and France Petroud for having prevented the business office to be a boring place. Thanks to the whole Analytics for Solutions team (or whatever the name it could have now), including (more of less long-time) temporary and permanent employees. I think I have been lucky to prepare my PhD thesis there. Finally, I grown up a lot during these four years, especially in diplomacy and strategic vision. Didier Pellegrin and François Bonnard helped me to understand how companies really work, with a lot of self-control and a bit of irony. I doubt I could find such as good managers in the rest of my professional life.

This work has been partially funded by the Artemis Arrowhead project under grant agreement number 332987 (Artemis/Ecsel Joint Undertaking, supported by the European Commission and French Public Authorities). Being part of the European Arrowhead project allowed me to travel a lot and to meet interesting and kind people. I would like to thank all the partners and organizers. We shared bears, wine and experience; that allowed me to become more confident when speaking English. Enjoying European cities was particularly pleasant with Maxime Louvel (from CEA), Dimitri Yanculovici (from Orange Business Services) and other partners from CEA, LCIS, Airbus and Thalès.

Finally, I have to thank people in my personal life who allowed me to begin this PhD thesis, to persevere during more than three years, and to defend it in the end. In the first category come my mother and step-father who allowed me to have a rational mind and to employ it during six long years at University; but also Swan without whom these years would have been far less funny and interesting. In the second category come my in-laws and my old school friends - Pierre, Aymeric, and Aurélie - who were my breaths of fresh air. Thank you for that and for the rest. Last but not least, I will probably defend this PhD because of my husband and son, who constantly remind me why I am doing that. I love you all.

Contents

List of Figures

List of Tables

List of Abbreviations

COP	**C**onferences **O**f (the) **P**arties
OECD	**O**rganisation (for) **E**conomic **C**o-operation (and) **D**evelopment
RTE	**R**éseau (de) **T**ransport d'**É**lectricité (electricity transportation network)
PhD	**P**hilosophiae **D**octor (doctor of philosophy)
MILP	**M**ixed **I**nteger **L**inear **P**rogramming
BAU	**B**usiness **A**s **U**sual
LC	**L**ocal **C**ontroller
SO	**S**trategic **O**ptimizer
KPI	**K**ey **P**erformance **I**ndicator

To my husband and son, without whom nothing I do would make sense.

Chapter 1

Introduction

Contents

1.1 Electricity Consumption Regulation: Norms and Market Laws

Nowadays, everybody knows about global warming and the responsibility of Human kind in it. One of the main sources of CO2 emissions (and thus global warming) is electricity production through combustion of fossil fuels, such as coal.

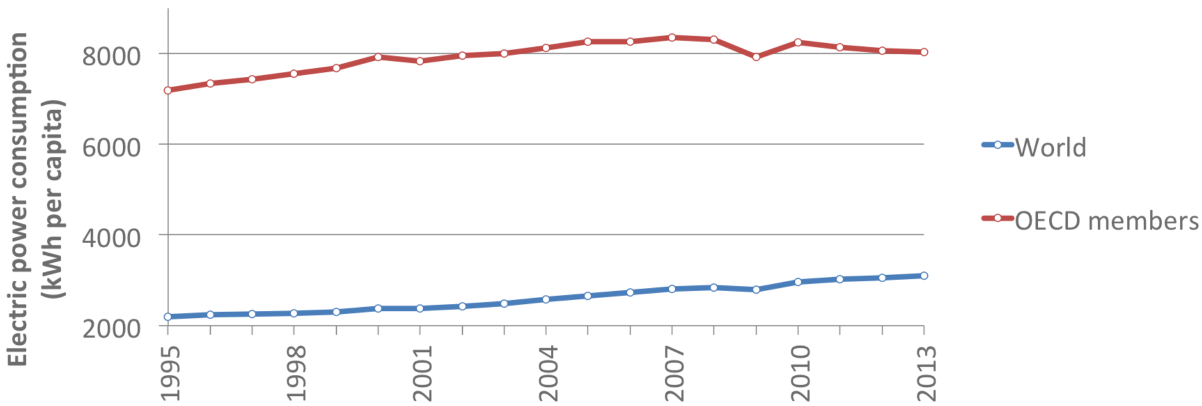


FIGURE 1.1 – Based on IEA data from IEA Statistics © OECD/IEA 2014 (www.iea.org/statistics), Licence: www.iea.org/t&c; as modified by The World Bank. (2014a). *Electric power consumption (kWh per capita)*. The World Bank

One way to act on the electricity production impact on climate change, is to reduce electricity consumption. But, despite of the successive Conferences Of the Parties (COP), the global electricity consumption still grows. As it can be seen on Figure 1.1, it has been multiplied by more than 1.4 between 1993 and 2013. Indeed, emerging countries consumption grows whilst the access to electricity democratizes. Meanwhile, developed countries have to reduce their electricity consumption, in order to limit the global consumption growth, and its impact on the environment. We can see on Figure 1.1, that the Organisation for Economic Co-operation and Development (OECD) members have succeeded to decrease their electricity consumption of almost 4% from 2007. However, this is not sufficient to reverse the global trend.

Electricity efficiency norms are put in place to try fix this issue. For example, an objective of energy efficiency for European countries, called the Directive 2006/32/EC of the European

Parliament and of the Council of 5 April 2006 on energy end-use efficiency and energy services and repealing Council Directive 93/76/EEC, was enforced in 2006. Similar initiatives are taken all around the world either by subsidizing actors who act on their electricity efficiency, or by taxing actors who do not. Thus, companies, and industries in particular, are more and more constrained by these norms and taxes.

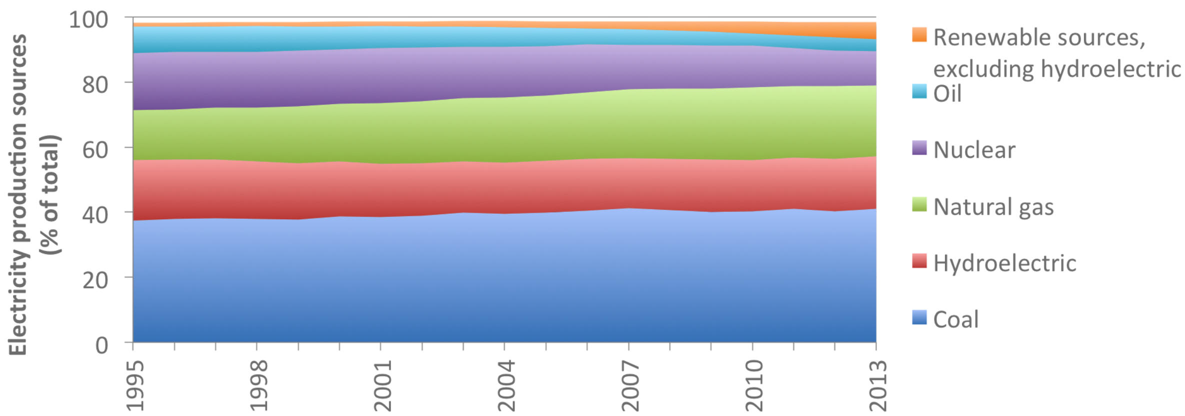


FIGURE 1.2 – Based on IEA data from IEA Statistics © OECD/IEA 2014 (www.iea.org/statistics), Licence: www.iea.org/t&c; as modified by The World Bank. (2014b). *Electricity production sources*. The World Bank

A second way to reduce the electricity production impact on climate change, is to use energy sources that generate less CO2 and methane emissions. As said before, coal is the main source of CO2 emissions, but we can see on Figure 1.2 that its usage is slowly increasing since 1993. On the other hand, the usage of renewable electricity sources has also increased during those past few years. Renewable sources induce far less CO2 emissions, but they are intermittent power sources. Thus, electricity transmission system operators have to cope with the volatility of those power sources. Indeed, countries electricity networks are managed by companies, that are in charge of maintaining the network equilibrium. For example, RTE, the French electricity transmission system operator explains its role in the following way.

> Our roles and missions go way beyond the implicit meaning of the term "power transmission". Because electricity can only be stored in limited amounts, it needs to be used as soon as it is generated. At the heart of the power system, we are responsible for keeping the balance between supply and demand. By ensuring that we have the ability to fulfill this role on a daily basis, we provide our customers with economical, reliable and clean access to power supply. (RTE)

In order to balance the French power system (in terms of frequency and voltage), RTE relies on three energy reserves. Each of them has an increasing response time: from fifteen seconds to half an hour. All producers connected to the transmission system have to keep a portion of their power unused, in order to contribute to the primary reserve. Big producers have to contribute also to the secondary reserve. The tertiary reserve, called the balancing mechanism, is used only when the secondary one is not sufficient. Participating in this tertiary reserve is on a voluntary basis, assuming that at least 10MW are proposed. Contracts are concluded with voluntary producers and remunerated monthly. They have to provide everyday power they have promised. Consumers can also be solicited whenever it is needed. The most profitable offers are kept by RTE to balance the network.

The above mechanisms are the example of the French network of energy, but everywhere in the world the demand-response market emerges. Laws regulating this new market emerge as well, like the European mechanism of internal market of energy: Directive 2009/72/EC of the European Parliament and of the Council of 13 July 2009 concerning common rules for the internal market in electricity and repealing Directive 2003/54/EC. In Albadi and El-Saadany, 2008, the possible kinds of demand-response programs, that balance a power network by playing with incentive payments, are presented. They are ways to avoid power peeks while saving money at the transmission system operator side. But for customers to participate in the demand-response market, they have 1) to be big customers (at least 10MW in France), and 2) to be able to forecast and shift their consumption.

The keen interest of smaller customers in the demand-response market induces the emergence of new actors: the aggregators. Aggregators incite small customers to shift their consumption using the same incentive mechanisms as the transmission system operators. Then, they aggregate multiple customers offers into a bigger block of energy that they can sell on the demand-response market. For example, the Energy Pool company is an aggregator that declares having a primary reserve and 2500 MW of flexible load.

In that context, customers are interested in how to reduce their power consumption peaks. The optimization algorithms needed have to take into account a variable electricity tariff, but also the other customer's constraints and objectives. In practice, new optimization problems, or even more complex versions of previous problems, have to be studied. The goal of this PhD thesis is to study the different ways to solve these problems and to compare the different methods in term of efficiency and genericity.

1.2 Research Work Context

This PhD thesis is realized within Schneider Electric, the global specialist in energy management and automation, operating in over 100 countries with around 160,000 employees worldwide. Schneider Electric develops connected technologies and solutions to manage energy and process in ways that are safe, reliable, efficient and sustainable.

This PhD work was accomplished as part as the Analytics for Solutions team, under the supervision of Claude Le Pape in Schneider Electric. Claude Le Pape, PhD in Computer Science of the University Paris XI, is an expert of planning and scheduling. He is co-author of a hundred of publications including a reference book and twenty papers in books and international journals. Analytics for Solutions is a corporate Research and Development team, composed of around twenty researchers and developers focused on the development of new techniques for energy efficiency.

Moreover, this work was part of the Arrowhead European project. This ARTEMIS project (2013-2017) federated several dozens of industrial and academic partners. It aimed to enable collaborative automation by networked embedded devices. The application domains considered were (among others) the manufacturing industry, building management, electric mobility, electricity production, and the effective implementation of the virtual energy market, that is the interaction hub of these different application areas. Schneider Electric supervised the work package one, dedicated to industrial applications. The main goal for Schneider Electric was to realize three pilot applications with partners in the domains of: water distribution, manufacturing, and elevators. These pilots supposed the installation of the required infrastructure (sensors, actuators, communication modules, etc.) and the development of optimization tools allowing to reduce cost, energetic consumption, or environmental impact of the concerned activities.

Three optimization problems taken from those applications are proposed by Schneider Electric to motivate and evaluate this PhD work. Each problem is declined in two versions: a deterministic version, considering that the different parameters value are known in advance; and a stochastic version. The possibility to use or not the same models and methods for the 3×2

problems considered is one of the main contributions of this work. In order to support fundamental research in optimization and control needed for this thesis, two research laboratories collaborate with Schneider Electric: the Gipsa-lab and the LIRMM.

Gipsa-lab is a mixed research unit in Grenoble, internationally recognized for the research achieved in Automatic Control, Signal and Images processing, Speech and Cognition. For the Control Department, the main challenge is to suggest tools for analysis, control and diagnostics of dynamic systems in order to ensure an efficient, safe and reliable operation. Inside this department, this PhD thesis is part of the SYSCO team (non linear systems and complexity), composed of around forty permanent and non-permanent people. In particular, this PhD thesis is supervised by Mazen Alamir, CNRS Research Director. Mazen Alamir has a doctoral degree in control from Grenoble-INP, an engineering degree in avionics from ENSICA-Toulouse, and an engineering degree in mechanical engineering from Grenoble-INP. He is a specialist of non-linear model predictive control under constraints, and particularly of its application, in real-time, on fast dynamic systems. He is author of a hundred and fifty publications, including fifty in international journals, two monographs and a dozen of patents from industrial collaborations.

The LIRMM (laboratory of computer science, robotics and microelectronic of Montpellier) is a mixed research unit, with a wide range of research domains from information technologies to systems passing by human interfaces. The MAORE team (algorithms and methods for scheduling and networks) is composed of around fifteen permanent and non-permanent members. These members use combinatorial optimization, graph theory, mathematical programming, and constraint programming to solve discrete optimization problems exactly or approximately. In this team, Rodolphe Giroudeau, Assistant Professor accredited to direct research, supervises this PhD thesis. He obtained his doctoral degree in Computer Science from University of Evry in 2000, and his HDR (authorization to supervise research) in 2012 from University of Montpellier. He is specialized in approximability and complexity of scheduling problems and is author of around fifty publications including twenty one in international journals.

Now let us introduce more precisely the research work conducted during those three years.

1.3 Research Work Introduction

Let us first introduce briefly the optimization problems proposed by Schneider Electric to lead this thesis research work. The corresponding systems are represented on Figure 1.3. The first use-case studied is the optimization of the energy bill of a multisource elevator, under a maximum power peak constraint. A multisource elevator is an elevator plugged to multiple sources of energy. The goal is to decide how much power has to be taken from each source over the time horizon. The second problem considered is the energy optimization of a manufacturing plant. There are production activities to schedule and the question is: can we shift them in time to save on the electricity bill, without exceeding their due date? The third and last problem concerns the computation of pumping plans for a drinking water network. There is a variable water demand and the goal is to supply water towers by pumping water efficiently, while minimizing the electricity cost.

Several ways to address the problems stated can be considered. A first one consists in forgetting the physical details of the industrial processes considered, and to reduce to a scheduling problem under resource constraints (including energy). The electricity is a renewable resource and its usage cost is variable in time, and more or less known in advance.

Scheduling problems under resource constraints are known to be particularly difficult, among the most difficult NP-hard problems (cf Garey and D. S. Johnson, 1979). Their resolution can require the combination of multiple resolution methods, like Mixed Integer Linear

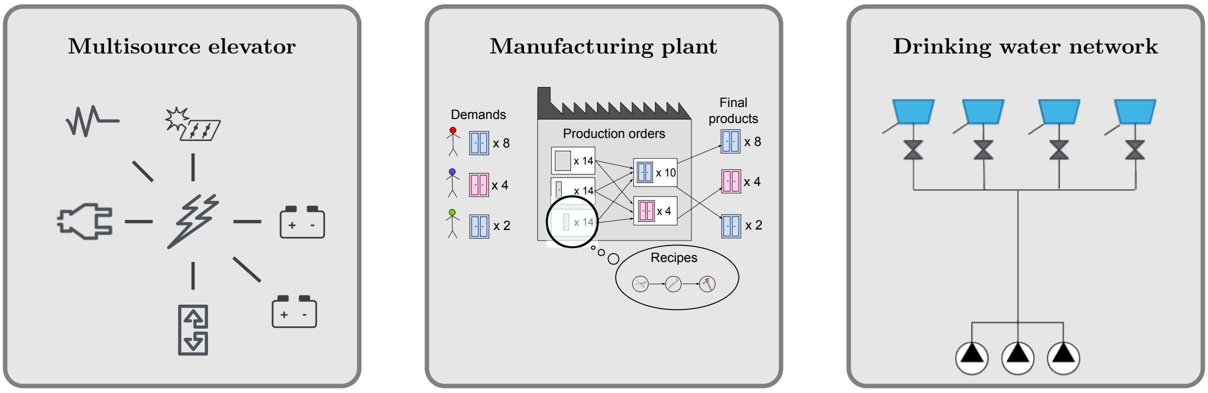


FIGURE 1.3 – The three optimization problems studied.

Programming, Constraint Programming, Dynamic Programming, Local Search with or without meta-heuristics, or even specific operational research algorithms (cf Brucker, 2007; Baptiste, Le Pape, and Nuijten, 2001; Danna and Perron, 2003; Danna, Rothberg, and Le Pape, 2003; Maravelias and Grossmann, 2004).

Energy and CO2 emission concerns add a new dimension to those problems, in term of optimization criteria and constraints. Power resources (with a capacity expressed in Watt) and energy resources (with a capacity expressed in Watt-hour), are available in limited quantities and their cost is defined on time periods. The notion of energy reasoning, introduced by Erschler, Lopez, and Thuriot, 1991 is often used to enforce the treatment of classical resource constraints, and could be useful in this context. More recently, energetic considerations were taken into account for: the management of co-generation systems and the scheduling of furnaces in a still mill (cf Haït and Artigues, 2011); modeling chemical plants under energetic constraints (cf Castro, Harjunkoski, and Grossmann, 2011).

The multicriteria dimension of the scheduling problems is obviously strengthened, due to the impossibility to combine multiple criteria (like tardiness, production quality, or CO2 emission) into a single economical objective. The generation of multiple solutions (ideally Pareto-optimal), and the capacity to let the final user making his own trade-off between different criteria, are important elements of acceptance of such an optimization tool.

In all the cases mentioned above, classical scheduling formulations were expanded by adding energy-related constraints and cost factors. In some cases, it could be more relevant to decompose the global problem and to exploit existing optimization systems, developed without any energy consideration, but whose the operators of the considered process are used to. An open question is to know how much is it possible to decouple the already solved optimization problem from the energy concern.

A related question is to determine whether optimization should be centralized or decentralized. There are two good reasons to avoid defining an optimization problem centralized and detailed in the same time: resolution efficiency when the global problem is too big and too complex; and the robustness of the system. Indeed, a decentralized system in which sub-problems are solved locally is intrinsically more robust to disturbances than a centralized system. The best way to proceed in often to link optimization problems defined at different levels. For example, an aggregated model can be used at the global system level to define the energy consumption (or CO2 emission) limits for different sub-systems. Corresponding sub-problems can then be solved under these constraints and the results obtained be used by the aggregated model to modify the limits and iterate. Such an approach has been developed for the energy consumption of a building and its “zones” in the context of a PhD thesis in the Gipsa-lab and Schneider Electric (cf Lamoudi, Alamir, and Béguery, 2011).

Reduction to a scheduling problem could be difficult or not sufficient when considered processes are not discrete (as this is the case for the elevator and the water network). Several solutions can be considered in such a case: coupling an optimization module approximating physical details of the process with a simulator (cf for example Veronique Boutin and Bergerand, 2013 for a water network); or defining the global optimization problem as a predictive control problem, solved repeatedly at regular time intervals (cf for example Lamoudi, Alamir, and Béguery, 2011). The interest of such an approach, based on the explicit representation of the considered process dynamic, is double: proposing easy-to-use set-points for (semi-)continuous industrial processes; and increasing the robustness of a decision regarding the approximations that have been made in the model, and the stochastic properties of the system studied.

Taking into account those stochastic properties becomes particularly crucial when real-time strategies for energy consumption adjustments are considered. In the current model, an energy provider or an aggregator asks (a few hours in advance) consumers to erase all (or a part of) their consumption in exchange of an amount of money known in advance. When the production scheduling of a plant, or the pumping plan of a water network is set, we do not know yet if there is an interest in erasing consumption at a given time. Concepts of robust and stochastic optimization (cf for example Benoist et al., 2001; Ben-Tal, El Ghaoui, and Nemirovski, 2009) can be useful in that context. However, let us note that, regarding the potential impact of an erasure decision and the lack of knowledge of the probability of erasure being useful, it is possible that a simple management of possible scenarios or a tree of schedules answering some constraints would be sufficient (cf for example Drummond, Swanson, and Bresina, 1994).

Based on those observations, we proposed different solving methods for each of the three problems. Some work has also been done for handling the sensors data and building demonstrators for each of the three Arrowhead generations.

Our work on solving the optimization problem of the multisource elevator is related in Chapter 2. It was conducted jointly with Mazen Alamir, Rodolphe Giroudeau and Claude Le Pape, my supervisors, but also with Véronique Boutin (Schneider Electric). It is partially based on the internship of Laurent Billet in Schneider Electric before the beginning of this thesis. Moreover, a demonstrator was built for the first two Arrowhead generations, coupling data and an elevator simulation coming from Sodimas, our optimization modules and a graphical interface allowing user interactions (cf Véronique Boutin et al., 2014). The data were transmitted between software components in different places through the LINC middleware developed in CEA (cf Louvel and Pacull, 2014). This demonstrator was designed conjointly by Véronique Boutin and Julie Crouzet; a glimpse of the user interface is given on Figure 1.4. Software components were developed by Maxime Louvel (CEA) and I.

For the manufacturing plant use-case, we had to put energy sensors in place, to handle production and energy data, to build energy models and to compute energy-aware production plans. All these aspects are discussed in Chapter 3. Research work on the scheduling problem were conducted jointly with two of my supervisors (Rodolphe Giroudeau and Claude Le Pape) and two Master students (Grigori German and Mustapha Haouassi). Data mining and energy modeling were studied with the help of two of my colleagues in Schneider Electric: Jean-Louis Bergerand and Dimitri Yanculovici, a Master student: Pierre-Alexis Berseneff, and Claude Le Pape. Finally, a demonstrator has been built by Dimitri Yanculovici for the two last Arrowhead generations. A view of the graphical interface developed is shown on Figure 1.5.

The third and last use-case, the drinking water network, has not been fully investigated during this thesis. We focused on the two other subjects and the main results we have on this subject come from two internships. Moreover, a former Schneider Electric trainee, Gratien Bonvin, is currently doing a PhD thesis on the same subject. Thus we briefly recall the first results below and point out Gratien Bonvin's publications for the interested reader, but there is no chapter in this PhD thesis dedicated to this subject. A first quadratic optimization model was proposed to compute energy-aware pumping plans. Optimized plans are more stable

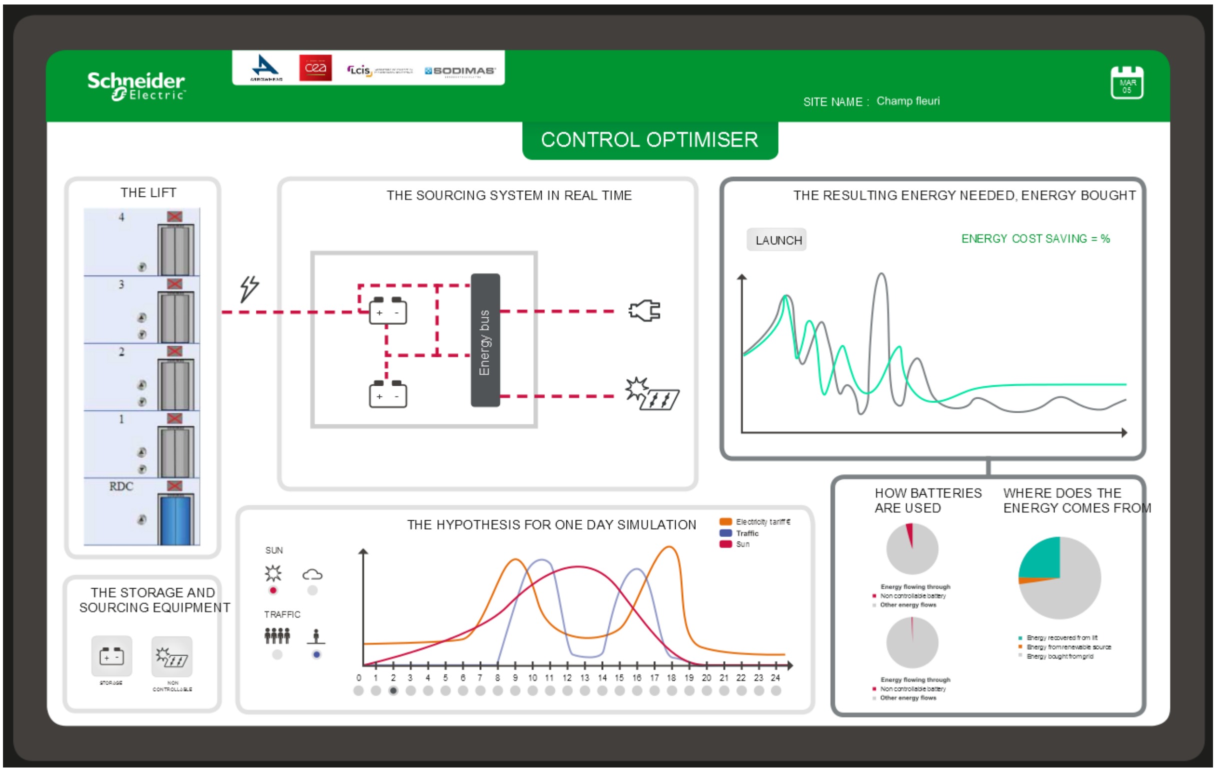


FIGURE 1.4 – Multisource Elevator demonstrator.

and less costly than Business As Usual (BAU) ones, as they fully exploit pumps efficiency and off-peak hours. These pumping plans are briefly discussed in Chapter 4. This approach has been improved by Gratien Bonvin, in the context of his PhD thesis (cf Bonvin, Samperio, et al., 2015; Bonvin, Demassey, et al., 2016). A second intern, Amaury Bonnemaison, developed a simulation model of the studied water network. Amaury Bonnemaison, Alfredo Samperio, and I, used this detailed model to evaluate the feasibility and accuracy of the pumping plans. First results show that the electricity bill can be improved of 15% by using optimized pumping plans.

Now, let us dig into the multisource elevator use-case, and the optimization methods we propose to handle it.

rrowhead Manufacturing (Generation 3)

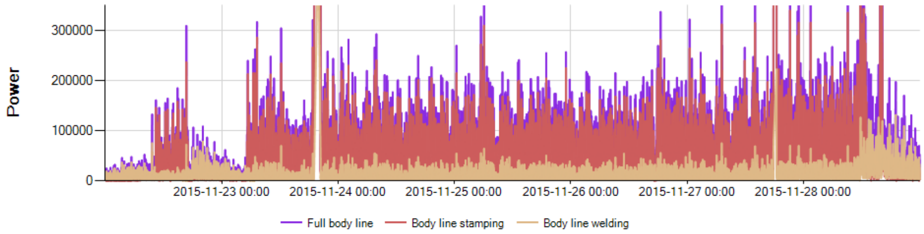
From energy monitoring to manufacturing process optimization... powered by the Arrowhead framework

1) Get data from the plant :

Demo version :
- Generation 2
- Generation 3

Data source :
- plant (4 machines)
- plant (light)
- plant (heavy)
- . plant
- Arrowhead demo EN_NCOS_04a
- Arrowhead demo EN_NCGS_31a
- Arrowhead demo EN_NCOS_32a
- Arrowhead demo EN_NCOS_02a

Data come from a manufacturing plant in France. This plant produces electrical cabinet.4 machines have been instrumented with electricity sensors. These sensors send data through radio (ZigBee)which goes through two gateways dedicated to data filtering, sampling and formating (using Linc middleware from CEA)Then data are sent to a database and presented by a visualisation tool.



2) Plant data source synchronization :

Available synchronization services among Arrowhead ecosystem (arces.unibo.it):

Service inputs :

Original energy consumption C:\Arrowhead\SVN internal\manufacturing\dev\data\G3 demonstration\Sarel_W48\0 Initial data\SAREL_Energy_BodyLine_2015_48.csv

Original production schedule C:\Arrowhead\SVN internal\manufacturing\dev\data\G3 demonstration\Sarel_W48\0 Initial data\SAREL_Schedule_2015_48.json

Call web service *http://RM-machine.cloudapp.net:8080/RAWS/process/manufacturingEnergyProfiling/1.2/synchronizeProductionAndEnergyData*

Service output :

startDatetime	endDatetime	duration	energyBodyLine_Wh	energyBodyStamping_Wh	energyBodyWelding_Wh	energyBodyLine_Wh_pow
2015-11-22T00:00:15.000+0100	2015-11-23T05:45:26.000+0100	107111,0	347906,90	207358,05	140548,85	11693,15
2015-11-23T05:45:26.000+0100	2015-11-23T05:45:37.000+0100	11,0	162,19	135,2	26,99	53081,35
2015-11-23T05:45:37.000+0100	2015-11-23T06:27:05.000+0100	2488,0	23011,13	19040,34	3970,79	33295,85
2015-11-23T06:27:05.000+0100	2015-11-23T06:27:32.000+0100	27,0	284,90	249,03	35,86	37986,00
2015-11-23T06:27:32.000+0100	2015-11-23T06:27:51.000+0100	19,0	247,26	207,60	39,66	46848,32

3) Characterize reference energy consumption :

Available charcterization services among Arrowhead ecosystem (arces.unibo.it):

Original energy consumption C:\Arrowhead\SVN internal\manufacturing\dev\data\G3 demonstration\Sarel_W48\0 Initial data\SAREL_Energy_BodyLine_2015_48.csv

Original production schedule C:\Arrowhead\SVN internal\manufacturing\dev\data\G3 demonstration\Sarel_W48\0 Initial data\SAREL_Schedule_2015_48.json

Synchronized production and energy *See above data source synchronization result*

Call web service *http://RM-machine.cloudapp.net:8080/RAWS/process/manufacturingEnergyProfiling/1.2/characterizeReferenceEnergyConsumption*

Result overview :

name	value	error	tStatistic	pValue
energyPerProductAENN5120807071M0	100,07	41,74	2,40	0,02
energyPerProductAENNA07058M0	213,87	75,56	2,83	0,00
energyPerProductAENNA06976M0	60,15	60,79	0,99	0,32
energyPerProductAW15110330JCRA00005M0	152,82	189,22	0,81	0,42
energyPerProductAW15100596JCRA00002M0	76,13	105,63	0,72	0,47
energyPerProductAW15110386JCRA00003M0	21,72	46,66	0,47	0,64

FIGURE 1.5 – Manufacturing plant demonstrator.

Chapter 2

Multisource elevator

Contents

The first use-case studied during this thesis concerns elevators connected to multiple sources of energy. In this document, they are called "*multisource elevators*". In Section 2.1, we explain why multisource elevators make sense. Then, Section 2.2 provides a formal description of the sourcing problem. In Section 2.3, a brief literature review is discussed. Section 2.4 introduces our control method. Performance of this control method, depending on uncertainties, can be guaranteed due to the method explained in Section 2.5. Finally, Section 2.6 presents experimental results and Section 2.7 concludes this chapter.

2.1 Multisource Elevators: Why and How?

In buildings intended to welcome public, several pieces of equipment have to keep a power supply in case of grid failure (i.e.: emergency lighting, elevators dedicated to disabled people evacuation, etc). Power supply of emergency equipment is regulated by law. For example, in France, Decree of 25 June 1980 *approving the general dispositions of the safety regulation against the risks of fire and panic in establishments open to the public* states that batteries have to be used, and can be coupled with one (or several) gensets.

In this context, batteries have to be installed in buildings and can be exploited to save a part of the electricity bill. Indeed, they could be connected to devices that produce energy in the building to be (re)filled and they could supply energy when the electricity is expensive. To make this work, the system needs electronic components in order to connect all devices and ensure that the voltage of each device is respected. Moreover, a relevant sizing of batteries has to be performed in order to enable their use for savings without impacting their usage for safety. Also, bigger batteries could be more expensive but could avoid the need of gensets.

In order to study the feasibility of using batteries in the way described above, we chose a use case where an elevator is connected to batteries, and other sources (or sinks) of energy. The "energy hub" also guarantees that the elevator can take disabled people out of the building in case of fire, during power outage. The considered system is presented on Figure 2.1. We

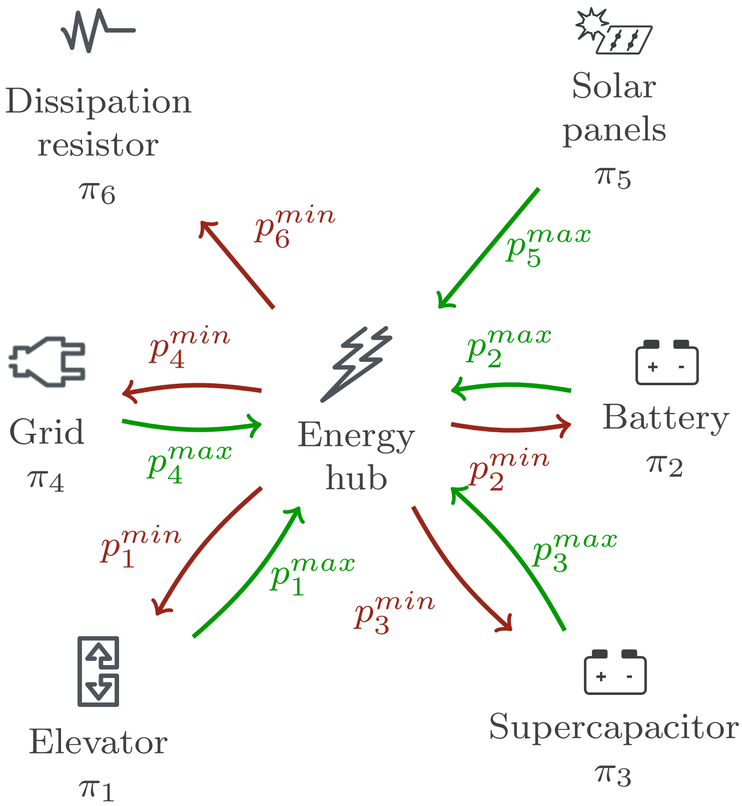


FIGURE 2.1 – The multisource elevator use case.

consider six devices connected to the same energy hub, including the elevator. Each device can consume and/or produce energy, depending on its physical capabilities. The energy hub, and the devices connected to it, can be seen as a star graph, rooted in the energy hub. An arc from a device to the energy hub means that the device can produce power up to p^{max} whereas an arc from the energy hub to a device means the device can consume power up to $-p^{min}$ (with p^{min} negative).

Each of the devices connected to the energy hub is described below:

π_1 **the elevator** An elevator can consume power, but also produce power by recovering energy from braking. Indeed, most of the elevators work with a counterweight, and depending on the direction of the elevator, and on the ratio between the cabin mass and the counterweight mass, the motor has to be started or the brakes have to be applied. Moreover, applying brakes generates heat that can be converted to energy. As we can see on Figure 2.2, when the cabin is heaviest than the counterweight and the elevator goes down, brakes have to be applied and energy can be recovered. At the opposite, when the cabin is heaviest than the counterweight and the elevator goes up, the motor has to be used and energy is consumed.

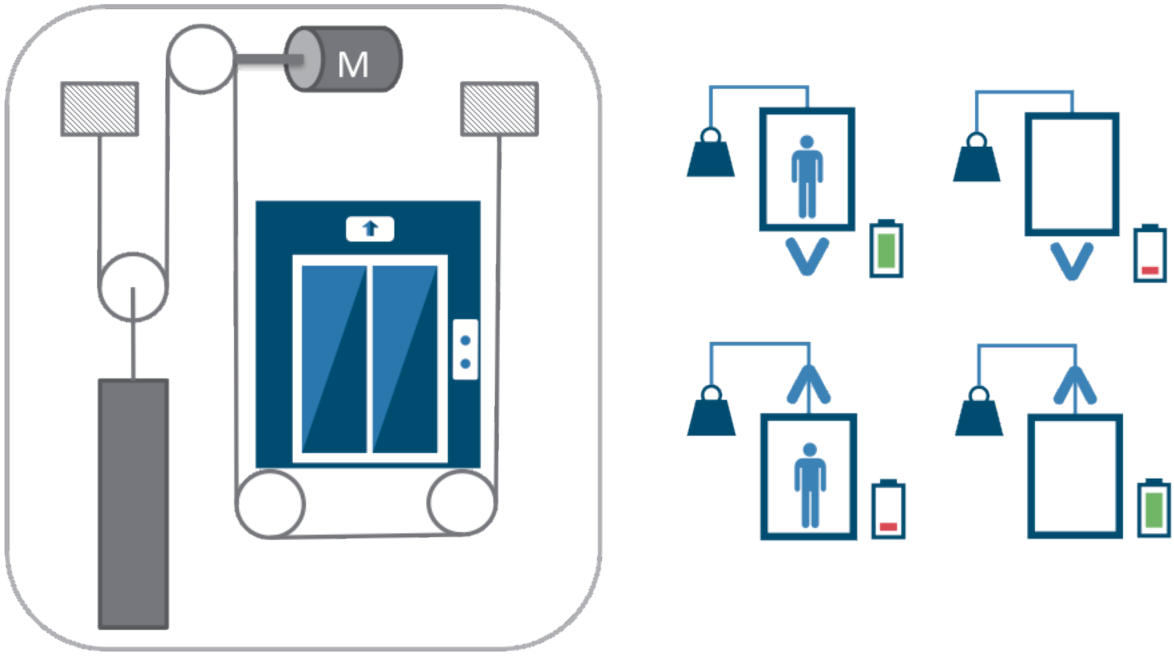


FIGURE 2.2 – When energy can be recovered from an elevator.

π_2 **the battery** A battery is plugged to the energy hub to give autonomy to the elevator in case of grid failure and to save a part of the energy bill, taking advantage of a non-constant energy tariff. The battery should be big enough to achieve both goals. But the power capabilities of the battery are in all the cases insufficient to activate the elevator.

π_3 **the supercapacitor** Supercapacitors have great power capabilities but a small energy capacity. Thus, the supercapacitor ensures that the elevator can move at any time, in case of power outage. Indeed, the battery cannot give enough power to allow the elevator moving, while the supercapacitor can. On the other hand, the supercapacitor does not contain enough energy to allow multiple elevator travels, while the battery does. This is why a battery and a supercapacitor are both plugged into the energy hub. Moreover, the supercapacitor is more expensive than the battery, while usage age both of them.

π_4 **the grid** The grid is able to provide power as well as consume power when re-selling energy is allowed.

π_5 **the solar panels** We consider that solar panels are installed on the roof top and partially dedicated to supplying the elevator.

π_6 **the dissipation resistor** Finally, a dissipation resistor has to be added to the system in order to convert electricity into heat when there is too much energy.

Given such a multisource system, one has to decide from which source(s) electricity shall be taken (and to which sink energy shall be given) over time. In the following section, the "*sourcing problem*" is formalized, as well as its application to the multisource elevator.

2.2 The Sourcing Problem

We call "*prosumers*", entities that consume and/or produce energy. An energy hub allows each prosumer to consume power produced by all other prosumers at the same time.

Definition 1. *Let $\mathcal{P}$ be a set of n_p prosumers, all connected to the same energy hub h.*

We differentiate three kinds of prosumers: the set $\mathbb{S}$ of storage units, the set $\mathbb{P}$ of controllable prosumers and the set $\mathbb{E}$ of the others. Thus, these sets form a partition of the whole set of prosumers.

Definition 2. *Let $\mathbb{S} = \{\pi_1^{\mathbb{S}}, \ldots, \pi_{n_{\mathbb{S}}}^{\mathbb{S}}\}$ be the set of storage units, $\mathbb{P} = \{\pi_1^{\mathbb{P}}, \ldots, \pi_{n_{\mathbb{P}}}^{\mathbb{P}}\}$ be the set of controllable prosumers or consumers, and $\mathbb{E} = \{\pi_1^{\mathbb{E}}, \ldots, \pi_{n_{\mathbb{E}}}^{\mathbb{E}}\}$ be the set of the other prosumers. Then, $\mathcal{P} = \mathbb{S} \cup \mathbb{P} \cup \mathbb{E}$.*

Each prosumer can consume and/or produce power, depending on its physical capabilities.

Definition 3. *For a given prosumer $\pi_i \in \mathcal{P}$, let $p_i^{min} \leq 0$ (resp. $p_i^{max} \geq 0$) be the minimum (resp. maximum) instantaneous power of π_i. Then, at a given time t, let $p_i^{min} \leq p_i[t] \leq p_i^{max}$ be the instantaneous power produced by π_i if $p_i[t]$ is positive, or consumed by π_i if $p_i[t]$ is negative*[1].

A system composed of an energy hub and its prosumers can be represented by a star oriented graph.

Definition 4. *Let $\mathcal{G} = (\mathcal{P} \cup h, \mathcal{A})$ be a star oriented graph rooted in h where $\mathcal{P} \cup h$ are the nodes of the graph and $\mathcal{A}$ are the weighted arcs. There is an arc (π_i, h) if $p_i^{max} > 0$ and the weight of the arc is p_i^{max}. In the same way, there is an arc (h, π_i) if $p_i^{min} < 0$ and the weight of the arc is p_i^{min}.*

We suppose that time can be sampled in a regular, uniform way [2].

Definition 5. *Let $\tau \in \mathbb{R}$ be the sampling period (expressed in hours), and $H \in \mathbb{N}$ be the number of periods considered. Then time-steps are expressed in the following way: $t_l = t_{l-1} + \tau = l \times \tau, \forall l \in \{0, \ldots, H\}$.*

Finally, each controllable prosumer has a set of energy cost functions that give the price associated to an energy consumption or production of this prosumer. There is a cost function per time bucket; a time bucket being a time period during which the tariff does not vary. For example, a peak/off-peak tariff (0h - 5h - 19h - 0h) can be represented by three linear cost functions of the power used.

Definition 6. *$\forall \pi_i \in \mathbb{P}$, let $\mathcal{B}$ be the set of time buckets and $\{\forall b \in \mathcal{B}, cost_{\pi_i}^b : \mathbb{R} \times [p^{min}, p_i^{max}] \rightarrow \mathbb{R}\}$ be the energy cost functions associated to π_i. We will note $b(t)$ the time bucket containing time t. For example, $cost_{\pi_i}^{b(t)}(\tau, p)$ is the cost incurred when the power is p over a period τ beginning at time t, during the time bucket b. $cost_{\pi_i}^{b(t)}(\tau, p)$ is positive if p is negative and negative if p is positive.*

[1]Power is expressed in Watts and energy in Watt hours.

[2]In practice, the models we have designed and used would be easy to generalize to time sampled in a given non uniform way, e.g.: aggregating more time during the night when the elevator is seldom used

Then, we can define the sourcing problem:

Instance: a set of prosumers $\mathcal{P} = \mathbb{P} \cup \mathbb{S} \cup \mathbb{E}$,

a graph $\mathcal{G} = (\mathcal{P} \cup h, \mathcal{A})$,

a period $\tau \in \mathbb{R}$,

a time horizon $H \in \mathbb{N}$,

a set of cost functions: $\bigcup_{\pi_i \in \mathbb{P}} \{\forall b \in \mathcal{B}, \mathrm{cost}^b_{\pi_i}\}$

Solution: S, a $n_p \times H$ matrix of $p_i[t]$

Question 1: given $p^{max}_{hub} \in \mathbb{R}$ the allowed quantity of power to be taken or given outside of the energy hub, does a matrix S exist such that:

$$\forall \ell \in \{0, \dots, H-1\},$$

$$-p^{max}_{hub} \leq \sum_{i=1}^{n_p} p_i[t_\ell] \leq p^{max}_{hub} \quad (2.1)$$

Can the energy hub be autonomous?

Question 2: given $\{\forall \pi_i \in \mathbb{P}, p^{max}_{\mathbb{P}_i} \in \mathbb{R}\}$ the allowed power peak of each controllable prosumer π_i, does a matrix S exist such that:

$$\forall \pi_i \in \mathbb{P},$$

$$\max_{\ell \in \{0,\dots,H-1\}} p_i[t_\ell] \leq p^{max}_{\mathbb{P}_i} \quad (2.2)$$

Can power peaks purchased from controllable prosumers be upper bounded?

Question 3: given $cost^{max}_{hub} \in \mathbb{R}$ the allowed energy bill, does a matrix S exist such that:

$$\sum_{\ell=0}^{H-1} \sum_{\pi_i \in \mathbb{P}} \mathrm{cost}^{b(t_\ell)}_{\pi_i}(p_i[t_\ell] \times \tau) \leq cost^{max}_{hub} \quad (2.3)$$

Can the electricity bill be upper bounded?

The multisource elevator is an application of the sourcing problem where $\mathbb{P} = \{\pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6\}$ and the partition of $\mathcal{P}$ is the following: $\mathbb{E} = \{\pi_1, \pi_5\}$, $\mathbb{P} = \{\pi_4, \pi_6\}$, $\mathbb{S} = \{\pi_2, \pi_3\}$. An illustration of the partition of prosumers $\mathcal{P}$ is given on Figure 2.3.

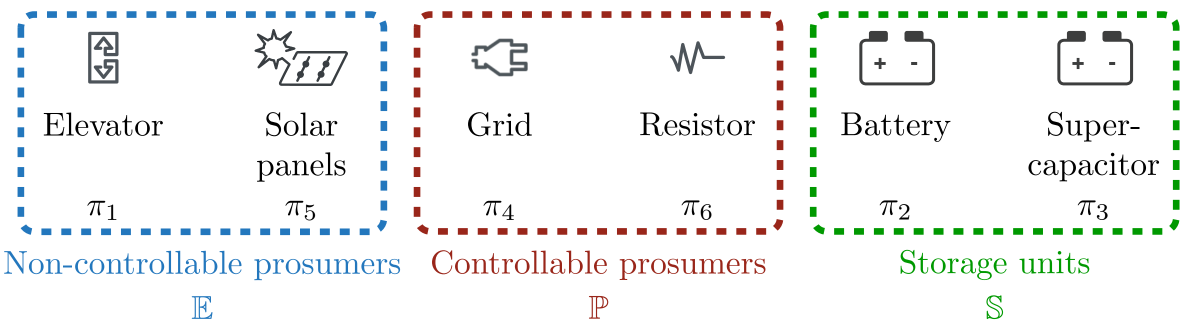


FIGURE 2.3 – Prosumers of the elevator sourcing problem.

As the multisource elevator problem was suggested by a real-world application, the three objectives have to be achieved in that case. The energy hub autonomy becomes a constraint and the question becomes: given data $p_{\mathbb{P}}^{max} \in \mathbb{R}$ and $cost_{hub}^{max} \in \mathbb{R}$, can power peaks from and to the grid and the energy bill remain bounded, respectively by $p_{\mathbb{P}}^{max}$ and by $cost_{hub}^{max}$?

2.3 Energy Sourcing for Elevators: State of the Art

In order to regulate the energy consumption of an elevator, the energy consumption itself can be decreased, or energy can be recovered and stored into storage units to be reused later. In (Sachs, 2005), smart ways to choose elevator physical components (motor, drive, etc) are summarized, as well as appropriate sizing methods.

On the other hand, elevators coupled with supercapacitors have been studied by many researchers and companies. Some of them investigate how to commute softly between the grid and a supercapacitor, like in (Paire et al., 2010; Tominaga et al., 2002). In (Paire et al., 2010), a physical multisource system is designed to power an elevator. Rules are used to charge or discharge batteries depending on whether the electrical current is below or above a given reference. Likewise, (Tominaga et al., 2002) presents three rule-based methods to control a battery coupled with an elevator. This method takes into account peak/off-peak tariffs and reduces energy consumption cost by storing energy recovered from the elevator. These methods allow to control the system in real-time, but cannot optimally take into account external considerations such as the electricity tariff or the battery state of health. Therefore, these control methods may not always be efficient regarding economical objectives.

Works have also been conducted on how to take into account future energy consumption to minimize the energy bill, using storage units. In (Bilbao and Barrade, 2012), a General Energy and Statistical Description (GESD) of the possible missions of an elevator is proposed, as well as a dynamic programming-based energy manager. The energy manager is inspired by stock management theory and minimizes the sum of energy (i) absorbed from the grid, (ii) dissipated in the braking resistor and (iii) not provided to the elevator. The optimization is done offline. This method allows to find an optimal solution regarding economical objectives from probabilities of consumption. But elevator usage is unpredictable by nature and what is to be done when the strategy is unfeasible is not investigated.

We study the sourcing problem introduced in Section 2.2, where several energy sources are supposed to be available to be used by the elevator and the problem is to choose between them over time. Our resolution method is composed of two coupled controllers: one for handling real-time issues and a second to compute a long-term sourcing strategy. In this chapter, advantages and drawbacks of each of these controllers are investigated, as well as their interactions. In addition, a Certification Framework (presented in Section 2.5) is used to guarantee customer savings on the energy bill and a limit on the power peak consumed from the grid.

2.4 Proposed Solution: Coupled Strategic and Tactical Optimization

To handle real-time control, we propose a centralized rule-based algorithm for the energy hub. Note that in practice, we could also overlay a controller already implemented. We call these controllers "*Local Controllers*", and we abbreviate LC. They have to be embedded and highly reactive, thus they cannot compute the best sourcing strategy on a long time-frame. Such a sourcing strategy is computed with a "*Strategic Optimizer*" (abbreviated SO), which sends next strategic instructions to LC regularly.

Definition 7. *Let us call the plan computed by SO a* strategy, *and the set-point computed by LC a* tactic.

Let us note that previous versions of that control method were published in (Desdouits, Alamir, Veronique Boutin, et al., 2015; Desdouits, Alamir, Giroudeau, and Le Pape, in press). Hereafter, we start by presenting data that feed components and interactions between them.

2.4.1 Data and Interactions

LC has to be highly reactive. For the multisource elevator application, we estimate a relevant time-step is one second, iterated every second. On the other hand, SO has to consider sufficiently long time-steps to get relevant forecasts, and a sufficiently long horizon to take into account energy price variations. Thus, a fifteen minutes period with a 24h horizon is relevant for the multisource elevator, and the problem is re-solved every hour.

The dynamic of the interactions between LC and SO is illustrated on Figure 2.4. In a real

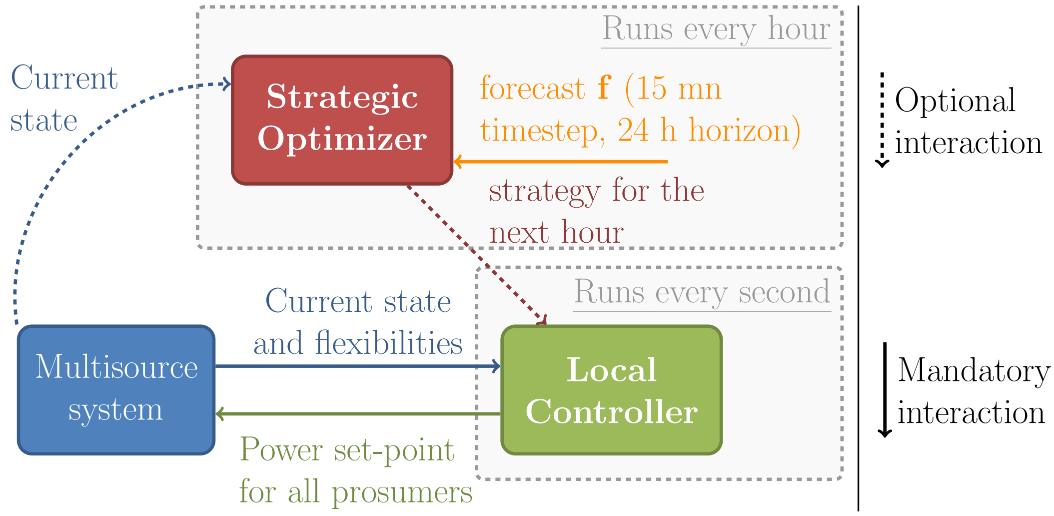


FIGURE 2.4 – Software components interactions.

product, LC would probably be embedded into the energy hub. While SO could be proposed as a web service. We can see on Figure 2.4 that LC and SO are separated components that communicate only through a strategic instruction. An instruction is composed of a target time, and an array of n_p cells: one per prosumer connected to the energy hub. For storage units, an instruction is expressed as a target state of charge. For controllable prosumers, an instruction is expressed as mean power. For the other prosumers (that are supposed non controllable), instructions are empty. An instruction can be sent over a network or shared by components running into the same computer. That allows flexible business models.

Moreover, SO needs to know the current state of charge of every storage unit and the current availability of controllable prosumers. It is also fed by a forecast **f**, that is a $|\mathbb{P} \cup \mathbb{E}| \times H$ matrix, with H the number of periods considered by SO. For prosumers in $\mathbb{P}$, the tariff is forecasted, whereas for prosumers in $\mathbb{E}$, the produced and consumed quantity is forecasted. The way forecasts are computed is explained in Subsection 2.5.1.

Finally, LC applies the computed tactic that is a vector of n_p power values, on the multisource system. LC is fed with current solar production, electricity tariff and flexibility of the elevator due to the current users' calls.

2.4.2 Strategic Optimizer

The problem solved by SO consists of finding the best sources of energy to be used, factoring in the storage capacities, during a long time frame. The goal is to minimize costs related to energy purchasing and battery usage within the time frame, while ensuring the hub is autonomous.

A Generic Formulation Depending on Prosumers Kind

Let us suppose that the optimization period is a constant τ, and that the number of periods in the optimization horizon is denoted H.

Definition 8. *Let $e_i[t] = p_i[t] \times \tau$ be the energy amount produced by prosumer π_i over a period τ.*

Thereafter decision and state variables are defined, depending on prosumers kind:

- Storage units: $\forall \pi_i \in \mathbb{S}$,
  - $\forall l \in \{0, \dots, H-1\}, 0 \leq e_i^{ch}[t_l] \leq -p_i^{min} \times \tau$ (resp. $0 \leq e_i^{dis}[t_l] \leq p_i^{max} \times \tau$) the amount of energy charged into (resp. discharged from) the prosumer π_i (that is a storage unit) between time t_l and time $t_{l+1} = t_l + \tau$. Thus $e_i[t_l] = e_i^{dis}[t_l] - e_i^{ch}[t_l]$.
  - $\forall l \in \{0, \dots, H\}, 0 \leq x_i[t_l] \leq 1$ the state of charge of the prosumer π_i at time t_l.
- Controllable prosumers: $\forall \pi_i \in \mathbb{P}$,
  - $\forall l \in \{0, \dots, H-1\}, 0 \leq e_i^{purch}[t_l] \leq p_i^{max} \times \tau$ (resp. $0 \leq e_i^{sold}[t_l] \leq -p_i^{min} \times \tau$) the amount of energy purchased from (resp. sold to) the prosumer π_i (that is a controllable prosumer) between time t_l and time $t_{l+1} = t_l + \tau$. Thus $e_i[t_l] = e_i^{purch}[t_l] - e_i^{sold}[t_l]$.
- Non-controllable prosumers: $\forall \pi_i \in \mathbb{E}$,
  - $\forall l \in \{0, \dots, H-1\}, p_i^{min} \times \tau \leq e_i[t_l] = \mathbf{f}_i[t_l] \leq p_i^{max} \times \tau$ the forecasted production (if positive) or consumption (if negative) of the prosumer π_i between time t_l and time $t_{l+1} = t_l + \tau$. These e_i are not decision variables but constant data given by an external forecast.

Remark: Although storage units charge and discharge could be modeled as a single variable e_i, two variables (e_i^{ch} and e_i^{dis}) are used in our model, because there are two different yields that impact the charge and the discharge. However, we do not want to charge and discharge the same storage unit at the same time. Both variables are minimized in the objective function to avoid this issue (cf Lemma 2.4.1 and the associated proof hereafter).

The following constraints must be taken into account:

- A minimum energy amount must be kept into storage units.

$$\forall \pi_i \in \mathbb{S}, \forall l \in \{0, \dots, H\}, \qquad x_i[t_l] + \rho_i^{minSOC}[t_l] \geq c_i^{minSOC} \tag{2.4}$$

  For this constraint, a new slack variable is defined: $0 \leq \rho_i^{minSOC}[t_l] \leq c_i^{minSOC}$ is the percentage of storage unit state of charge under a given minimum value c_i^{minSOC} at time t_l. Then, c_i^{minSOC} is the ratio of the storage unit state of charge that the energy hub needs to ensure security in case of grid failure. The ρ_i^{minSOC} variables must be null except in the case of grid failure, so they are penalized with a strong coefficient in the objective function (cf Equation (2.9)).

- The energy-related equation of the energy hub must be satisfied: the sum of consumed and produced energy between time t_l and time $t_{l+1} = t_l + \tau$ must be equal.

$$\forall l \in \{0, \dots, H-1\}, \qquad \sum_{i \in \mathcal{P}} (e_i[t_l]) = 0 \tag{2.5}$$

- The state of charge of storage units must be updated at each time-step with the energy charged and discharged.

$$\forall \pi_i \in \mathbb{S}, \forall l \in \{0, \dots H-1\}, \quad x_i[t_{l+1}] = x_i[t_l] + \frac{c_i^{cy}}{c_i^{ce}} \times e_i^{ch}[t_l] - \frac{1}{c_i^{dy} \times c_i^{ce}} \times e_i^{dis}[t_l] \tag{2.6}$$

  where c_i^{ce} is the energy capacity of the storage unit and c_i^{cy} (resp. c_i^{dy}) is the charging (resp. discharging) yield of the storage unit. Yields are normalized between 0 and 1.

- Let a new slack variable $\forall l \in \{0, \dots H-1\}, 0 \leq \rho_i^{stab}[t_l] \leq p_i^{max} - p_i^{min}$ be the difference between the amount of energy purchased from a controllable prosumer π_i at time t_l and the amount of energy purchased from the same controllable prosumer at time t_{l-1}. This value has to be penalized in the objective function. The associated constraints are:

$$\forall \pi_i \in \mathbb{P}, \forall l \in \{0, \dots H-1\}, \qquad e_i[t_{l-1}] - e_i[t_l] - \rho_i^{stab}[t_l] \leq 0 \tag{2.7}$$

$$e_i[t_l] - e_i[t_{l-1}] - \rho_i^{stab}[t_l] \leq 0 \tag{2.8}$$

  Please note that, for the first period, $e_i[t_{l-1}]$ is set to the corresponding instruction computed for prosumer π_i during the last run of SO. If the current execution is the first one, $e_i[t_{l-1}]$ is set to zero.

- Depending on the exact use case, cyclical constraints (or additional cost factors), such as requiring the battery to be full at the beginning of the morning can be added.

Given those constraints, our economical objective function is given by Equation (2.9).

$$\begin{aligned}
\text{Minimize} \sum_{l=0}^{H-1} \Bigg[& \\
& \sum_{\pi_i \in \mathbb{P}} \Bigg(-c_i^{sold}[t_l] \times e_i^{sold}[t_l] + c_i^{purch}[t_l] \times e_i^{purch}[t_l] \\
& + \min\left[\frac{\min\limits_{\pi_j \in \mathbb{P}}(c_j^{purch}[t_l])}{10}, \frac{\min\limits_{\pi_j \in \mathbb{S}}(c_j^{aging})}{2} \right] \times \rho_i^{stab}[t_l] \Bigg) \\
& + \sum_{\pi_i \in \mathbb{S}} \Bigg(\frac{c_i^{aging}}{2} \times e_i^{ch}[t_l] + \frac{c_i^{aging}}{2} \times e_i^{dis}[t_l] \\
& + 2 \times \max_{\pi_j \in \mathbb{P}}(c_j^{purch}[t_l]) \times \rho_i^{minSOC}[t_l] \Bigg) \Bigg]
\end{aligned} \tag{2.9}$$

where $c_i^{purch}[t_l]$ is the electricity buying price at time t_l and $c_i^{sold}[t_l]$ is the electricity selling price at time t_l thus $-c_i^{sold}[t_l] \times e_i^{sold}[t_l] + c_i^{purch}[t_l] \times e_i^{purch}[t_l]$ is the electricity bill for the l^{th} period and prosumer π_i. These constants are given by the cost function cost_{π_i}, which is considered linear in the current formulation.

On the other hand, c_i^{aging} is a coefficient that allows to have a linear approximation of the impact of the storage unit usage on its aging: $c_i^{aging} = \frac{c_i^{inve}}{c_i^{cye} \times c_i^{ce}}$, the constant c_i^{inve} represents the investment cost of the storage unit; c_i^{ce} is the energy capacity of the storage unit and c_i^{cye} is the mean number of cycles that the storage unit can bear.

Remark: As the battery aging cost is just a way to discourage the controller from using the battery, a first order approximation was chosen. In reality, "small" charges and discharges further impact storage units but we ignore this effect here. This cost could be tuned depending on the results of long-term simulations (typically several years) of the controller and its impact on the battery lifetime. On the other hand, auto-discharge of storage units is neglected for the moment.

Moreover, a minimum energy amount must be kept into storage units in order to ensure autonomy in case of grid failure. As explained above, we cannot ensure that with a hard constraint because we need to allow consuming this reserve during a grid failure. Thus, we use the soft constraint (2.4) with the slack variable ρ_i^{minSOC} that is minimized in the objective function. We could have considered that, in case of grid failure, a different operating mode that allows violating this constraint would be set. But a soft constraint, with a soundly chosen coefficient (twice the maximal price of electricity), does the same job in a simpler way.

In order to reduce the chattering of the energy purchased from controllable prosumers, we minimize the slack variable called ρ_i^{stab}. Finally, both e_i^{ch} and e_i^{dis} are penalized, in order to avoid they could be both positive in the same time.

Lemma 2.4.1. *In all optimal solutions,* $\forall l \in \{0, \dots, H-1\}, \forall \pi_i \in \mathbb{S}$, $e_i^{ch}[t_l] = 0 \vee e_i^{dis}[t_l] = 0$.

Proof. Assume $\hat{\mathbf{e}}$, an optimal solution, such that there exists $0 \le l \le H-1$ where both $\hat{e}_i^{ch}[t_l]$ and $\hat{e}_i^{dis}[t_l]$ are strictly positive. Then, let us consider another solution $\mathbf{e}'$ where all decision variables have the same value except that:

$$e_i^{ch'}[t_l] = \hat{e}_i^{ch}[t_l] - \min(\hat{e}_i^{ch}[t_l], \hat{e}_i^{dis}[t_l])$$

$$e_i^{dis'}[t_l] = \hat{e}_i^{dis}[t_l] - \min(\hat{e}_i^{ch}[t_l], \hat{e}_i^{dis}[t_l])$$

As $e_i^{ch'}[t_l]$ and $e_i^{dis'}[t_l]$ are both penalized in the objective function, then solution $\mathbf{e}'$ admits a lower cost than solution $\hat{\mathbf{e}}$ that was optimal by assumption. □

Note that all variables are weighted with a fraction of the energy tariff in order to ensure the right setting of the objective (reserve for grid failure, electricity cost, ...).

The overall linear formulation is then:

$$\begin{aligned} &\text{Minimize } (2.9) \\ &\text{Subject to } (2.4) - (2.8) \end{aligned}$$

The Multisource Elevator Formulation

In the elevator context, we choose to set c_2^{minSOC} to 0.2 and c_3^{minSOC} to 1.0 because the super-capacitor has to be fully charged in case of grid failure and a 20% charged battery can supply the super-capacitor for several travels. Moreover, the battery is a lead-acid battery with an energy capacity of 3000 Wh, an investment cost of 300 € and a maximum power of 2880 W. The super-capacitor has an energy capacity of 120 Wh, an investment cost of 2400 € and a maximum power of 57600 W. Thus, storage units aging costs are the following ones:

$$c_2^{aging} = \frac{c_2^{inve}}{c_2^{cye} \times c_2^{ce}} = \frac{300}{20000 \times 3000} = 0.000005$$

$$c_3^{aging} = \frac{c_3^{inve}}{c_3^{cye} \times c_3^{ce}} = \frac{2400}{20000 \times 120} = 0.001$$

Table 2.1 shows the linear program applied to the multisource elevator case, in vector form. Variables in bold are column vectors of dimension H, and matrices. Let A, B, A' and B' be

TABLE 2.1 – A linear formulation for the multisource elevator sourcing problem.

$$\text{Minimize} \Bigg[-\mathbf{c_4^{purch}} \odot \mathbf{e_4^{purch}} + 5 \times 10^{-6} \times \mathbf{e_2} + 0.001 \times \mathbf{e_3} + 2 \times \mathbf{c_4^{purch}} \odot \rho_\mathbf{2}^\mathbf{minSOC} + 2 \times \mathbf{c_4^{purch}} \odot \rho_\mathbf{3}^\mathbf{minSOC} + \min\left(\frac{\mathbf{c_4^{purch}}}{10}, 2.5 \times 10^{-6}\right) \odot \rho_\mathbf{4}^\mathbf{stab} \Bigg]$$

$$\begin{array}{rcl}
\sum_{i=1}^{6} \mathbf{e_i} & = & 0 \\
\mathbf{e_4^{sold-}} - \mathbf{e_4^{sold}} - \rho_\mathbf{4}^\mathbf{stab} & \leq & 0 \\
\mathbf{e_4^{sold}} - \mathbf{e_4^{sold-}} - \rho_\mathbf{4}^\mathbf{stab} & \leq & 0 \\
\mathbf{x_2^+} - x_2[t_1] + \mathbf{A}\mathbf{e_2^{ch}} + \mathbf{B}\mathbf{e_2^{dis}} & = & 0 \\
-\mathbf{x_2^+} - \rho_\mathbf{2}^\mathbf{minSOC} & \leq & -0.2 \\
\mathbf{x_3^+} - x_3[t_1] + \mathbf{A}'\mathbf{e_3^{ch}} + \mathbf{B}'\mathbf{e_3^{dis}} & = & 0 \\
-\mathbf{x_3^+} - \rho_\mathbf{3}^\mathbf{minSOC} & \leq & -1.0
\end{array}$$

$$\begin{array}{lllllll}
0 & \leq & \mathbf{e_i^{ch}} & \leq & -p_i^{min} \times \tau & ,\forall \pi_i & \in \mathbb{S} \\
0 & \leq & \rho_\mathbf{4}^\mathbf{stab} & & & & \\
0 & \leq & \mathbf{e_i^{dis}} & \leq & p_i^{max} \times \tau & ,\forall \pi_i & \in \mathbb{S} \\
0 & \leq & \mathbf{e_4^{purch}} & \leq & 50000 \times \tau & & \\
0 & \leq & \mathbf{x_i} & \leq & 1 & ,\forall \pi_i & \in \mathbb{S} \\
0 & \leq & \mathbf{e_6} & & & & \\
0 & \leq & \rho_\mathbf{i}^\mathbf{minSOC} & \leq & 0.2 & ,\forall \pi_i & \in \mathbb{S}
\end{array}$$

$H \times H$-matrices:

$$\begin{array}{l}
\mathbf{A} = diag\left(-\frac{c_2^{cy}}{c_2^{ce}} = -\frac{0.9}{3000}\right) \\
\mathbf{B} = diag\left(\frac{1}{c_2^{dy} \times c_2^{ce}} = \frac{1}{0.9 \times 3000}\right) \\
\mathbf{A}' = diag\left(-\frac{c_3^{cy}}{c_3^{ce}} = -\frac{0.9}{120}\right) \\
\mathbf{B}' = diag\left(\frac{1}{c_3^{dy} \times c_3^{ce}} = \frac{1}{0.9 \times 120}\right)
\end{array}$$

Moreover, the symbol $\odot$ states for the Hadamard product (element-wise product).

As we choose 15 minutes periods ($\tau = 0.25$) and a 24 hours horizon, $H = 24 \div 0.25 = 96$. That gives us a linear program with 96 periods. In practice, we have 12 vectors of H decision variables each and 10 constraints per time step. Thus every hour we solve a linear problem with 1152 variables and 960 constraints. Building and solving the problem with GLPK (*GNU Linear Programming Kit, Version 4.54* 2014) takes about one second.

2.4.3 Local Controller

Our LC is a centralized rule-based controller that computes, in real-time and for a single time-step, a sourcing tactic. The tactic depends on: 1) the relative priority associated to every prosumer, 2) current flexibility of every prosumer, 3) the current strategic instruction if any, or a default instruction otherwise. As SO is based on energy forecasts, some strategic instructions may be infeasible at some points, and LC has to find the best trade-off between current instruction and current situation.

Principles of the Rule-based Algorithm

Each prosumer connected to the energy hub is associated with a priority number. A *priority list* is defined as a permutation of the prosumers set $\mathcal{P}$: $(\pi_{l_1}, \ldots, \pi_{l_{n_p}})$, ordered by decreasing priority numbers. A priority number represents the importance of satisfying a prosumer relatively to the others and is linked to the quality of service. The prosumer that has the biggest priority number will get its preferred flexibility, unless this is impossible. On the other hand, the prosumer that has the smallest priority number will be used to produce or consume the power needed by the other prosumers as much as possible.

Moreover each prosumer has a list of *flexibilities*, that can be discrete power values:

$$(p_i^1, p_i^2, \ldots, p_i^m)$$

or power intervals:

$$([p_i^{1,min}, p_i^{1,max}], \ldots, [p_i^{m,min}, p_i^{m,max}])$$

Flexibilities are ordered by decreasing preference order of the prosumer. For example, if an elevator π_1 is stopped and empty, it can move to a higher floor (flexibility p_1^1), or move to a lower floor (flexibility p_1^2), or stay still and consume standby power (flexibility p_1^3). The preferred flexibility p_1^1 of the elevator is to go in the direction that corresponds to the first user call and stop on the way if other calls have been made on intermediate floors and for the same direction. On the other hand, some prosumers have no discrete flexibilities but a set of possible intervals. For example, a battery can consume or produce a power value bounded by its minimum and maximum power bound.

The third, and last, parameter that influences LC is the *strategic instruction*. If there is no strategic instruction available, LC decides itself of a default instruction. That allows LC to work alone if its link with SO is broken. Setting the default instruction influences performances of the tactic.

Given these three inputs, LC builds a *decision tree* for the current time step. The decision tree has a level per prosumer, ordered by decreasing priority order. In a given level, every node has as many children as the number of flexibilities of the prosumer corresponding to the next level. A tactic is obtained by a depth-first search in the tree, and composed of a power value per prosumer. Such a tactic is also called a "*control vector*" in this subsection.

During the depth first search, if the node holds a single power value, this value is chosen. Otherwise, a default value is chosen in the given interval. If a strategic instruction is available for the current prosumer, the value in the interval, nearest to the instruction value is chosen; otherwise, the value in the interval, nearest to zero is chosen. When a leaf is reached, the algorithm checks if the sum of the chosen power values is equal to zero. If not, the difference is compensated, as much as possible, by each node through backtracking in the tree. When the tree root is reached, if the sum of the power values chosen is null, the solution is kept. If not, the depth first search continues. That way, the first found solution is the best one regarding the prosumers priority order, the preferences of each prosumer and the strategic instruction.

This way to compute a tactic could be used for a wide range of multisource systems. Thus the detailed algorithm is presented in Chapter 4, where its genericity is discussed.

Parameters Values and Objectives

This rule-based algorithm can be tuned depending on the objective, through parameter values described above. We cannot influence prosumers flexibilities but we can choose priority order and default instruction. There are three versions of LC:

MinPeaks The goal of the first controller considered is to minimize power peaks purchased from controllable prosumers.

Opportunistic The second controller considered is supposed to minimize dissipated energy and thus the energy bill.

Secure The third controller considered minimizes storage units usage while guaranteeing that storage units will be ready in case of grid failure.

Priority order and default instruction associated with each of these three controllers are shown in Table 2.2. Let us recall that a strategic instruction is composed of target state of charge for storage units and target power for controllable prosumers, as explained in Subsection 2.4.1. Moreover, LCs classify storage units in three categories:

- those under their minimum state of charge $\mathbb{S}_1 = \{\pi_i \in \mathbb{S} \mid x_i < c_i^{minSOC}\}$,
- those usable to absorb power peaks $\mathbb{S}_2 = \{\pi_i \in \mathbb{S} \setminus \mathbb{S}_1 \mid p_i^{max} \geq \max_{\pi_j \in \mathbb{E}}(-p_j^{min})\}$,
- the other ones $\mathbb{S}_3 = \mathbb{S} \setminus (\mathbb{S}_1 \cup \mathbb{S}_2)$.

TABLE 2.2 – Three LCs and their parameters.

(A) Priority orders[3].

	MinPeaks	Opportunistic	Secure
$\mathbb{P}$	3	3	1
$\mathbb{S}_1$	4	4	3
$\mathbb{S}_2$	2	2	4
$\mathbb{S}_3$	1	1	2
$\mathbb{E}$	5	5	5

(B) Default instructions[4].

	MinPeaks	Opportunistic	Secure
$\mathbb{P}$	standby cons	0 W	0 W
$\mathbb{S}_1$	c_i^{minSOC}	c_i^{minSOC}	c_i^{minSOC}
$\mathbb{S}_2$	100%	x_i	100%
$\mathbb{S}_3$	100%	x_i	x_i
$\mathbb{E}$			

For every LC, the non-controllable prosumers are those with the biggest priority number, meaning that they should not be used to balance the energy hub, if it can be avoided. Note that in case they are multiple non-controllable prosumers on the same energy hub, they cannot always be given relative priority numbers. For example, if different actors (two production plants, two different buildings, etc) need a big amount of energy at the same time, and their is not enough energy, a compromise has to be found. In that case, a more sophisticated controller (probably finding Pareto-optimal solutions) should be used. However, for rather small energy hubs like the multisource elevator, non-controllable prosumers can easily be prioritized. The elevator has a bigger priority number than the solar panels. The elevator would also have a bigger priority number than an air-conditioner in the same building for example.

The MinPeaks LC attributes the second biggest priority numbers to storage units under their minimum state of charge, in order to re-fill them if needed. Then come the controllable prosumers, in order to reduce power peaks from and to them. Finally come storage units used to absorb power peaks, and lastly other storage units. That way, the battery absorbs all power demands if possible, and the supercapacitor can stay fully charged for cases it is needed. Default instructions are used only when no strategic instruction is available. In this

[3]Let us note that the prosumer with the biggest priority number is the one that has the highest priority.
[4]A target state of charge of x_i means the storage unit has to stay at its current state of charge.

case, controllable prosumers are supposed to supply the amount of power consumed by non-controllable prosumers when they are in standby mode. The storage units are preferred to be fully charged.

The Opportunistic LC uses the same priority orders as the MinPeaks LC, only the default instructions change. In order to be opportunistic, no power is taken from the grid by default, and all the storage units shall stay at their current state of charge. Note that the Opportunistic LC's behavior is comparable to a classical rule-based controller.

Finally, the Secure LC gives to controllable prosumers the lowest priority in order to avoid using storage units if possible. The default instruction of storage units usable to absorb power peaks is to be fully charged. The other storage units are supposed to stay at their initial state of charge.

Example of Decision Tree

This rule-based algorithm is based on a decision tree. We can see in Figure 2.5, a decision tree built by LC. Tree nodes represent flexibilities and contain either a possible produced power value, or a power interval with consumption and production bounds. Tree levels are associated with prosumers, in decreasing priority order. In order to find the best tactic, the controller

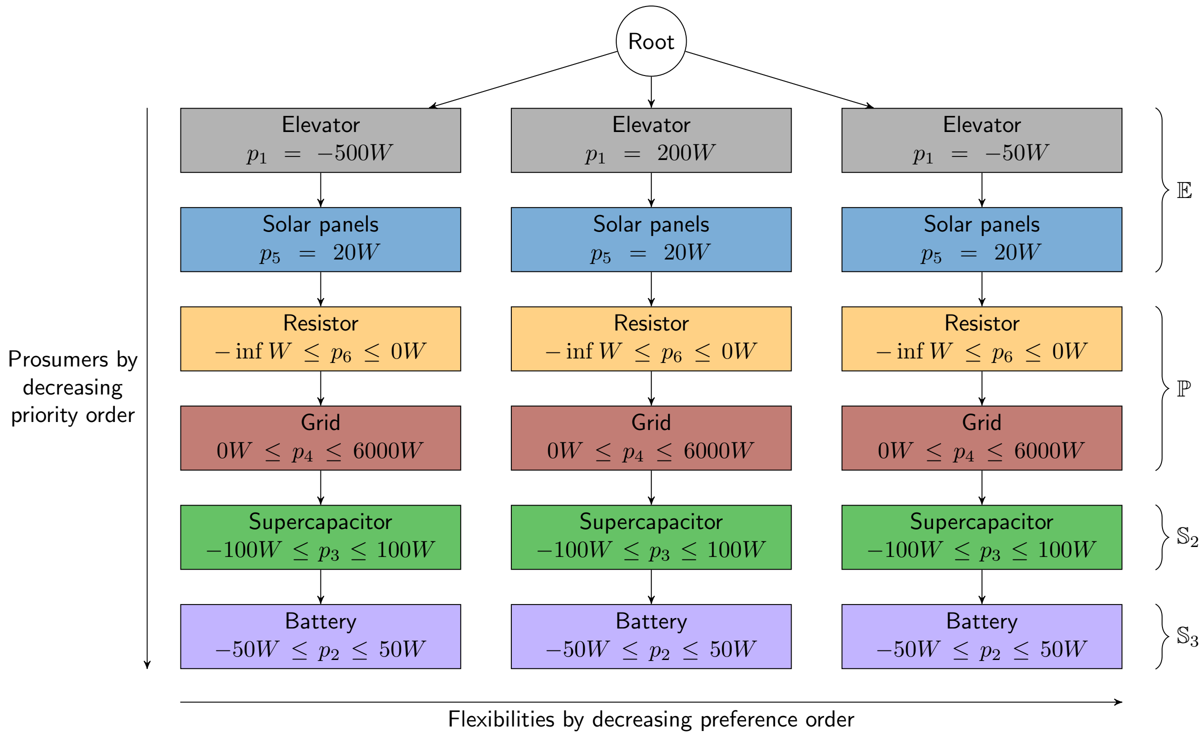


FIGURE 2.5 – An example of the MinPeaks Local Controller decision tree.

searches in the tree, depth first and builds a set-points vector p composed of a power value per node encountered.

An example is presented below to give the reader a clearer idea of how a tactic $p(t)$ is computed, at a given time-step t. $p(t) = [p_1(t), p_2(t), p_3(t), p_4(t), p_5(t), p_6(t)]$ is the control vector composed of n_p chosen power values. We suppose that a strategic instruction is available, valid for the next five minutes, and sets as targets:

- a mean power of 200W for the grid,
- a state of charge of 100% for the supercapacitor,

- a state of charge of 100% for the battery.

A tactic is computed by the LC by trying to apply the strategic instruction if possible, and adapting it if not.

1. Starting from root, the first node encountered is the one this highest priority: the elevator, its preferred flexibility is to consume 500 W.

$$p(t) = [-500, p_2(t), p_3(t), p_4(t), p_5(t), p_6(t)]$$

2. The second node encountered corresponds to solar panels, that produce 20 W.

$$p(t) = [-500, p_2(t), p_3(t), p_4(t), 20, p_6(t)]$$

3. The strategic setpoint for the resistor is 0 W, that is into its flexibility interval.

$$p(t) = [-500, p_2(t), p_3(t), p_4(t), 20, 0]$$

4. The strategic setpoint for the grid is to purchase 200 W, that is into the flexibility interval.

$$p(t) = [-500, p_2(t), p_3(t), 200, 20, 0]$$

5. Suppose that the target state of charge of the supercapacitor is 100% and the current state of charge is 50.0%, so the supercapacitor needs to be charged. If we suppose that the target time of the set-point is five minutes later and the supercapacitor energy capacity is 60 Wh, then 30 Wh have to be charged in five minutes. The corresponding mean power is $p_3^{mean} = 30 \div (5 \div 60) = -360$ W, but the lower bound of the flexibility interval is -100 W. The chosen power value $p_3(t)$ is then -100 W.

$$p(t) = [-500, p_2(t), -100, 200, 20, 0]$$

6. In the same way, the battery has to be charged and the minimum value of its flexibility interval is -50 W.

$$p(t) = [-500, -50, -100, 200, 20, 0]$$

7. Now, the current node is a leaf, so we need to check if the sum of the control vector is null. As, $\sum(-500, -50, -100, 200, 20, 0) = -430$, the current prosumer (that is the lowest priority) needs to change its power value as much as possible to get the sum to zero. So, the battery changes its power value to 50 W.

$$p(t) = [-500, 50, -100, 200, 20, 0]$$

8. The backtracking continues and the current node is the supercapacitor. The sum currently is $\sum(-500, 50, -100, 200, 20, 0) = -330$ and the supercapacitor needs to changes its power value to 100W.

$$p(t) = [-500, 50, 100, 200, 20, 0]$$

9. The current node is the grid and the sum currently is $\sum\{-500, 50, 100, 200, 20, 0\} = -130$. So, the grid needs to changes its power value to 330W.

$$p(t) = [-500, 50, 100, 330, 20, 0]$$

10. Now, the sum is null, and the tree traversal can go back to the root. The current solution is admissible and optimal regarding priority order and flexibilities.

Note that we could avoid some of the backtracks by "propagating" the sum constraint as we go down the tree; but this would make the implementation more complex while backtracks very rarely occur. Indeed, the storage units are sized appropriately to compensate the biggest elevator consumption peaks.

Convergence and Performances

Now let us explain why this algorithm always converges. The needed hypothesis are that: each prosumer has, at least, one flexibility and a finite number of flexibilities (thus the tree has a finite size and a solution always exists).

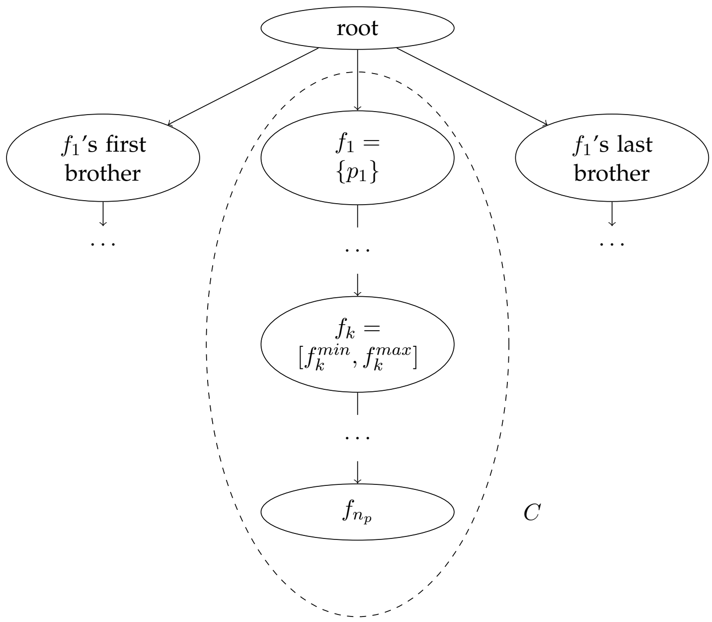


FIGURE 2.6 – Example of search tree for a LC.

Lemma 2.4.2. *When going down the tree, tree leaves can be reached in finite time.*

Proof. There are a finite number of prosumers. When going down the tree, each prosumer has at least one flexibility. A default value in the flexibility set or interval can be deterministically computed in finite time. Thus, the control vector can be filled in with n_p power values, and a tree leaf can be reached. □

Lemma 2.4.3. *From a tree leaf, the tree root can be reached in finite time.*

Proof. Let us note $C = (f_1, \ldots, f_{n_p})$ a path of flexibility nodes from the tree root to the tree leaf f_{n_p}. Let us note $V = (p_1, \ldots, p_{n_p}) \mid \forall i \in \{1, \ldots, n_p\}, p_i \in f_i$ the corresponding control vector. When going back from the tree leaf f_{n_p} to the tree root, if the current node $f_i \in C$ corresponds to a flexibility interval, the chosen power value p_i is deterministically adapted in finite time to make $\sum_{j=n_p}^{i} p_j$ tend towards zero. Then f_{i-1}, parent of f_i, is considered. In the case where f_i corresponds to a discrete flexibility, and $\sum_{j=n_p}^{i} p_j = 0$, then the parent node f_{i-1} is considered. If f_i corresponds to a discrete flexibility and $\sum_{j=n_p}^{i} p_j \neq 0$, the next brother of f_i is considered and a new sub-branch of the tree is explored. If there are no more brother to explore, the discrete flexibility f_i is kept and f_{i-1}, parent of f_i, is considered[5]. Thus, each node

[5] Let us note in that case, the flexibility node in f_i's brothers that gets the sum nearest to zero should be kept in order to get a better tactic, but that would complicate the algorithm.

is considered only once and from every node, the parent node is reached in finite time. By induction, the tree root can be reached in a finite duration. □

Thus, the algorithm always converges but does it converge towards the most valuable solution (i.e.: the control vector with the sum nearest zero)?

Lemma 2.4.4. *Given the following hypothesis:*

1. *a control vector with a null sum exists with the current flexibilities,*

2. *prosumers with a discrete set of flexibilities are the first prosumers in the search tree and,*

3. *each prosumer with a flexibility interval has only one contiguous interval,*

then the equilibrium is reached by this rule-based algorithm.

Proof. According to Hypothesis 1, a control vector with a null sum can be found from the current flexibilities. Let us note C the first path of flexibility nodes that can be obtained from a depth-first search in the rooted tree T, and such that there exists a control vector V with a null sum and composed of a power value per flexibility node in C:

$$\exists \text{ a path } C = (f_1, \ldots, f_{n_p}) \in T \text{ such that } \#C = n_p,$$
$$\exists \text{ a vector } V = (p_1, \ldots, p_{n_p}) \text{ such that } \forall i \in \{1, \ldots, n_p\}, p_i \in f_i \wedge \sum_{j=1}^{n_p}(p_j) = 0 \quad (2.10)$$

When going from the root to the leaf f_{n_p} along C, the first met nodes are those corresponding to discrete flexibilities, according to Hypothesis 2. Let us note f_l the last node along C corresponding to a discrete flexibility. In control vector V, $\sum_{i=1}^{n_p}(p_i) = 0$ and $(f_1, \ldots, f_l)$ are discrete flexibilities, thus $\sum_{i=1}^{l} p_i = -\sum_{i=l+1}^{n_p} p_i$. Therefore, it is possible to find power values in $f_{l+1} \times \ldots \times f_{n_p}$ (with $\times$ the Cartesian product) such that their sum equals $s = -\sum_{i=1}^{l} p_i$.

Let us suppose that the algorithm returns a control vector with a non-null sum. Thus, the whole search tree is gone through. In particular, Path C is examined, along which a control vector $V' = (p_1', \ldots, p_{n_p}')$ is built. Since the control vector V' is built along C too and nodes f_1 to f_l correspond to discrete flexibilities, $\forall i \in \{1, \ldots, l\}, p_i' = p_i$. When going back from the leaf f_{n_p} to f_l, at each level i, a power value p_i' is chosen in f_i' to bring $\sum_{j=i}^{n_p} p_j'$ towards s. Because it is possible to find power values in $f_{l+1} \times \ldots \times f_{n_p}$ such that their sum equals s and because flexibility interval corresponds to a single contiguous flexibility interval (according to Hypothesis 3), the algorithm builds V' such that $\sum_{i=l+1}^{n_p} p_i' = s$. Thus control vector V' has a null sum and is returned by the algorithm. That contradicts the hypothesis of the algorithm returning a control vector with a non-null sum. □

In the multisource elevator system we consider, the only prosumer with a discrete set of flexibilities is the elevator. Moreover, the elevator has the highest priority in the three local controllers proposed. Thus, Lemma 2.4.4 holds and if the equilibrium exists, it can always be reached in our use-case: the elevator can get its preferred flexibility (and not be affected by power supply considerations). If the equilibrium does not exist, all the tree nodes are considered and the algorithm terminates with a control vector that has a non-null sum. That would mean that energy sources cannot supply enough power at this time step. That could occur with the multisource elevator if the grid is down and the storage units are empty for example.

What to do if hypothesis of Lemma 2.4.4 do not hold, as well as the reasons why the lemma is not correct without those hypothesis, would have to be handled in future work.

2.5 Robustness to Forecast Uncertainties and Performance Guarantee

One of the core questions we often encounter in the optimization of design, planning, and control, is the robustness of the proposed solution to uncertainty in the measured (or predicted) data. The robustness of a solution could be defined as its capacity not to be invalidated and stay satisfactory whatever are the uncertain input data. As the SO accuracy depends on the forecasts accuracy, a robustness study has to be performed to identify situations where using our rather complex control solution gives better results than using a simple rule-based controller.

Let us consider two different use cases where a certification has to be given on the control method achievements. In the first one, a customer wants to respect a limit of power consumption enforced by local standards, electricity contract or by the electricity tariff (that can be much more expensive above a given power limit). In this use case, the aim is to determine the optimal sizing of a battery and a supercapacitor in order to certify that the customer will never exceed the given power limit. In the second use case, a customer already has storage units and wants to use them to benefit from Time-Of-Use tariffs or participate in the demand-response market. We want to certify that, if he buys our control solution, he will gain X % on his daily electricity bill with a prescribed high probability.

As we want to provide a guarantee that a certain condition will be satisfied at least a great percentage of the days, the uncertainties to be considered concern all elements that vary from a day to another. Let us note that these elements may be more or less known or predictable at time t_0: the state of charge of the battery is known at t_0; a weather forecast enables to estimate solar production precisely enough for our needs; and a forecast of the elevator's energy needs can also be available but often much less precise. The control method takes such forecasts into account. To certify a property on a given percentage of the days, we have to consider these forecasts as part of the uncertainty set, even though the controller take them as data.

There are an infinite number of daily scenarios of elevator usage; thus we cannot simulate them all. A particular method, that can be used to offer performance guarantee, relies on the fact that an event (such as exceeding P_{max} on a given day) which is not observed over a representative set of N samples (with a large N) has low probability of occurrence (depending on N). In this section, we investigate the use of such a method for the design and control of a multisource elevator.

First an elevator statistical model has been built, to allow drawing elevator usage scenarios. These scenarios are mandatory to allow guaranteeing performance of our control method. In Subsection 2.5.1, the elevator statistical model is presented, as well as the uncertainty set composed of uncertain data and their associated forecasts. Then, in Subsection 2.5.2, the principles of the randomized algorithm are introduced. Let us note that a previous version of this work was published in (Desdouits, Alamir, Giroudeau, and Le Pape, in press).

2.5.1 Data Generation and Forecasts

We draw samples of elevator calls according to a statistical model. This model distinguishes multiple types of travels: morning and afternoon arrivals and departures, lunch breaks, interfloor travels, arrivals and departures of external visitors. Statistical laws are identified based on historical data among the following set of eleven candidate laws:

- Uniform
- Gaussian
- BiUniform
- BiGaussian
- QuadriUniform
- BiLinear

- BiQuadratic
- BiCubic1
- BiCubic2
- BiBeta1
- BiLogistic

As an illustration, the reader is invited to look at Figure 2.7, corresponding to the arrival of people in the morning for two different floors of the building: the horizontal axis corresponds to the number of people arrived and the vertical axis to the time of arrival (in minutes after midnight). The black dotted curve is the actual observation, while the plain coloured curves correspond to approximations of the arrival time distribution by classical statistical laws. According to the average difference between the predicted arrival time (based on a random generation) and the effective arrival time (coming from historical data) on a similar day, all laws except Uniform and BiUniform are good enough for our purpose. For the first experiments we

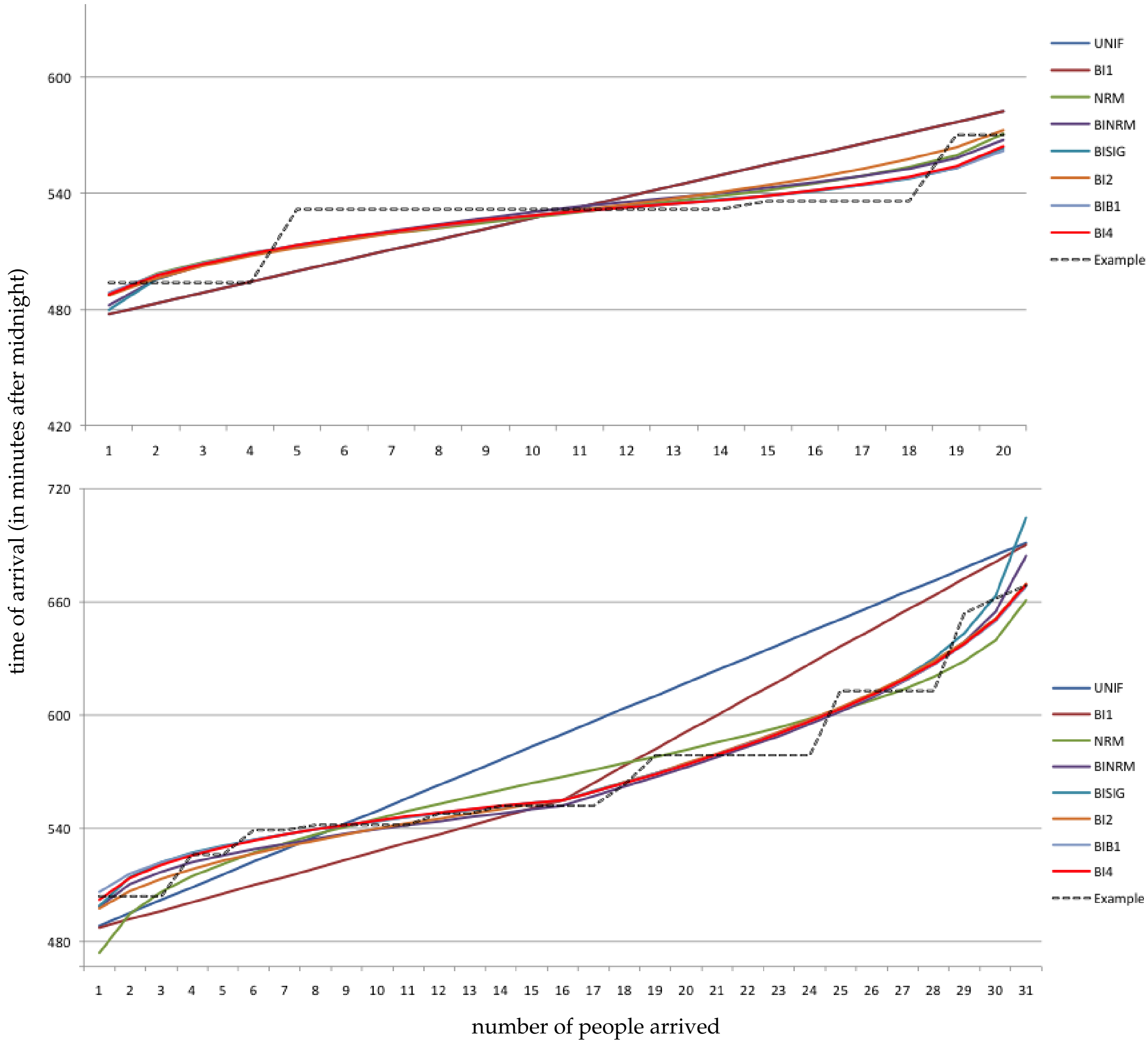


FIGURE 2.7 – Arrival time of people for two different floors, approximation by statistical law compared to real data.

used only Gaussian Laws (average error of 30% of the standard deviation), but further investigations using BiCubic2 (average error of 26% of the standard deviation), for example, could be interesting. However, we expect that would not induce significant changes in the results.

For each travel type and relevant pair of floors, these laws provide information on the number of people moving during the day (depending on day-of-week, week-of-year, etc.), the distribution of their weights, the distribution of the times of the people movements during the day, and the probability that two similar movements are grouped (i.e., several people going to lunch together, etc.). The attendance of the office building is tuned depending on several kinds of days: very low on Sundays, a little bit greater on Saturdays, average on Wednesdays and rather high otherwise. Moreover, the week of year modulates these values. A random generator is used on this basis to generate scenarios.

A forecast of elevator energy consumption is built by averaging energy production or consumption on several daily scenarios of the same kind. Thus, there is one forecast per kind of day. An alternative could be to use standard machine learning techniques to directly forecast energy production or consumption for each SO period.

Definition 9. *A drawn scenario of daily elevator consumption, for the kind of day D, is denoted $w^D_{cons} \in W^D_{cons}$. The associated forecast is denoted $\mathbf{f}^D_{cons}$.*

The second kind of uncertainties is solar radiation, uniformly drawn between two bounds: a typical cloudy day and a typical sunny day. The predicted solar production, known at the beginning of each day, is drawn in the interval: [daily profile minus ten percent ; daily profile plus ten percent]. Note that solar production samples are uniformly drawn, regardless of the kind of day.

Definition 10. *A drawn scenario of daily solar production, is denoted $w_{prod} \in W_{prod}$. The associated forecast is denoted $\mathbf{f}_{prod}$.*

Thus, let $\mathbf{f}^D$ be the forecast that feeds SO. It is deterministic and depends only on the kind of day considered.

Definition 11. *$\mathbf{f}^D = [\mathbf{f}^D_{cons}, \mathbf{f}_{prod}]$, a $2 \times H$ matrix, with H the number of periods considered for the predictions.*

The last kind of uncertainty is the initial state of charge of storage units at the beginning of a simulated day. These states of charge are uniformly drawn between 0% and 100%. Let us note that if we had historical data, we could use a better distribution that the uniform one.

Definition 12. *Drawn initial state of charge of storage units are denoted $w_{initSOC} \in [0, 100]^2$.*

Finally the uncertainty set considered contains the three sources of uncertainties stated before, as well as the forecasts (even if they depend only on the kind of day).

Definition 13. *Let $W = \bigcup_D (W^D_{cons}) \times W_{prod} \times [0, 100]^2$ be the uncertainty set considered and $w^D \in W$ be a scenario drawn for a given day D such that: $w^D = (w^D_{cons}, w_{prod}, w_{initSOC}, \mathbf{f}^D)$.*

2.5.2 Randomized Algorithm Principles

The principles of the randomized algorithm used are the following. Some key design parameters, relevant for the given problem, are chosen. Then, for all possible values of the design parameters, control algorithms are simulated with several uncertainty scenarios. The design vector, that gives admissible solutions and the best objective value on simulated scenarios, is kept.

Because there is an infinite number of possible uncertainty scenarios, we cannot simulate them all. Thus, in order to get a certification that simulation results are representative, a statistical approach is adopted. To obtain this certification, we use results shown in (Alamo et al., 2015) on randomized algorithms. We briefly recall them, and apply them to our context below. Let:

- Θ be the set of possible values of design parameters we want to optimize (i.e.: storage unit possible sizes, etc). The cardinality of Θ is denoted n_C.
- $\theta \in \Theta$ be a vector with n_θ components.
- W be the set of uncertainties (Definition 13), composed of possible elevator consumption curves during a day, solar production curves, initial state of charge, and forecasts.
- $w^D \in W$ an uncertainty scenario drawn for a given day D, as stated in Definition 13.
- $u^*(\theta, w^D)$ be a vector of target state of charge for the battery and target energy purchase from the grid over the day, given by SO.
- $J(\theta)$ be an objective function to minimize.
- C be a set of constraints on design parameters.
- $g : \Theta \times W \to \{0, 1\}$ be a function that returns zero if all constraints $c \in C$ are satisfied and one otherwise.

Then, we use the following result from (Alamo et al., 2015): if we generate N Independent and Identically Distributed samples $\{w^{(1)}, \ldots, w^{(N)}\}$ from W according to the probability Pr_W and then solve the following sampled optimization problem:

$$\min_{\theta \in \Theta} J(\theta) \text{ subject to } \sum_{l=1}^{N} g(\theta, w^{(l)}) = 0 \tag{2.11}$$

with

$$N \geq \frac{1}{\eta}\left(\frac{e}{e-1}\right)\left(ln\frac{n_C}{\delta}\right) \tag{2.12}$$

then, the probability for the optimal solution $\hat{\theta}_N$ to the optimization problem, to violate the design constraint is smaller than η, with a confidence probability no smaller than $1-\delta$. In other terms, we can certify, with a probability $1-\delta$, that the control strategy will satisfy the design constraint in at least $100 \times (1-\eta)$ percent of the cases. In order to solve this optimization problem, we use Algorithm 1.

Let us note that the number of constraint violations $g(\theta, w^{(l)})$ could be authorized to be non-null. For example, an electricity tariff could be such that a given number of power peaks over a maximum value could be authorized over a single year. In that case, the number N of samples to draw would be bigger. We did not consider this case in this manuscript.

2.6 Experiments

For the simulation purpose, we developed in Matlab (The MathWorks Inc., 2015) a simulation engine with dynamic time steps depending on events occurring. All experiments were conducted on a Lenovo ThinkPad W540, Intel Core i7 2.80 GHz, 32 Go RAM.

The building considered is a business tower with nine floors and the elevator has the following characteristics:

Algorithm 1 Randomized algorithm inspired of (Alamo et al., 2015)

```
S ← ∅                                    ▷ N-set i.i.d. samples to be drawn from W
for each kind of day D do
    let n^D be the number of D-days in year
    draw a set S^D of ⌈(n^D × N)/365⌉ samples of D-days
    S ← S ∪ S^D
end for ▷ at this point S = {w^(1), ...} is a set of at least N samples, representative of every kind of day
for each θ ∈ Θ do
    for each w^(l) do
        compute the optimal sourcing strategy u*(θ, w^(l))
        simulate a day w^(l) using u*(θ, w^(l))
        check if the constraints C are respected: g(θ, w^(l))
    end for
    if Σ_{l=1}^{N} g(θ, w^(l)) = 0 then                    ▷ θ is feasible
        compute J(θ)                   ▷ the objective function value for θ
    end if
end for
keep the feasible θ for which J(θ) is minimal
```

standby consumption 50 W

cabin mass 750 kg

counterweight mass 850 kg

nominal velocity 1.0 ms^{-1}

We simulate user calls to the elevator with the statistical model described in Subsection 2.5.1. Moreover, we have implemented a tactic to answer user calls to the elevator. This tactic considers calls in chronological order to choose the destination of the elevator. But, the elevator stops along the way if another call destination is on this way. When the first call in the chronological order is over, the next one is considered. Most of the elevators seem to continue in the same direction after having terminated the first call, until all the other calls going to the same direction are met. But we did not implement this functionality. On the other hand, we have an energy model of the elevator that allows us to compute energy consumption regarding the chosen travel and the weight of passengers. Data used in this model include weights of cabin, counterweight and cable, cable length, elevator base power and nominal speed, altitude of departure and arrival floors, and efficiency for both energy-consuming and energy-producing travels.

We consider that 2 square meters of solar panels are installed on the roof top of the building, and are dedicated to the elevator. These solar panels are supposed well oriented towards the sun and having a yield of 15%.

The following bounds p^{min} and p^{max} on power consumption and production are considered for every prosumer, depending on their physical capabilities:

the elevator π_1**:** $p_1^{min} = -12000$, $p_1^{max} = 7000$

the battery π_2**:** $p_2^{min} = -2880$, $p_2^{max} = 2880$

the supercapacitor π_3**:** $p_3^{min} = -57600$, $p_3^{max} = 57600$

the grid π_4**:** $p_4^{min} = -\infty$ (or 0 if re-selling is disabled), $p_4^{max} = +\infty$

the solar panels π_5**:** $p_5^{min} = 0$, $p_5^{max} = 200$

the resistor π_6**:** $p_6^{min} = -\inf$ because we consider that the resistor was sized big enough for the considered multisource elevator, $p_6^{max} = 0$.

These values were chosen as realistic as possible, regarding our knowledge of the current market. The resistor and grid power capabilities were chosen infinite for the sake of the experiments.

Given this context, many experiments were conducted with different parameters variations. Table 2.3 summarizes the context of each experiment conducted. The first two columns give a reference to the theoretical concept that was experimented, as well as a reference to the experimental results obtained. As a reminder, Subsections 2.4.2 and 2.4.3 refer respectively to the strategic and local controllers, while Section 2.5 is about robustness. The next four columns entitled "Controllers" show which kind of controllers have been used.

The "Forecasts" columns show the way forecasts sent to the Strategic Optimizer were generated in the experiment. "exact" means that the amount of energy in every forecasted period is the same as the amount that could be computed after the simulation (in other terms, the forecast is perfect). If forecasts are not exact, 50 sample days were drawn and averaged to compute the elevator consumption forecast. "pessimistic" means that the forecasts were weighted by the storage units charging (or discharging) yield. This point is explained with more details in Subsection 2.6.2.

The "Energy tariff" columns indicate which kind of electricity contract was used to compute the energy bill. Finally, the "Context" column shows if grid resell was considered or not, and the energy capacity of the storage units.

TABLE 2.3 – Design of experiments and associated parameters.

Reference		Controllers				Forecasts		Energy tariff			Context		
cf theory	cf experiment	strategy	min peaks LC	opportunistic LC	secure LC	exact	pessimistic	flat	peak off-peak	indexed on spot	grid resell	C_2^{ce} (kWh)	C_3^{ce} (Wh)
2.4.2	2.6.1	✓				✓	✗		✓		✗	3	120
2.4	2.6.2	✓	✓			✓	{✓, ✗}		✓		✗	3	120
2.4.3	2.6.3	{✓, ✗}	✓	✓	✓	✓	✓		✓		✗	3	120
2.5	2.6.4	✓	✓			✗	✓		✓		✗	$\{3, 6\}$	$\{30, 60, 120\}$
2.5	2.6.5	{✓, ✗}	✓	✓	✓	✗	✓	✓	✓	✓	{✗, ✓}	3	60

In the experiments, the score of a solution is evaluated according to: the maximum power peak purchased from (or sold to) the grid, the number of those power peaks over a given power limit and the daily cost c^{daily} Key Performance Indicator (KPI). This includes: c^{ebill} be the energy bill of the whole day and c^{aging} be the aging cost associated to the storage units usage that has been done during the day. In addition, depending on the controller, final states of charge can be different. Then we note c^{soc} the cost associated to refill (or to empty) storage units to match their initial state of charge. We consider that corresponding energy is purchased (or sold) at the cheapest tariff.

Definition 14. *The daily cost is defined as* $c^{daily} = c^{ebill} + c^{aging} + c^{soc}$.

For the sake of the example, the elevator we consider would consume almost 3kWh during a typical week day. That means that without batteries, solar panels and energy recovery, the electricity bill with a flat tariff would be about 38 euro cents per day. This is too small to hope big economical gains but perspectives are discussed in Section 2.7.

2.6.1 A Typical Strategy

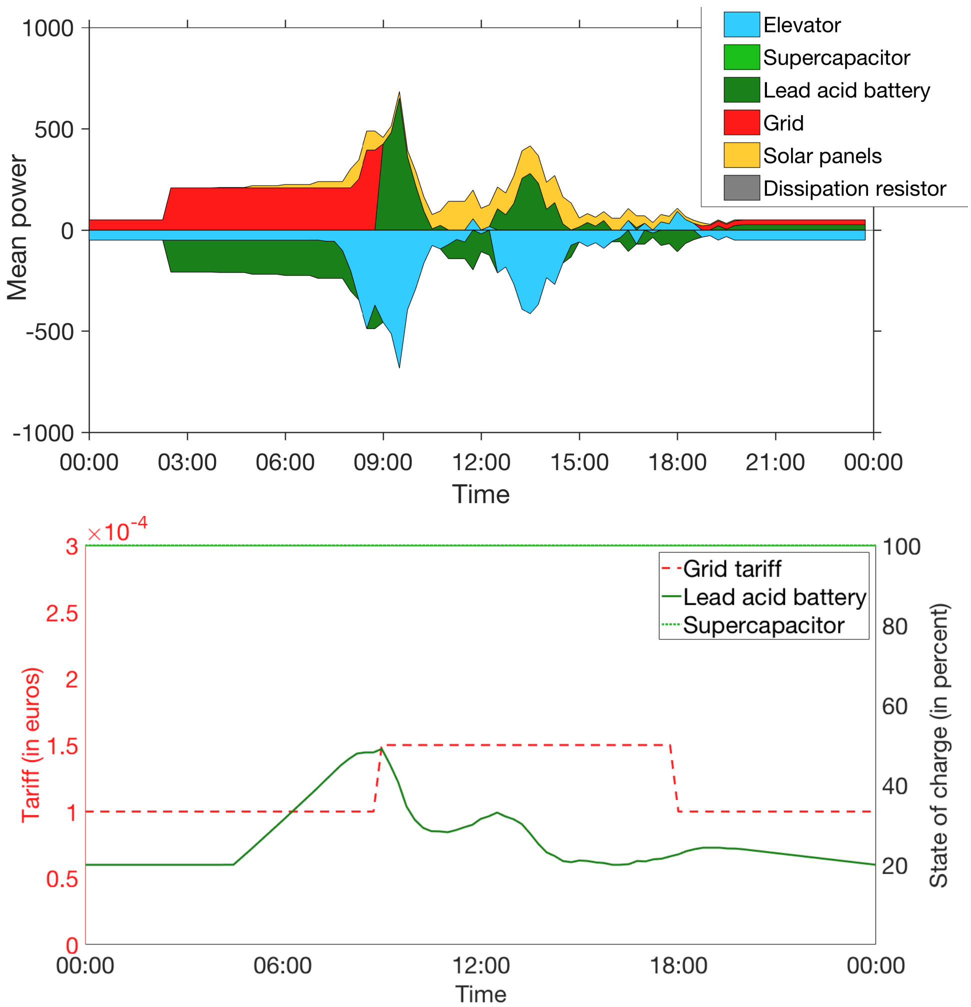


FIGURE 2.8 – A typical strategic plan.

Results obtained by SO, during a typical day, with a peak/off-peak tariff, are illustrated on Figure 2.8. The first sub-figure is an energy layers plot where positive power represents power that is injected in the energy hub (produced by prosumers) and negative power represents power that is taken from the energy hub (consumed by prosumers). The second sub-figure presents the storage units state of charge and the grid tariff. We can see that, when the electricity is cheap (before 09:00), the energy is purchased from the grid (in red), and stored in the battery (in green). After 09:00, no more energy is purchased and the energy needed is discharged from the battery.

In this situation, SO tries to take advantage of the off-peak tariff to charge the battery and avoid to purchase energy from the grid during the peak tariff. We can see that the battery is charged just enough to achieve this goal (about 50% at 09:00).

2.6.2 Forecast Period and Information Loss

In this sub-section, the point is to characterize the error made by SO due to the time periods aggregation. This error is evaluated (on a typical day) by comparing results of the MinPeak LC following the strategy with results computed by SO before the beginning of the day.

The plan computed before the day by SO is the same as on Figure 2.8 and the corresponding MinPeaks tactic is illustrated on Figure 2.9. The first sub-figure represents the power consumption and production of the different prosumers. Clearly, the consumption peaks of the elevator (in blue; that reaches 8 kW) are absorbed by storage units (in green). While in every cases, the maximum *power peak* purchased from the grid is under 400 W, that is far less than 8 kW. The second sub-figure shows the evolution of the storage units state of charge and the grid tariff. In brown, we can see the strategic instructions that are quite well followed by the battery.

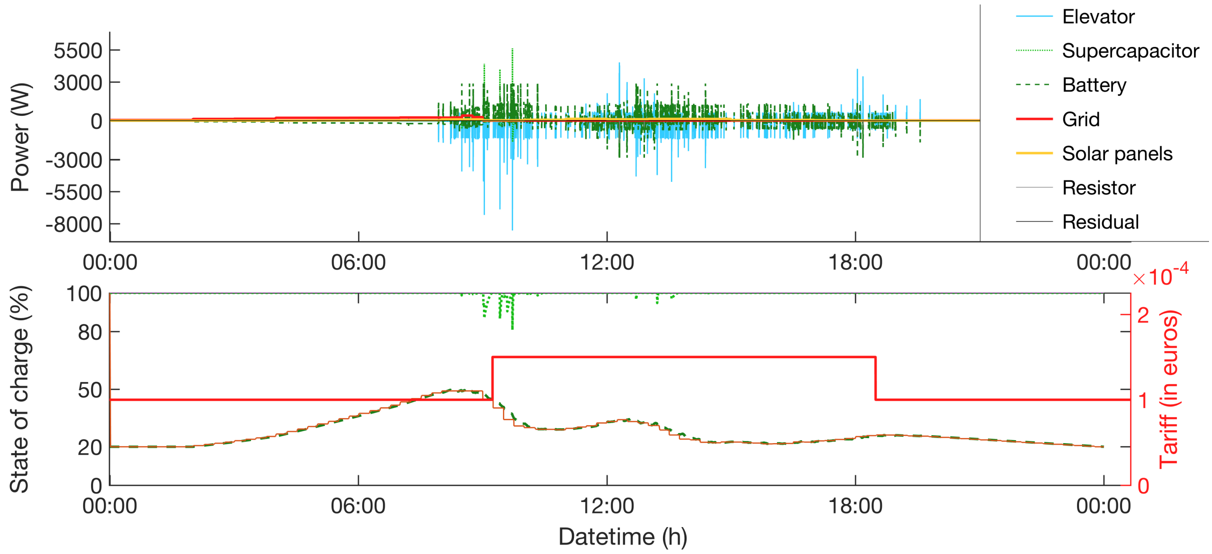


FIGURE 2.9 – MinPeaks LC following strategy.

Now, let us compare numeric results of LC and SO. As the accuracy of SO depends on 1) charging and discharging yields of storage units, and 2) forecasts accuracy. We decided to consider ideal yields (100 %) and realistic yields (90 %). On the other hand, forecasts are exact, in order to evaluate the error due to time periods aggregation, and the impact of the storage units yield. Table 2.4 summarizes the results obtained by SO and by the MinPeaks LC following the instructions, in three cases.

When storage units *yields are ideal*, the energy bill and the amount of energy purchased obtained by the strategy are very close to ones obtained by the tactic. Both values are less than 3 % higher with tactic than with strategy. Though all parameters are ideal, this small *error is due to the energy aggregation* in 15 mn periods into forecasts. Indeed, when summing positive and negative energy values over a period, only the difference is kept. Then, SO take into account only this small amount of energy to purchase (or discharge). In reality, if production occurs before consumption, produced energy is stored and discharged later to supply consumption. But, if consumption occurs before production, the energy has to be found elsewhere before the production could be stored. When yields are ideal and storage units not empty, that has no impact on the energy bill. But when storage units are empty the energy has to be purchased

TABLE 2.4 – Strategic and tactical results with different yields.

view point	yield	forecast	energy bill	aging costs	energy purchased
SO	100 %	exact	0.12 €	0.01 €	1164.3 Wh
LC	100 %	exact	0.12 €	0.13 €	1187.8 Wh
ratio $\frac{LC}{SO}$	100 %	exact	1.03	**13.59**	1.02
SO	90 %	exact	0.14 €	0.01 €	1382.2 Wh
LC	90 %	exact	0.17 €	0.33 €	1640.5 Wh
ratio $\frac{LC}{SO}$	90 %	exact	1.26	31.59	**1.19**
SO	90 %	exact + yield	0.17 €	0.01 €	1705.7 Wh
LC	90 %	exact + yield	0.17 €	0.10 €	1654.5 Wh
ratio $\frac{LC}{SO}$	90 %	exact + yield	0.97	8.67	**0.97**

and that explains the small differences observed above. On the other hand, the aging cost computed by LC is far bigger than those estimated by the SO because the latter did not take into account power peaks. Indeed, to achieve instructions at the end of a strategic period, LC cannot always charge storage units with average power computed from the instruction. It has to absorb consumption and production peaks, and thus to charge and discharge storage units many times within the period. The amount of energy charged into and discharged from storage units is thus far bigger (about 14 times in this case) than computed by SO.

Now let us look at the impact of *realistic yields* on these results. We can see that, in the three metrics, results are worse than before. The reason is that non ideal yields drive non null *energy losses*. Thus, the above mentioned discrepancy due to aggregation is emphasized in presence of non ideal yield.

In order to make SO take into account almost all energy has to transit through storage units, we *integrate the impact of storage units yield into consumption forecasts* of prosumers in $\mathbb{E}$. The strategy becomes pessimistic because in the end not all energy will go through storage units. But the yield taken into account in forecasts becomes a tunable parameter for the pessimistic prediction. Moreover, re-computing a strategy every hour prevents LC to deviate too far away from the target, even if the strategy is not perfectly accurate. In this experiment, we choose a yield value of 0.9, that is the real storage units yield. We can see that the error of SO is really reduced compared to the previous experiment. That also improves LC results, especially on aging costs.

The last thing that has to be explained in the table is: why are *aging costs higher when yields are ideal* than when yields are realistic and forecasts take yields into account? Aging costs are higher in the former because, the battery is emptied at the beginning of some periods and the supercapacitor has to supply the elevator before being refilled by produced energy. On the other hand, when the strategy is pessimistic, this situation occurs less frequently. Since supercapacitors are much more expensive than regular batteries, aging costs in the first experiment are higher than in the third one.

2.6.3 Local Controller Parametrizations

Let us compare results of the different LCs, averaged over fifty elevator usage samples drawn. The initial state of charge of storage units are: 20% for the battery and 100% for the supercapacitor. First of all, let us look at Figures 2.10a, 2.10b and 2.10c that allow us to compare the three different tactics, without any strategy.

Figure 2.10a corresponds to the MinPeaks LC and we can see that there are a few power peaks from the grid at the beginning of the day. This is because not enough energy was produced before consumption, so energy had to be purchased to supply the elevator. Moreover at the end of the day, the battery is not at its minimum state of charge and the supercapacitor is full.

Figure 2.10b corresponds to the Opportunistic LC. With this tactic, all available energy is used as soon as possible, thus the supercapacitor supplies the elevator at the beginning of the day and is emptied. Then, there are many power peaks purchased from the grid in the morning and in the middle of the afternoon.

Figure 2.10c corresponds to the Secure LC. This tactic purchases from the grid all energy needed to preserve storage units from aging. The supercapacitor, that allows the elevator to travel during grid failures, is maintained full. The battery, that has to supply supercapacitor during grid failures, is maintained at its minimum state of charge.

Finally, the MinPeaks Controller following strategy is illustrated on Figure 2.9 and commented in the previous subsection.

Now, let us look at numerical results shown in Table 2.5. First, note that a LC on its own does not take into account electricity tariff at all. On simulated days, we can see that the only efficient

TABLE 2.5 – Comparison of numerical results of different tactics.

Strategic Optimizer	Local Controller	p^{max} W	nb^{peaks}	c^{ebill} €	c^{aging} €	c^{soc} €	c^{daily} €
✓	MinPeaks	**317.7**	**0**	0.13	0.12	0.00	0.24
✗	MinPeaks	7839	**2.1**	0.19	**0.51**	-0.03	**0.66**
✗	Opportunistic	8627	**2.9**	0.12	0.07	0.01	**0.20**
✗	Secure	8775	**3.3**	**0.24**	**0.00**	0.00	0.24

way to reduce the maximum power peak purchased from the grid is the MinPeaks Controller following the strategy. That way, the purchasing peak is about 5% of the maximum authorized power peak. The tactics on their own do not absorb power peaks, because a predictive strategy is necessary to achieve such a goal. On the other hand, the savings the MinPeaks Controller makes on the electricity bill are compensated by the aging costs it occurs. The obtained daily cost is the same as the one of the Secure LC. Let us note that, if the gap between on- and off-peak hourly cost grows, the daily cost of this MinPeaks LC following the strategy also decreases.

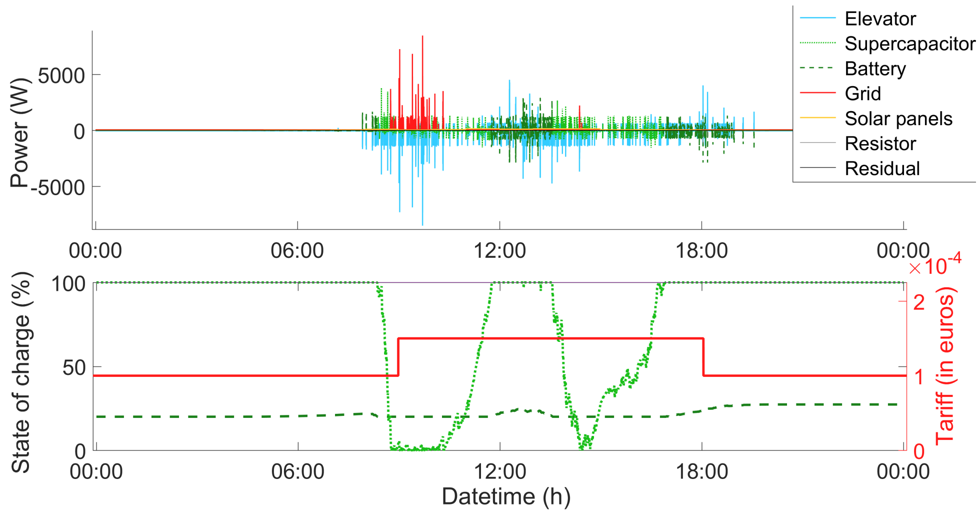
If the MinPeaks LC is on its own (for example during a long network failure), the daily cost increases dramatically but the number of (and the maximum) power peaks stay below the other controllers on their own.

However, if the goal is only to minimize cost, without taking into account power peaks, the Opportunistic LC alone is the best (as far as the ratio between the high/low prices is moderate).

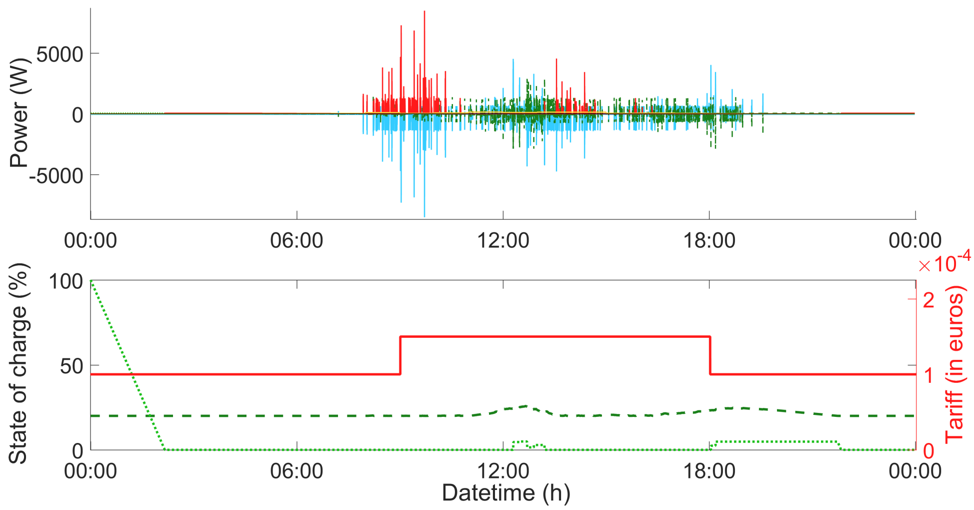
Finally, the Secure tactic does not use storage units at all and thus, has a higher energy bill, but no aging cost.

2.6.4 Which Storage Units to Avoid Power Peaks?

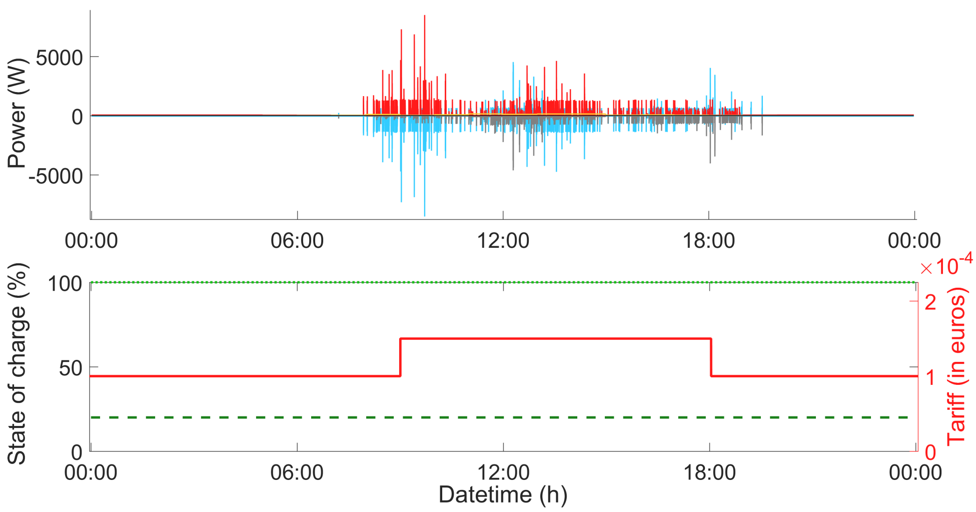
In this sub-section, suppose that the objective is to sell our control solution to a customer admitting a very big annual penalty if he purchases electricity from the grid over a given power value p_{max}. Thus, we want to ensure (with a confidence of $1 - \delta$) that there is probability lesser



(a) MinPeaks LC.



(b) Opportunistic LC.



(c) Secure LC.

FIGURE 2.10 – LC on their owns.

than η to exceed the power value p_{max}. Among solutions satisfying this condition, the most profitable solution should be chosen.

The theoretical results presented in Section 2.5 are used to decide which battery and supercapacitor the customer should buy to be certified to achieve his goal. In this experiment, we consider that the customer subscribes to a peak / off-peak typical French tariff and uses our MinPeak LC coupled with our SO. Now let us define:

- A function $f(\theta, w, u^*)$ that gives the daily cost c^{daily} obtained by the local controller at the end of the day, depending on: θ the considered design parameter, w the daily scenario, and u^* the strategy pre-computed by SO.
- A set $\mathcal{B} = \{3000, 6000\}$ of possible battery energy capacities (in Wh).
- A set $\mathcal{S} = \{30, 60, 120\}$ of possible supercapacitor energy capacities (in Wh).
- A set $\mathcal{F}$ of maximal daily costs certified to the customer (in €).
- The resulting set of design parameters $\Theta = \mathcal{B} \times \mathcal{S} \times \mathcal{F}$, and $\theta \in \Theta$ a vector with $n_\theta = 3$ components; the first component is the battery capacity, the second is the supercapacitor capacity, and the third is the corresponding certified daily cost.
- The design constraint c_1, that states the maximum power peak from (or to) the grid should be below p_{max}:
$$c_1 : \max_{t \in \{0,\ldots,H\}} (|p_4(t)|) \leq p_{max}$$
- The design constraint c_2, that states the daily cost should be below the third component of the design parameter:
$$c_2 : f(\theta, w, u^*) \leq \theta_3$$
- The resulting set of design constraints $C = \{c_1, c_2\}$.
- $J(\theta) = \theta_3$ the certified daily cost (cost of the worst sample).
- The maximum power peak allowed is $p_{max} = 6000$ W.
- Representativeness and confidence probabilities are $\eta = 0.05$, $\delta = 0.05$, and the corresponding number of drawn samples is $N = \lceil \frac{1}{\eta}(\frac{e}{e-1})(ln \frac{n_C}{\delta}) \rceil = 152$.

These parameters given, the optimization problem (2.11) is solved, using Algorithm 1. Let us note that, in practice, the third component of the design parameters does not have to be introduced. Indeed, θ_3 can be chosen as $\max_{w^{(l)} \in S} f(\theta, w^{(l)}, u^*)$. The experiments results are summarized in Table 2.6. In the context studied (chosen elevator, storage units yield, etc), the following report could be done to the customer.

> "The smallest supercapacitor (30Wh) does not permit to avoid all consumption peaks above 6 kW. The worst-case guaranteed daily cost with the 3kWh battery is 0.36 €, from 5 to 11 cents better than with the 6kWh battery. The best choice is the 3 kWh battery and the 60Wh supercapacitor, because they have the smallest investment cost, the smallest daily cost, and avoid peaks.
>
> For the sake of the comparison, the mean daily cost of a business as usual controller in the same case is 0.23 €. So the average gain is 0.28 € per day on these samples. But we know that on some days, the daily cost of the MinPeaks LC can reach 0.36 €, 1 cent more than the worst-case results of a business as usual controller. Thus, we can certify (with probabilities given above) that avoiding peaks above 6 kW will cost at most 1 euro cents more per day than with a classical controller. "

TABLE 2.6 – Storage units sizing for certified maximum power peak.

Battery (kWh)	Supercapacitor (Wh)	max peak (kW)	number of peaks	daily cost (€)
Worst case result certified				
3	30	11.8	2	**0.36**
3	60	5.2	0	**0.36**
3	120	3.0	0	**0.36**
6	30	11.8	2	0.47
6	60	5.2	0	0.45
6	120	3.5	0	0.41
Mean result on drawn samples				
3	30	1.76	0.10	-0.02
3	60	0.57	0	-0.05
3	120	0.51	0	-0.06
6	30	1.79	0.10	-0.07
6	60	0.58	0	-0.10
6	120	0.52	0	**-0.12**

Note that: 1) the mean daily cost computed on the 152 samples is -0.05 €, far better than the certified value (money is saved actually), 2) on average bigger storage units would bring more savings, 3) these results were obtained without considering reselling energy to the grid, reselling will be considered in the next subsection, 3) as the tariff gap grows, the daily cost induced by the MinPeaks LC decreases.

2.6.5 Which Tactic and Tariff to Get Savings?

Now, let us consider a case where a customer already has storage units and wants to use them to improve his energy bill. Several control solutions, and electricity tariffs, must be compared to tell this customer which ones are the best for him. Savings on his electricity bill he will achieve, with the proposed control solution and tariff, have to be certified.

For that purpose, let us define:

- A set $\mathcal{T}$ of possible electricity tariffs: $\mathcal{T}$ = {flat (0.00013), peak/off-peak (0.00015 / 0.00010), indexed on spot with the same coefficient for both purchase and sale of energy (between 0.0002991 and 0.0009386)} (€/Wh). These tariffs are representative of those can be found in France. The flat tariff is cheaper than the others but does not allow a customer to take advantage of his storage units by shifting consumption or inject into the grid. The peak/off-peak tariff is cheaper than the flat one during off-peak hours (18:00 - 9:00) and more expensive otherwise. The indexed on spot tariff is more expensive than the other ones but varies significantly every hour and remunerates the injection of energy at a high value.

- a set $\mathcal{C}$ of possible controllers: $\mathcal{C}$ = {MinPeaks LC that reduces power peaks following strategic instructions from SO, Opportunistic LC that minimizes dissipated energy, Secure LC that does not use storage units at all}. The MinPeaks LC is coupled with the strategic optimizer to take advantage of the storage units to smooth power peaks. The other controllers are on their own to reduce the installation cost of the solution, while achieving savings, or reducing costs.

- A set $\mathcal{M}$ of possible maximum power values from and to the grid:

$$\mathcal{M} = \{[-50000, 0], [-50000, 20000]\}(W)$$

  The first possibility denies re-selling electricity to the grid, while the second one allows re-selling.

- A set $\mathcal{F}$ of maximal daily costs that could be certified to the customer (in €).

- The resulting set of design parameters $\Theta = \mathcal{T} \times \mathcal{C} \times \mathcal{M} \times \mathcal{F}$, and $\theta \in \Theta$ a vector with $n_\theta = 4$ components.

- The design constraint $c^{'}$, that states the daily cost should stay below the 4th component of the design parameter:

$$c^{'}:\ f(\theta, w, u^*) \leq \theta_4$$

- The resulting set of design constraints $C^{'} = \{c^{'}\}$.

- $J(\theta) = \theta_4$ the certified daily cost (cost of the worst sample).

- Two probabilities $\eta = 0.05$, $\delta = 0.05$, and $N = \lceil \frac{1}{\eta}(\frac{e}{e-1})(ln\frac{n_C}{\delta})\rceil = 187$.

The results of the experiment conducted are given in Table 2.7. In this context, we would

TABLE 2.7 – Certified savings on the energy bill.

Strategic Optimizer	Local Controller	tariff resell	flat ✗	flat ✓	peak/off-peak ✗	peak/off-peak ✓	indexed on spot ✗	indexed on spot ✓
Worst case result certified								
✓	MinPeaks		0.34	0.23	0.36	0.05	1.39	**-0.08**
✗	Opportunistic		0.31	0.31	**0.27**	0.27	1.21	1.21
✗	Secure		0.59	0.30	0.53	0.25	2.25	1.06
Mean result on drawn samples								
✓	MinPeaks		-0.10	-0.21	**-0.05**	-0.33	0.18	**-1.23**
✗	Opportunistic		-0.03	-0.03	**-0.02**	-0.02	0.10	0.10
✗	Secure		0.34	0.13	0.30	0.11	1.35	0.54

give the following report to the customer.

> "A business as usual controller that does not use storage units achieves a certified daily cost of 0.59 € with the flat tariff; while a certified daily cost of 0.25€ is achieved with the peak/off-peak tariff, when re-selling is allowed.

> If re-selling is not allowed, the Opportunistic LC coupled with the peak/off-peak tariff provides the best guarantee. The associated certified cost is 0.27 € per day while the average cost obtained is -0.02 € (which is actually a gain). Let us note that a better average is obtained by the MinPeaks LC on the tested samples but with more variability from day to day.
>
> If re-selling energy to the grid is allowed, the indexed on spot tariff should be chosen, along with the MinPeak LC coupled with SO. The corresponding certified daily cost is -0.08 € per day. That means that the elevator operational cost and the storage units investment cost are compensated by the gain on the electricity bill. Moreover the mean daily cost obtained on those samples is -1.23 €."

Let us note that: 1) bigger is the battery, better would be the MinPeaks LC results; 2) the Opportunistic LC gets the same results with or without re-selling, because no energy is re-sold even if it is allowed; the Secure LC's best results occur with the peak/off-peak tariff, when re-selling electricity is allowed, because it re-fills storage units if they are not full at the beginning of the day.

2.7 Conclusion on Optimal Sourcing Strategies

In this chapter, we formalized the sourcing problem and its application to a multisource elevator. We gave a method to solve it, composed of a Local Controller coupled with a Strategic Optimizer. That allows us to tackle real-time issues while taking into account long-term objectives based on forecasts. A linear formulation allows us to compute a strategy and a centralized rule-based algorithm gives us a tactic. Three Local Controller parametrizations are illustrated, and a customer should choose the most adapted to its studied energy hub. Moreover, a way to certify, under uncertainties, savings achieved by a control method for a multisource elevator, is proposed. This Certification Framework is based on a randomized algorithm and its associated bound on the number of needed drawn samples, proved in (Alamo et al., 2015). Using this algorithm, a certified bound on the maximum power peak purchased from (or sold to) the grid by a multisource elevator can be given, as well as a lower bound on the savings achieved. Such a solution could be used by a building manager who wants to participate in the demand-response market. In the same way, advice can be given to a customer regarding the control solution he should buy for his multisource elevator, and the minimum economical gain induced can be certified.

The experiments show that power peaks can be avoided and that a multisource elevator can be made energy cost-free, though this is strongly dependent on the context. In the experiments we ran, gains up to 0.67 € per day could be certified and gains up to 1.57 € per day could be observed. The key point of the method presented is the possibility to find the best design in a given context, and the ability to certify the associated savings. This allows a manager to arbitrate if the chosen design worth to be installed or not.

A legitimate criticism of this work could be that economical gains are very low. First, let us recall that these results are for a unique elevator, while in practice several elevators could share the same battery. Second, reducing power consumption peaks could be very useful to be demand-response aware and to respect future energy limitation laws. In those cases, minimizing the energy bill is only an appreciable addition to the consumption peaks minimization. For the multisource elevator use case, having storage units allows the elevator to evacuate disabled people in case of fire. Using these storage units to minimize the energy bill could dampen the investment. Third, energy prices are going to increase in future years, as will do storage units performances. That will also increase the benefit of this solution. Finally, the proposed solution can be applied to other multisource systems where it can be much more profitable. In fact, as the consumption increases, the profitability also increases, especially if reselling energy is possible.

On the other hand, we will have to compare the cost of maintaining our rather complex solution, regarding the customer value in each use case. If the maximum power peak purchased from the grid is not an issue and the energy tariff considered is a typical French peak/off-peak tariff, the Opportunistic LC (or a similar classical rule-based controller) should be used, as it is simpler to install and maintain.

Having tested this method in a real building (maybe coupled to a Building Management System) would have been highly valuable, but we did not have this opportunity. Moreover, comparing results obtained with several elevators models, or even with a set of several elevators would have been interesting.

Chapter 3

Energy Optimization of a Manufacturing Plant

Contents

In the previous chapter, potential gains were low and multiple elevators would have to be considered to achieve real savings. Here, a manufacturing plant with about 1M€ electricity bill is considered and energy savings could be big. Section 3.1 presents the studied manufacturing plant and introduces potential and issues with its energy optimization. Then, Section 3.2 presents briefly our work on the data acquisition system, and the energy modeling of production activities. In Section 3.3, the scheduling problem we consider is formalized, and a brief state of the art is discussed. The energy model feeds with real data the production scheduling formulations given in Section 3.4. In Section 3.5, experiments are conducted on Benchmarks and instances built from the real plant data, to compare benefits of each proposed method. Finally, the chapter is concluded by Section 3.6.

3.1 Stakes and Potential of Energy Optimization in a Real Manufacturing Plant

Nowadays, the electricity price increases, the demand-response market emerges (Albadi and El-Saadany, 2008), and European regulation pressure on environment is becoming stronger. With this new energy challenge, the usage of energy has to come under scrutiny. In manufacturing plants, the first step, and first need to improve energy efficiency, is to measure and understand the energy consumption. The second step consists of introducing some flexibility in the production plan to cope with the energy price variation. Indeed, if improving efficiency of machines themselves is excluded, one way to reduce the energy bill is to schedule the most consuming production activities when the electricity is cheap.

Here comes the challenge of the "zero stock policy" inherited from kanban (Sugimori et al., 1977), one of the current production management strong principles. In a factory, one of the only ways to get some flexibility without compromising the delivery date is to introduce some storage and to change the inventory policy. By introducing the variable price of energy, the time intervals when energy is spent are not equivalent. Some time intervals are more attractive than others to consume energy. These constraints can be taken into account in the scheduling of production.

The electricity cost function can be constant over time, a peak/off-peak tariff, or even a more complex cost function. In this chapter, a general form of electricity tariff is studied. Time buckets are considered, and in each time bucket, a piecewise-linear function of the instantaneous power consumed gives the electricity cost per time unit. In order to evaluate the potential impact of taking into account such an electricity tariff in a production scheduling problem, a real manufacturing plant is studied.

The studied Schneider Electric manufacturing plant produces electrical enclosures and electrical cabinets from steel coils and electrical outlets and accessories using injection presses and extrusion machines. The plant is composed of three main activities:

1. Cabinet and cells manufacturing,
2. Plastic tubes extrusion,
3. Plastic electrical appliance injection molding.

We chose to focus on the cabinet manufacturing part. The amount of resources spent in the plant is significant: the annual costs of energy and water reaches more than 1.6 M€, out of which the major part (62%) is electricity, but the detailed split of this consumption is unknown.

The current scheduling strategy is led by a lowest stock possible and the activity is planned to produce the production order portfolio in due time, with the right quality level. No real attention is paid to the energy cost for the production line. Energy is considered as a utility that you consume to achieve quality production in due time. Our goal is to understand the energy consumption of the manufacturing process and to manage it, in order to optimize the energy bill, as well as other Key Performance Indicators of the plant manager.

Currently, production plans are computed in three phases, illustrated on Figure 3.1. The first phase consists in having always enough raw material ready to supply production lines. The second phase consists in building production orders from customer demands. The third

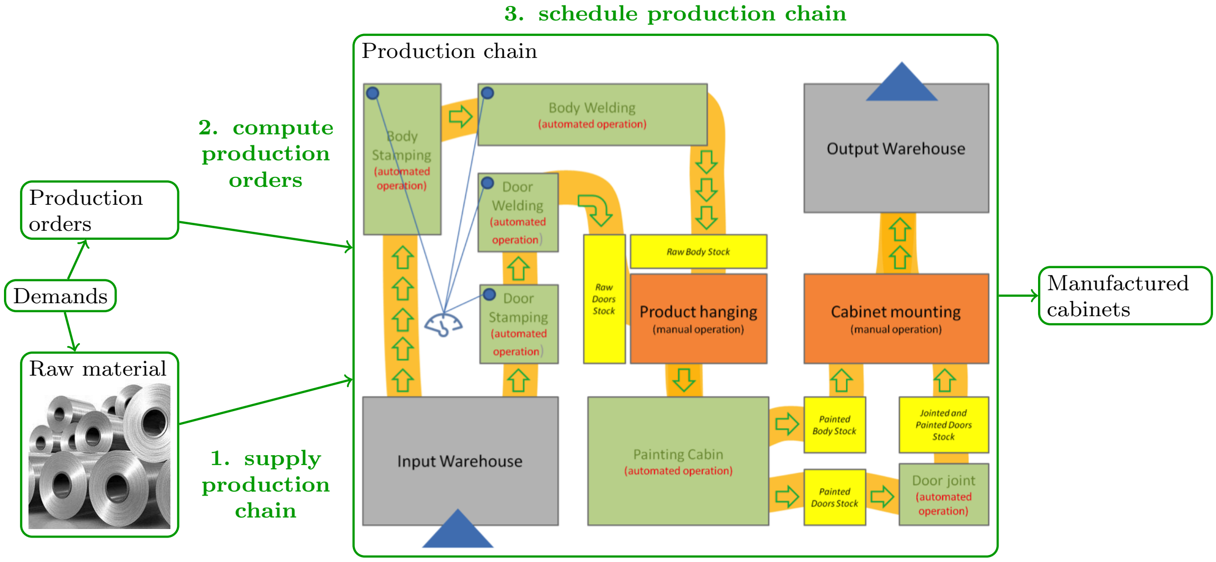


FIGURE 3.1 – De-coupled optimization problems.

phase is a scheduling problem that gives a weekly production plan from pre-computed production orders. This is the problem we address in this chapter.

A scheme of the production lines concerned by the scheduling problem is also given on Figure 3.1. There are two parallel production lines, one for the bodies and one for the doors of the electrical cabinets. Each production line has several stamping and welding machines. Then, bodies and doors are painted together on a carousel. Door gaskets are set, and need to be dried during six hours before bodies and doors could be assembled.

In addition to this pilot plant, we also used Benchmarks from the literature to test our algorithms. More precisely, we used 38 instances from (Le Pape and Robert, 2007) to which energy consumption and tariffs have been added. This enables to compare our results with results from the literature.

3.2 Data Acquisition System, Data Mining, and Energy Modeling

In order to enable automatic computation of production plans, we had to measure energy consumption of all production lines. The data acquisition system we put in place is briefly discussed in Subsection 3.2.1. From these data, we had to build energy models in order to feed the scheduling model.

Some work on energy modeling of a manufacturing plant can be found in the literature. For example, (Le et al., 2013) and (Dietmair and Verl, 2009) address energy modeling of machines. (Dietmair and Verl, 2009) have built a modeling framework for machine-tool energy consumption, based on the finite-state machines theory. (Le et al., 2013) use a similar model and give an approach to reduce the number of required sensors in process tracking by identifying machines operational states, using a neural network. Following a different path, (Fysikopoulos et al., 2014) propose a generalized approach to manufacturing energy efficiency by giving a definition of manufacturing energy efficiency at four different levels: Process level, Machine level, Production line level and Factory level. Interactions between these levels are studied and an example of scheduling is given at the production line level. Once again, energy models of machines are built using finite-state machines.

We tried to build energy models of production activities depending of the kind of reference produced, instead of energy models of machines. That way, we tried to simplify the energy modeling process, while getting precise enough models to schedule production accurately. Our energy modeling methodology is presented in Subsection 3.2.2. Let us note that a previous version of this work was published in (Desdouits, Bergerand, et al., in press).

3.2.1 The Energy Data Acquisition System

The energy data acquisition system is composed of hardware and software components involved in the data collection process: sensors, data concentrator, transmission to a database. This acquisition system is described in Figure 3.2. The focus was made on a selection of consuming machines in the production lines. Four machines have been selected; two per production lines: the door stamping, the door welding, the body stamping and the body welding. Those machines have been selected because their electricity consumption is significant enough and the assumption was made that the consumption profile would vary according to the type of cabinet produced. The goal was to get an accurate enough consumption profile of a set of machines. The system is built from a set of off-the-shelf components and bespoke components:

- The energy sensors: two prototypes of self-powered energy sensors have been used to monitor the machine electricity consumption: one type for mono-phased machines (E3 family Energy Estimator from Schneider Electric Innovation team) and one for three-phased machine (E4 family) ones. Both sensor types send the measured data through radio ZigBee Green Power (ZGP) protocol. The value sent is the cumulated value of energy measured since the beginning of their life; potential overflow is easily managed downstream. The information is not sent on a regular time basis, but on a regular quantum of energy.

- The Sologate gateway: this gateway has been used to take the advantage of both its support of the ZigBee protocol to receive sensors' data and its ability to expose the data through modbus protocol. The Sologate is capable of updating the energy values roughly every 100 ms.

- The LINC gateway: the role of this gateway is to be able to sample the energy value with a 1-second sampling rate. To achieve this, the LINC middleware provided by CEA (Louvel and Pacull, 2014) has been deployed on a Raspberry Pi. It has been specialized in order to collect data and manage the Energy Operation database format. It queries the Sologate gateway through Modbus protocol and builds an aggregation of those data into a single file that contains time series made of the time stamped changes of energy value. Consequently, the timestamps are not regular, but the number of samples is smaller.

- The Energy Operation database and visualization tool: this is a proprietary Schneider-Electric tool offering a data storage capability and a web interface.

A set of dashboards has been configured in order to show data and support analysis. For example, Figure 3.3 shows the global evolution of the consumption of each machine over a month period, the total consumption of the four machines per day and per machine, the weight of each machine in the total consumption. Energy values are expressed in kWh in all the diagrams. We can see that stamping is the most consuming activity; almost 70% of the global electricity consumption for the given month. With these dashboards, it is however difficult to determine whether the variation of electricity consumption from one reference to the other can be exploited to generate savings.

Figure 3.4 shows the evolution of the power consumption of the body production line (in Watts), over a two-month period. We can see that the baseline power is about 7500W for the body part of the production lane. A similar visual analysis helped us to highlight that the baseline power consumption of the whole lane is about 10 kW when machines are idle. Moreover, machines are never switched off and their energy consumption during non-productive periods

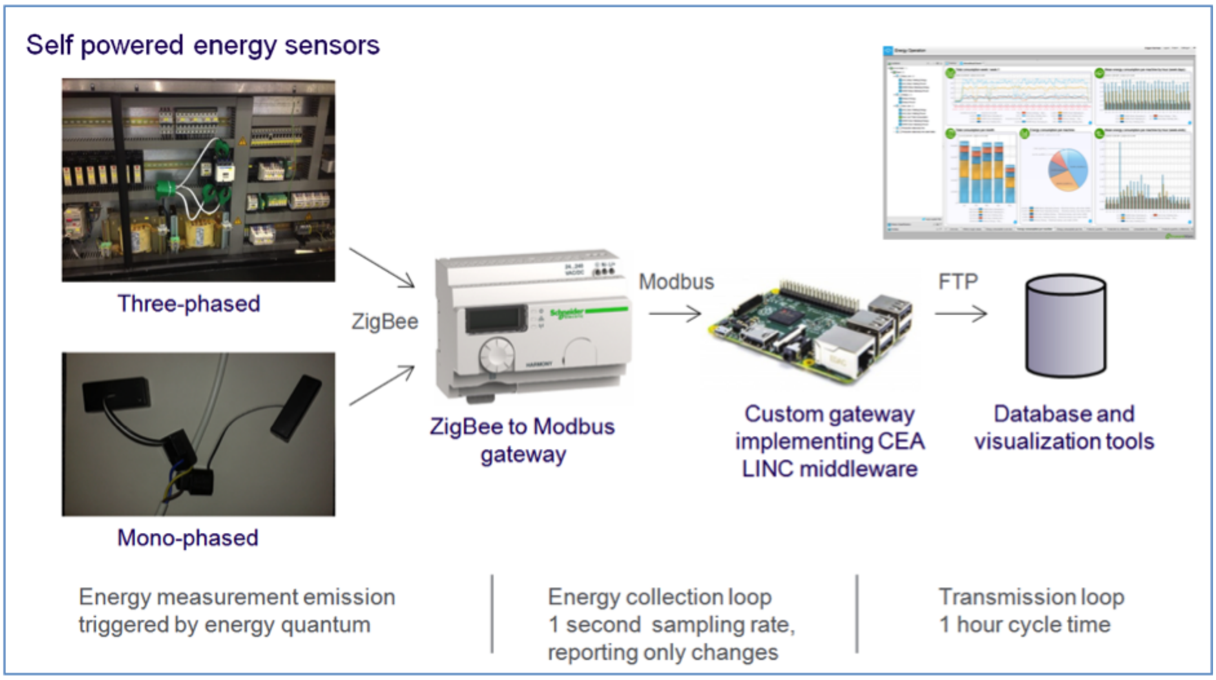


FIGURE 3.2 – Energy data acquisition system.

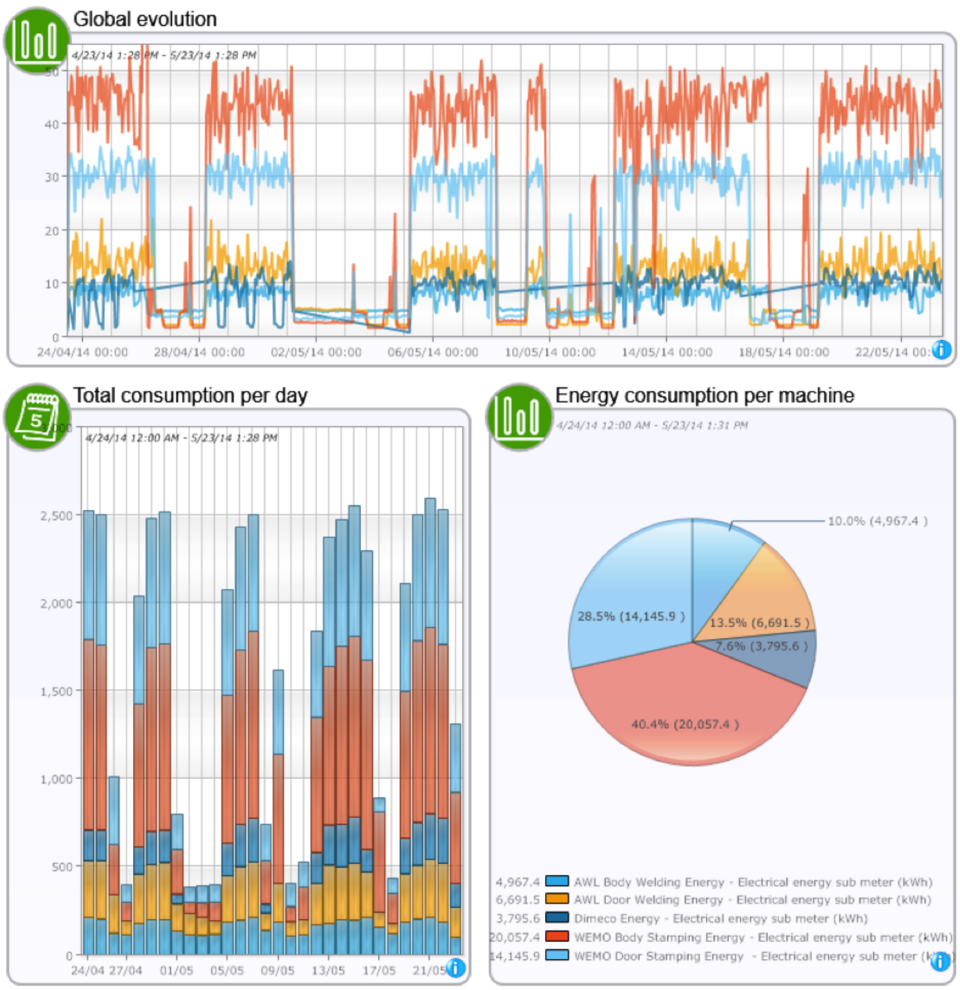


FIGURE 3.3 – Energy Operation dashboard example.

represents about 3% of the total energy consumption. Those observations are used in the following subsection, in order to check the consistency of the consumption model built. They also highlight flexibilities that can be exploited by the scheduler to reduce the electricity bill.

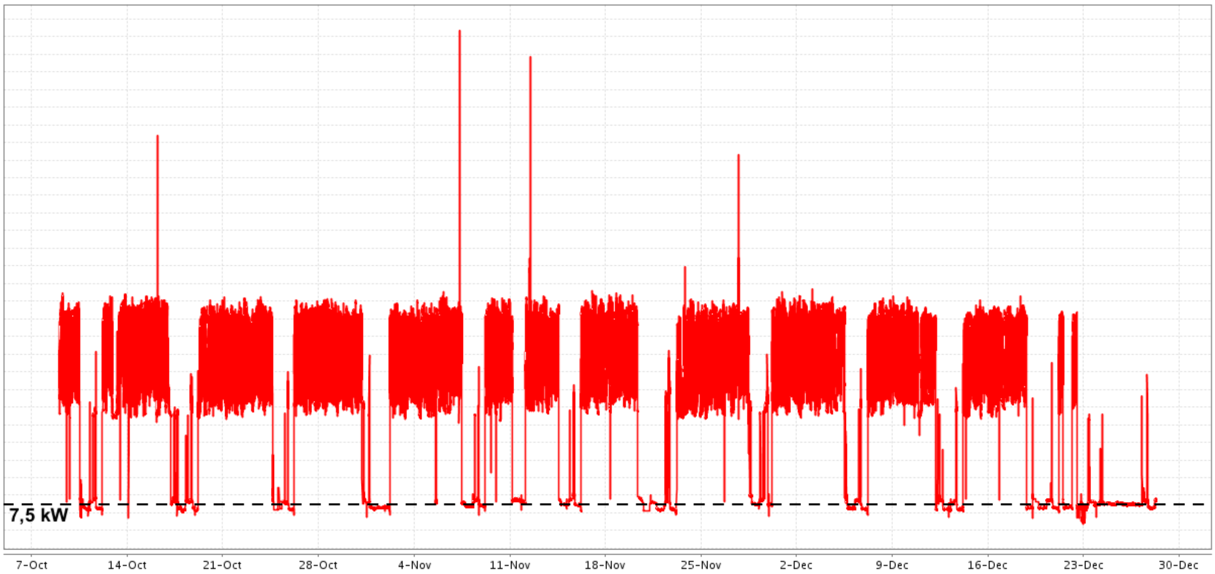


FIGURE 3.4 – Power consumption of the body line over time.

Besides, production data are extracted from the Manufacturing Execution Systems (MES) deployed into the plant. The extracted production data still need to be formatted and processed in order to retrieve the key information. It is to be noticed also that part of the production information is the result of a manual keying; therefore, the time stamping can suffer from some inaccuracy. In particular, start and end times of activities cannot be estimated accurately enough.

A first analysis showed that no production data are available on the Sundays, while the electricity consumption measured is higher than the baseload. The plant manager explained that this is due to Sunday maintenances, not logged in the production system. This observation confirmed that the energy acquisition system put in place is accurate enough to highlight anomalies.

3.2.2 Energy Modeling of Activities

From the gathered data, we wish to evaluate and model the energy consumption of the production activities executed in the plant. To enable this evaluation, we developed a procedure in two stages. Each of the two stages (further described below) is implemented as a web service on a Schneider Electric platform in the cloud. The first service synchronizes the schedule and energy consumption data. The second service uses the result of this synchronization to characterize the energy consumption for each relevant activity. The first service is optional: if the user already has synchronized data, the second service can be used directly.

The overall procedure shall be appropriate (sufficient) under the following theoretical assumptions. Sufficient production and energy data are available and these data are (globally) clean enough. If the energy consumption of the process depends on external factors, we are able to get the associated data and to take them into account in the modeling. For example, if the energy consumption varies with the external temperature, we consider that we can take into account the relevant weather data. If manufacturing activities can overlap in time and consume energy measured by the same sensor (e.g., either because the same machine works on several activities in pipeline or in parallel, or because a unique sensor is used for a group of machines), then the energy consumption of each activity shall not vary too much during the course of the activity. Otherwise, the activity would need to be decomposed in energetically homogeneous sub-activities.

Let us note that the second service provides model quality indicators. Bad values of these indicators are often reported when one of the above assumptions does not apply. A more complex "visual" version of the procedure is currently under study, in order to deal with more complex environments, for which the automatic procedure is insufficient. Services implementation and data analysis were performed with RapidMiner 5 (Hofmann and Klinkenberg, 2013), although in a version customized by Schneider Electric.

First Stage: Energy and Production Data Synchronization

The first stage consists of synchronizing the energy consumption data (one time series per sensor) with production data extracted from the manufacturing execution system (or another legacy system) monitoring the plant. The synchronization algorithm first divides the time line according to the energy time series and determines the portion of each production activity in each time interval. Roughly speaking, the synchronization algorithm consists in re-sampling the production data at the sampling rate of the energy data.

Moreover, the synchronization algorithm offers the option to aggregate or not contiguous time intervals with similar production information into a unique time interval. When aggregation is done, similar time intervals are replaced by a unique time interval, summing both the energy consumption and the production amount data. Such an aggregation presents two advantages: first, the resulting table is smaller and easier to read; second, minor differences during the execution of the same activities are automatically averaged. It presents two drawbacks: the time intervals in the resulting table have different durations and do not coincide with the sampling rate of the energy consumption sensor; second, energy differences during the execution of the same activities are no longer visible.

Second Stage: Construction of the Energy Model of the Plant

The second stage consists in constructing an energy model. It relies on multivariate regression analysis to identify a consumption baseline (i.e., power consumed in the plant, even when it is idle) and the energy required in order to execute a unit of each type of activity in the plant. The regression results give:

- $\mathbf{energy}_r$ is the energy consumed by producing one unit of reference r; this is the coefficient we want to compute for reference r;
- $\mathbf{baseload}$ is the power consumed by the line whatever the production;
- $\mathbf{error}_i$ is a term that represents regression model errors.

The regression can then be expressed in the following way:

$$\forall i \in \{1, \dots, H\},$$
$$totalMeasuredEnergy_i = \sum_{r \in R} (\mathbf{energy}_r \times batchSize_{r,i}) + \mathbf{baseload} \times \tau_i + \mathbf{error}_i \quad (3.1)$$

Where:

- H is the number of time periods considered;
- τ_i is the duration of the time period i;

- $totalMeasuredEnergy_i$ is the amount of energy measured by the energy sensor during time period i;
- R is the set of identifiers of product references;
- $batchSize_{r,i}$ is the quantity of reference r produced during period i.

For both the baseline and each activity, usual statistic quality indicators (absolute error, t-statistic, and p-value) are also provided, enabling to qualify the confidence in the model.

Validation and Application

The result is very sensitive to the precision and accuracy of historical data. We have first validated the algorithm on a set of 34 benchmarks in the ideal case (with perfect energy consumption and manufacturing production data). Then, we have introduced perturbations of these data in order to quantify the algorithm sensitivity. More precisely, we considered the following types of perturbations of the data:

1. a significant energy consumption baseline (i.e., energy consumed when no manufacturing activity is ongoing in the plant);
2. random fluctuations of the energy consumption and/or random errors from the sensors;
3. reduced frequency of sensor measurements;
4. imprecise manufacturing execution system data (start and end times of production activities).

Roughly, the obtained results show an appropriate robustness to perturbations (1) and (2), provided enough measurements are available. Perturbations of type (3) are well handled, as long as the frequency of sensor measurements remains below the characteristic frequency of the manufacturing process. Perturbations of type (4) are the most annoying. Indeed, the information available for making the analysis quickly degrades with the imprecision of activity start and end times.

For the Schneider Electric plant, we applied the two stages on the data of the body production line, with two different synchronization strategies:

- First, we kept periods of the same duration τ. We can see a part of the results obtained on Figure 3.5. There is a line per kind of referencer, and the “value” column corresponds to the $meanPowerReference_r$ value, in Watts per product unit. The three other columns are quality indicators (for example, pValue should be as near zero as possible). We obtained a baseload power consumption of 7299 W for the body production line, consistent with the value of 7500 W suggested by the visual analysis (see Figure 3.4). However, coefficients $meanPowerReference_r$ obtained have a very bad confidence level. This can be explained by a significant variability of the energy consumption during the production of a specific reference.
- The second synchronization strategy aggregates time periods with similar production, as explained above. The baseload obtained is higher (which is obviously wrong), but the activities consumptions are more realistic. The high baseload is due to imprecision in the production log. Indeed, when an activity begins several minutes before being logged, machines are considered idle while they are actually producing a reference and consuming energy. As consecutive non-productive periods are aggregated, this phenomenon virtually increases the computed baseload. On the other hand, activities consumptions are probably lower than the real consumptions (due to the too high baseload), but consistent

Row No.	name	value	error	tStatistic	pValue
421	Avg_productionAW16020116MCRA00002M0	0	1253.721598	0	1
422	Avg_productionAW15100223MCRA00003M0	0	1247.522016	0	1
423	Avg_productionAW12100676GHLA00062M0	994.936094	212.803318	4.675379	0.000003
424	Avg_productionAW13120363SRTA00044M0	427.319816	45.684442	9.353727	0
425	Avg_productionAENNA08361M0	0	243.265026	0	1
426	Avg_productionAW11120132TGFA40018M0	422.468515	78.641276	5.372096	0.000000
427	Avg_productionAW13110135TGFD10006M0	0	159.630895	0	1
428	Avg_productionAW16020486JCRA00003M0	6210.594032	1262.222747	4.920363	0.000001
429	Avg_productionAW14100675VMNA00006M0	0	1274.262075	0	1
430	Avg_productionA30348228_1M0	605.446678	101.797690	5.947548	0
431	Avg_productionAW15010097VMNB00002M0	0	1246.041148	0	1
432	Avg_productionAENNA08340M0	0	533.136979	0	1
433	Avg_productionA30349672_1M0	898.398615	219.565991	4.091702	0.000043
434	Avg_productionAW10040268TGFA70015M0	557.770394	99.196414	5.622889	0.000000
435	Avg_productionA30348784_1M0	0	88.273102	0	1
436	Avg_productionA18911119_1M0	480.680003	82.490633	5.827086	0
437	Avg_productionA18927440_1M0	511.016226	54.070091	9.450996	0
438	Avg_productionA18926924_1M0	440.193762	44.871857	9.810019	0
439	Constant	7298.940000	215.130005	33.928043	0

FIGURE 3.5 – Sample of the regression results, on non-aggregated data.

Row No.	name	value	error	tStatistic	pValue
399	Avg_productionAS3D06913B20077M0	2953.481888	421.398350	7.008765	0
400	Avg_productionAW15100223MCRA00003M0	1077.810216	847.588638	1.271619	0.203533
401	Avg_productionAW12100676GHLA00062M0	771.717322	144.584323	5.337490	0.000000
402	Avg_productionAW13120363SRTA00044M0	390.414726	31.041020	12.577381	0
403	Avg_productionAENNA08361M0	605.522986	165.284325	3.663523	0.000250
404	Avg_productionAW11120132TGFA40018M0	384.822108	53.431408	7.202170	0
405	Avg_productionAW13110135TGFD10006M0	386.323274	108.456892	3.561998	0.000369
406	Avg_productionAW16020486JCRA00003M0	4708.357835	857.598353	5.490167	0.000000
407	Avg_productionA30348228_1M0	542.735261	69.164922	7.846973	0
408	Avg_productionAW15010097VMNB00002M0	382.635910	846.659515	0.451936	0.651323
409	Avg_productionAENNA08340M0	1316.873950	362.226967	3.635494	0.000279
410	Avg_productionA30349672_1M0	710.383284	149.242043	4.759941	0.000002
411	Avg_productionAW10040268TGFA70015M0	428.793810	73.671419	5.820355	0
412	Avg_productionA30348784_1M0	984.897664	499.641908	1.971207	0.048723
413	Avg_productionA18911119_1M0	415.807485	56.050032	7.418506	0
414	Avg_productionA18927440_1M0	432.071421	36.740501	11.760085	0
415	Avg_productionA18926924_1M0	360.716762	34.892763	10.337867	0
416	Constant	12819.176718	151.402511	84.669512	0

FIGURE 3.6 – Sample of the regression results, on aggregated data.

between one another (see Figure 3.6). Even if this result is not perfect, this is much more satisfactory than the previous result, as it enables to identify the relative consumption of each activity, and hence the activities to perform when the price of electricity is low.

Finally, we built an approximation of the activities energy consumption by keeping the baseload computed with non-aggregated data, and increasing the energy consumption of activities (computed with aggregated data), according to the difference between the two baseloads. We found out that the power consumption of activities of the body production line varies between 13kW and 23kW, with a baseload of 7.3 kW. Due to the great variability of these consumptions, an efficient scheduling algorithm should be able to achieve savings on the electricity. Indeed, this estimation of the energy consumption of each activity is used during scheduling to consider the cost of electricity consumption as one of the optimization criteria to minimize. This incites the scheduler to schedule the activities for which the electricity consumption is high at times at which the tariff is low. This way, a trade-off between storage costs and energy cost can be done, depending on the electricity tariff.

3.3 Scheduling with Electricity Cost: Problem Statement, State of the Art and Complexity

The goal is to determine if it is worth considering energy costs in the scheduling of a given factory or workshop in a factory. And in particular, what kinds of compromises are worth considering between energy and intermediate or final product inventory. With this goal in mind, we implemented three Mixed-Integer Linear Programs to model the problem with different constraints relaxations. Moreover, a simple Local Search has been developed to tackle the complexity of the studied problem.

First, the scheduling problem studied is formalized in Section 3.3.1, and the Graham's notation is given. Then in Section 3.3.2, a brief review of related papers that can be found in the literature is made, along with the complexity results. Let us note that a previous version of this work was published in (Haouassi et al., in press).

3.3.1 Scheduling Problem

In this section, the constraints of the scheduling problem are first described, then an optimal solution is characterized, finally the corresponding Graham's notation is given. To make the problem description simple, we will focus on the case of our pilot plant. The reader can easily apply our approach to any generalized flow-shop or job-shop problem with time lags, energy resources and a combination of tardiness, earliness, intermediate storage and energy cost.

Instance

We consider a set of two types of materials: final materials (different references of electrical cabinets), and intermediate materials (cabinet bodies and cabinet doors, at different production states). Moreover, each material has an associated storage cost that models the cost of holding this material. For instance, the storage cost of a body is higher than the storage cost of the corresponding door since the body is bigger and more difficult to transport and to store. Moreover the body is composed of more steel than the door and thus its economical value is higher.

Definition 15. *Let M be the set of all those materials. $\forall mat_m \in M$, let $sc_m \in \mathbb{R}^+$ be the storage cost of one unit of the material mat_m.*

The plant receives a set of customer demands, each of them associated to a desired quantity of a given material and a due date. A tardiness cost for delivering a demand after the due date is decided, depending on the kind of demand.

Definition 16. *Let D be the set of customer demands, and $\forall dem_d \in D$, let:*

$duedate_d \in \mathbb{N}$ *be the due date of the demand,*

$fmat_d \in M$ *be the final product to satisfy the demand,*

$q_d \in \mathbb{N}$ *be the needed quantity of product,*

$tc_d \in \mathbb{R}^+$ *be the cost of delivering one time unit after $duedate_d$.*

Each customer demand has to be satisfied by a job, composed of six activities. For each job, the precedence graph between activities is the same in-tree.

Definition 17. *Let:*

J *be the set of jobs,*

$j_d \in J$ *be a job such that $j_d = \{a_1^d, \ldots, a_6^d\}$,*

G_d *be the precedence in-tree associated to a job j_d,*

A *be the set of all activities associated to every demand: $A = \bigcup_{j_d \in J} j_d$,*

G *be the precedence graph between all activities, a collection of in-trees: $G = (A, \mathcal{A}) = \bigcup_{j_d \in J} G_d$.*

Let us note that, depending on the context, we use the job-related indexing, or the absolute indexing of activities. For example, Activity a_2^d is the second activity inside Job j_d, but a_2 is the second activity in the whole set of activities A.

The plant has a set $R = \{r_1, \ldots, r_5\}$ of machines with unitary capacities. Each activity $a \in A$ has to be scheduled on a given machine. Moreover, for each job j_d, the same assigment of the activities of j_d on the machines is given.

Definition 18. *Let $R = \{r_1, \ldots, r_5\}$ be the set of considered machines. $\forall j_d \in J$, the assignment of the activities on the machines is: (a_1^d, r_1), (a_2^d, r_2), (a_3^d, r_3), (a_4^d, r_2), (a_5^d, r_4), and (a_6^d, r_5).*

Besides, each precedence arc $(a_i^d, a_j^d) \in \mathcal{A}$, holds the intermediate material reference and quantity and a minimum delay (time lag) between activities. Since the precedence graph is a disjoint collection of in-trees, every activity has at most one successor. Thus, data related to an arc (a_i, a_j) are considered as activity a_i data.

Definition 19. *$\forall a_i \in A$, let:*

vpt_i *be the processing time for every unit of material a_i produces (called variable processing time),*

pt_i *be the total minimum processing time of a_i,*

res_i *be the machine on which a_i has to be processed,*

p_i *be the power consumption of the activity a_i,*

$imat_i$ *be the intermediate material produced by a_i,*

q_i *be the number of units of this $imat_i$ activity a_i has to produce for each unit of final material needed,*

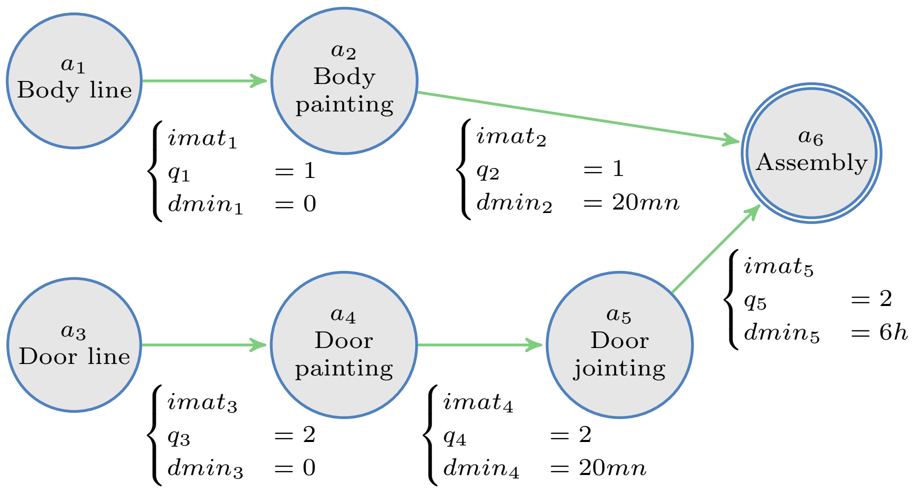


FIGURE 3.7 – Precedence graph of a job j_d.

$dmin_i$ be the minimum time lag between the production of each unit of $imat_i$ and the consumption of the same unit.

Figure 3.7 shows the precedence graph of a job j_d. The graph is an oriented in-tree with six activity nodes. Each activity is scheduled on a different machine, except for the painting activities that share the same painting machine (represented by an exclusive or). Each arc of the graph corresponds to an intermediate material, a quantity, and a minimum delay between activities. For example, in order to produce one unit of the final material, Activity a_6 has to consume one painted body and two painted jointed doors. In order to consume a painted jointed door, a drying delay of the joint of six hours has to be respected.

Let us note that, in practice, the painting process is a carousel on which doors and bodies are hung, painted, then unhung. In order to model correctly human resources who hang and unhang, the painting activities a_2 and a_4 have been split into two different activities with a painting delay between them. Let us also note that only the door line a_3 and the body line a_1 require a significant amount of electricity; other operations being low-consumers or manual.

Solution

A feasible solution to this problem is a schedule of the activities of every job. Both preemptive and non-preemptive solutions can be considered. Non-preemptive solutions are more easily applicable in a real factory, whereas preemptive solutions can really improve objective value.

Let st_i (resp. et_i) be the start time (resp. completion time) of the activity a_i. Let st_i^k be the time when activity a_i starts the production of the piece of material corresponding to the k^{th} piece of final material. The following constraints define the problem:

- the activities duration has to be respected:

$$\forall a_i \in A, \qquad st_i = st_i^1 \tag{3.2}$$

$$et_i = st_i^{q_d \times q_i} + vpt_i \tag{3.3}$$

$$\forall k \in \{1, \ldots, q_d \times q_i\}, \qquad st_i^{k+1} \geq st_i^k + vpt_i \tag{3.4}$$

  Equation (3.2) states that the starting time of an activity is the starting time of the production of its first piece of material. Equation 3.3 forces the completion time of an activity to equals the completion time of its last piece of material. Indeed, there are $q_d \times q_i$ pieces of material produced by an activity a_i because q_d is the number of final pieces of material required by the demand, and q_i is the number of intermediate pieces of material a_i has to produce for each piece of final material. Finally, Equation 3.4 ensures that a piece of material can be produced only after the previous one have been finished.

- at most one activity is scheduled on the same machine at the same time:

$$\forall (a_i, a_j) \in A \times A | res_i = res_j \wedge i \neq j,$$
$$st_i \geq st_j \Rightarrow st_i \geq et_j \quad (3.5)$$

- every piece of material $imat_i$ consumed by a_j must have been produced by a_i at least $dmin_i$ sooner:

$$\forall a_j \in A, \forall k \in \{1, \ldots, q_d \times q_j\},$$
$$st_j^k \geq \max_{a_i \in A | \exists (a_i, a_j) \in \mathcal{A}} \left(st_i^{\lceil k \times q_i / q_j \rceil} + vpt_i + dmin_i \right) \quad (3.6)$$

Indeed, when there is an arc $(a_i, a_j) \in \mathcal{A}$, then the k^{th} piece of material can be produced by a_j only when a_j can consume the corresponding $\lceil k \times q_i / q_j \rceil^{th}$ piece of material produced by a_i. Let us note that the kinds of precedence constraints to be considered (start-to-start, end-to-start, etc.) depend on the manufacturing process studied. The real plant studied in this document can be modeled by start-to-start precedence like the equation presented above.

Moreover, an optimal solution minimizes:

the storage costs

$$\sum_{mat_m \in M} (sc_m \times sq_m) \quad (3.7)$$

where sq_m is the number of units of mat_m stored, multiplied by the duration they were held in stock,

the tardiness costs

$$\sum_{d \in D} (tc_d \times l_d) \quad (3.8)$$

where l_d is the delivery tardiness of Demand dem_d,

the energy costs

Definition 20. *Let T be the time interval considered for the scheduling problem. Let p_{max} be the maximum power that can be consumed by the studied plant and $P = [0, p_{max}]$ be the related interval of definition of possible power values. Then, let us define $power : T \to P$ a function of time that gives the overall power consumption of the plant at time t, such that:*

$$power(t) = \sum_{a_i \in A | a_i \text{ is running at } t} p_i \quad (3.9)$$

We consider a set of buckets, that are time intervals where the energy cost function does not change. For each time t in a bucket $\mathcal{B}_l$, the electricity cost is given by the same piecewise-linear cost function $f_{\mathcal{B}_l}$, depending on $power(t)$. The power capacity, for a fixed bucket $\mathcal{B}_l$, is divided into capacity intervals (non necessarily uniform), such that in each interval the function $f_{\mathcal{B}_l}$ is linear.

Definition 21. *Let:*

$\mathcal{B}$ *be the set of buckets,*

$\mathcal{B}_l \in \mathcal{B}$ *be a bucket such that:* $\mathcal{B}_l = [\mathcal{B}_l^{inf}, \mathcal{B}_l^{sup}[$,

I_l *be the ordered set of power capacity intervals for a given bucket* $\mathcal{B}_l$,

$I^h \in I_l$ *be a power capacity interval such that:* $I^h =]cmin^h, cmax^h]$,

fc^h *be the fixed cost that occurs when:* $power(t) \geq cmin^h$,

vc^h *be the variable cost that occurs when:* $cmin^h \leq power(t) \leq cmax^h$,

$f_{\mathcal{B}_l} : P \to \mathbb{R}^+$ *be the piecewise-linear cost function for a given bucket* $\mathcal{B}_l$*, such that:*
$$f_{\mathcal{B}_l}(p) = \sum_{I^h \in I_l | p > cmin^h} \left[fc^h + vc^h \times (\min(cmax^h, p) - cmin^h) \right]$$

The total energy cost over a time period T is given by:

$$\mathcal{E} = \sum_{\mathcal{B}_l \in \mathcal{B}} \int_{\mathcal{B}_l^{inf}}^{\mathcal{B}_l^{sup}} f_{\mathcal{B}_l}(power(t))\, \mathrm{d}t \tag{3.10}$$

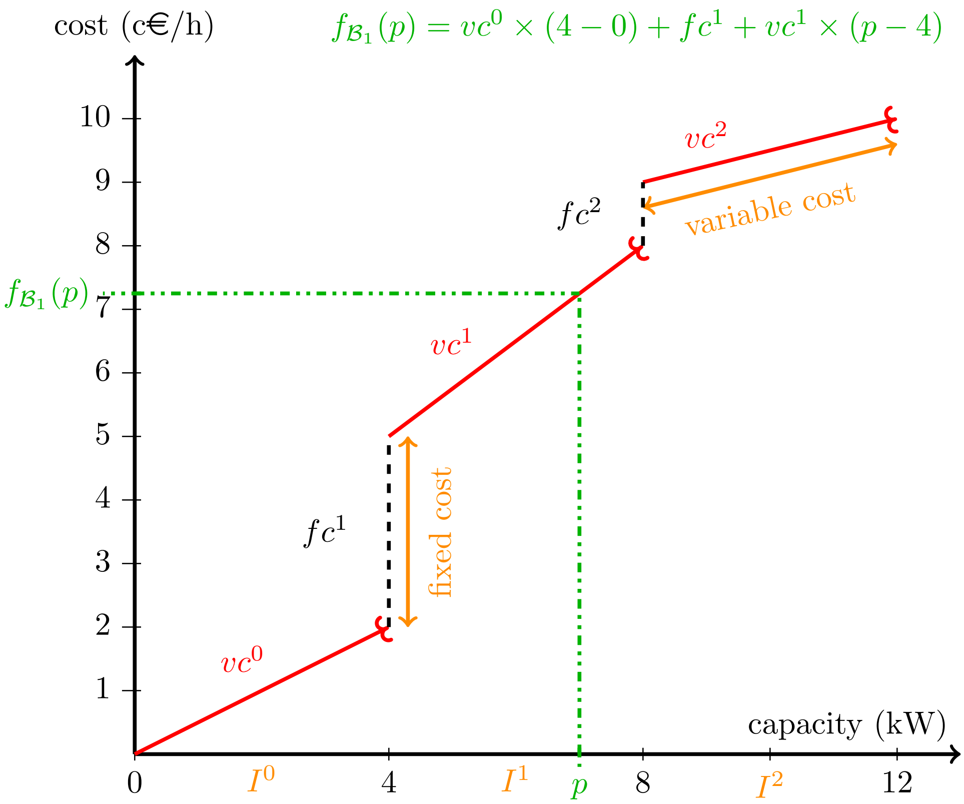


FIGURE 3.8 – Energy cost function of the time bucket $\mathcal{B}_1$.

Example of cost function: Figure 3.8 gives an illustration of the piecewise-linear cost function $f_{\mathcal{B}_1}$ for a fixed bucket $\mathcal{B}_1$. In this example, power capacity is partitioned in a uniform way, every 4 kW. The value of $f_{\mathcal{B}_1}(p)$ is computed by looking over each power capacity interval, such that p is lower or equal the maximum capacity of the interval: I^0 and I^1. For each of those capacity intervals, the related cost is given by a fixed cost plus a linear function of the power exceeding the minimal capacity of the interval. For example, the cost induced by the interval I^1 is given by $fc^1 + vc^1 \times (p-4)$. Finally, the obtained value $f_{\mathcal{B}_1}(p)$ is a cost in Euro cents per hour, that has to be integrated over time in order to obtain the total cost in Euros over a given time period.

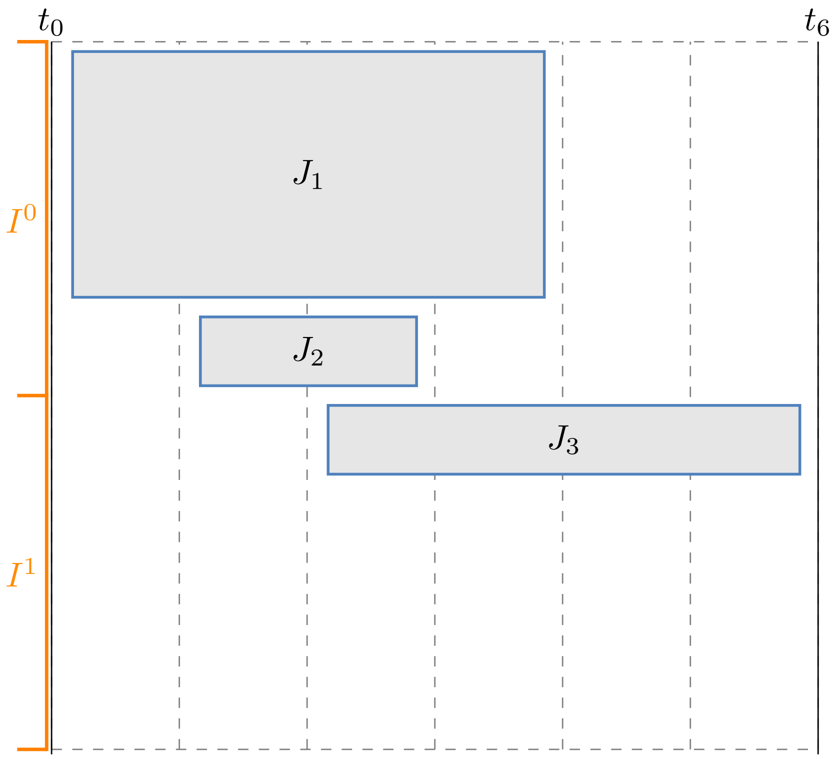


FIGURE 3.9 – Example of schedule.

Example of a schedule cost computation: The schedule given in Figure 3.9 consumes 3 kW between t_0 and t_1. With the cost function illustrated in Figure 3.8, that would cost $vc^0 \times 3 \times (t_1 - t_0) = 0.5 \times 3 \times 1 = 1.5$ c€. Moreover, between t_2 and t_3, the 5 kW power consumed would induce an electricity cost of $(vc^0 \times 4 + fc^1 + vc^1 \times 1) \times (t_3 - t_2) = (0.5 \times 4 + 3 + 0.75 \times 1) \times 1 = 5.75$ c€. The electricity cost of the whole schedule can be computed in the same way, looking over every time interval.

Graham's Notation

This problem is a generalized flow-shop problem with precedence, due dates, time-lags, and an energy resource. The objective considered is a trade-off between tardiness cost, storage (earliness) cost and energy cost. A convenient and classical notation for scheduling problems is the Graham's notation scheme introduced in (Graham et al., 1979). This notation denote scheduling problems as $\alpha \mid \beta \mid \gamma$, where the α field describes the machine environment, β the other constraints and γ the objective function. In (Zhang, 2010), a survey of complexity results for scheduling problems with time-lags is given, and the notation l_j for time lags in the β field of the Graham's notation is introduced. Finally, let us note $\mathcal{S}$ the storage costs and $\mathcal{E}$ the energy costs. Thus, our problem Π may be described as follows:

$$\Pi : F \mid res1, \text{in-tree}, d_j, l_j \mid \sum_j (w_j^T T_j) + \mathcal{S} + \mathcal{E}$$

We also address the preemptive case and note Π' the problem Π in which preemption is allowed.

TABLE 3.1 – A brief state of the art related to flow-shop scheduling with piecewise-linear electricity cost functions.

Flow-shop, lot-sizing, RCPSP	Linear energy cost functions and constraints	Piecewise-linear energy cost functions
(Graves, 1981; Brucker and Knust, 1999; Gonzalez and Sahni, 1978; Brucker, Drexl, et al., 1999; T. J. R. Johnson, 1967; Herroelen, De Reyck, and Demeulemeester, 1998; Sampson and Weiss, 1993; Palpant, Artigues, and Michelon, 2004; Pesek, Schaerf, and Žerovnik, 2007; Danna, 2004)	(Erschler, Lopez, and Thuriot, 1991; Le Pape, 1994; Baptiste, Le Pape, and Nuijten, 2001; Artigues, Lopez, and Haït, 2009; Artigues, Lopez, and Haït, 2013; Nattaf, Artigues, and Lopez, 2015; Nattaf, Artigues, Lopez, and Rivreau, 2016; Masmoudi et al., 2015; German, Desdouits, and Le Pape, 2015; Pecero et al., 2012)	(Ngueveu, Artigues, and Lopez, 2016)

3.3.2 Related Work

Flow-shop scheduling and lot-sizing problems have been broadly studied in the literature. In (Graves, 1981), a review of production scheduling is given, covering many aspects of our problem Π. More recent results are given in (Brucker and Knust, 1999), $\mathcal{NP}$-hardness is proven for several previously opened flow-shop problems. In particular, $F2|p_{ij} = 1, chains| \sum_j(T_i)$ is proven being $\mathcal{NP}$-hard. Since the problem Π is a generalization of the previous problem, Π is also $\mathcal{NP}$-hard. Notice that Π', where the preemption is allowed for Π, remains hard (see (Gonzalez and Sahni, 1978)). One could also consider Π as a RCPSP with unitary capacity resources (machines), a in-tree precedence graph and a more complex objective function. Let us point out (Brucker, Drexl, et al., 1999) for a survey about RCPSP.

Electricity-aware scheduling is an emerging trend. If one wants to participate in the demand-response market, one can consider maximum power-peak capacity, or electricity cost functions, both varying over time. In the first case, the key point is to manage electricity consumption to respect capacity constraints. The relevant notion of energetic reasoning was introduced in scheduling in the 1990s, in (Erschler, Lopez, and Thuriot, 1991; Le Pape, 1994; Baptiste, Le Pape, and Nuijten, 2001), and often used to strengthen the analysis of non-energetic resource constraints. Related research works are discussed below.

In (Erschler, Lopez, and Thuriot, 1991), a MILP formulation of the overlap of tasks on time-intervals is proposed.

In (Artigues, Lopez, and Haït, 2009; Artigues, Lopez, and Haït, 2013), the authors introduce the Energy Scheduling Problem (EnSP) where activities are defined by their required energy and minimum and maximum resource requirements. The problem consists in finding, for each activity, a starting and completion times but also a power allocation that can vary over time. The objective considered is the sum of the linear energy costs and power overrun costs. A two-step approach to solve the problem is given, composed of a time-based MILP model and a constraint propagation technique.

In (Nattaf, Artigues, and Lopez, 2015) and (Nattaf, Artigues, Lopez, and Rivreau, 2016), this problem is extended with time windows and linear efficiency functions. Mixed Integer Linear Programs, satisfiability tests and a hybrid branch-and-bound method are proposed to solve the problem.

Another extension of the EnSP, given in (Ngueveu, Artigues, and Lopez, 2016), consists in considering preemptive tasks with time windows and non-linear efficiency functions. They show that their problem is $\mathcal{NP}$-hard. The authors introduce a non-linear (due to the efficiency function) mixed integer formulation and an extended MILP formulation based on feasible subsets. A Branch&Price procedure, as well as a piecewise-linear lower and upper bounding framework are proposed. Piecewise-linear electricity tariff functions can be modeled as non-linear efficiency functions.

When the key point is no more the electricity capacity, but the electricity cost, complex electricity cost functions have to be considered.

In (Masmoudi et al., 2015), a short overview of energy-related objectives in lot-sizing and flow-shops problems is given, but only peak/off-peak electricity cost functions are considered.

In the same way, in (German, Desdouits, and Le Pape, 2015), a slightly different version of problem Π is studied: the time horizon is divided into several time buckets with a linear electricity cost function associated with each bucket. The authors also use a local search method to solve the problem and give a MIP formulation used as one of the local search operators.

On our side, a piecewise-linear electricity cost function, varying over time is considered, as explained in subsection 3.3.1.

In all the previously cited research works, problems are solved using MILP (sometimes coupled with Constraint Satisfaction Problems). In order to solve our complex industrial problem, we are also interested in local search techniques. The well-known RCPSP (defined in (T. J. R. Johnson, 1967)) minimizes the duration of the project subject to simple precedence constraints and availability constraints on renewable (but non-energetic) resources. This problem is NP-difficult and many approaches were tried to solve it in exact or approximated ways (Herroelen, De Reyck, and Demeulemeester, 1998). It has been treated with a lot of different local search techniques: see for instance (Sampson and Weiss, 1993; Palpant, Artigues, and Michelon, 2004; Pesek, Schaerf, and Žerovnik, 2007). A problem similar to ours (but with no energy resource) has been studied in (Danna, 2004) with a hybrid algorithm composed of mixed integer programming and local search techniques. This paper has provided some results on a number of public benchmark instances. In (Pecero et al., 2012), a local search algorithm is proposed on a similar scheduling problem as the one in (Artigues, Lopez, and Haït, 2009), yet with precedence constraints and with the goal of minimizing the energy consumption while preserving a given schedule performance (makespan). We inspired of those papers to implement a local search algorithm, that uses a MILP operator.

Table 3.1 summarizes the brief state of the art presented above.

3.4 Efficient Methods to Find Production Plans

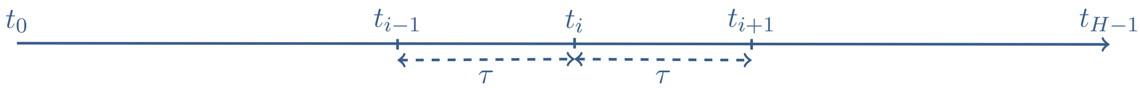
The scheduling problem introduced in the previous section is NP-difficult and very hard to solve optimally in practice. Thus we propose three different Mixed-Integer Linear formulations; each of them approximating one or several difficult notions of the original problem. This way, a plant manager could compare several production plans, computed in limited time with our methods, and choose the solution he prefers.

The following subsections are devoted to the presentation of three Mixed-Integer Linear formulations: the first one is time-independent, and models only a linear electricity cost function; the second one is time-based, non-preemptive and precedence-oriented; and the third one is time-based, preemptive and storage-oriented. In each case, we have to determine the start time of every scheduled activity in order to minimize the resource, tardiness and storage costs.

Definition 22. *In all formulations, the considered time horizon T is discretized such that:*

- *τ is the time step (ex: 10mn)*

- *H is the number of time steps considered (ex: 60)*
- *$T = \{t_0, t_0 + \tau, \dots, t_0 + (H-1) \times \tau\}$ is the set of instants considered*
- *$t \in T$ is a given time instant considered (ex: 6h45)*



3.4.1 Overlap MILP

In this subsection, we limited ourselves to the case in which the resource cost is, over each time interval, linear. This restriction allows us to model linear electricity costs (like peak/off-peak tariffs) easily, with a time-independent formulation, using previous results from (Erschler, Lopez, and Thuriot, 1991). Let the decision variables of this formulation be:

$\mathbf{st_i}$	$\forall a_i \in A$	the start time of the activity a_i
$\mathbf{et_i}$	$\forall a_i \in A$	the completion time of the activity a_i
$\mathbf{tard_d}$	$\forall dem_d \in D$	the (positive) amount of time between the due date of demand dem_d and its delivery date (0 if the demand is delivered before or at the due date)
$\mathbf{x_{i,j}}$	$\forall (a_i, a_j) \in A^2 \mid res_i = res_j \land i < j$	boolean variable equal to 1 if the activity a_i is before a_j
$\mathbf{overlap_{i,l}}$	$\forall a_i \in A$ $\forall \mathcal{B}_l \in \mathcal{B}$	real amount of time during which activity a_i overlaps the time bucket $\mathcal{B}_l$
$\mathbf{overlapMax_{i,l}}$	$\forall a_i \in A$ $\forall \mathcal{B}_l \in \mathcal{B}$	an auxiliary variable representing an upper bound of the variable $overlap_{i,l}$ when the overlap is non-null (can be negative otherwise)
$\mathbf{stock_m}$	$\forall mat_m \in M$	the total amount of time during which one unit of material mat_m is stored
$\mathbf{pos_{i,l}}$	$\forall a_i \in A$ $\forall \mathcal{B}_l \in \mathcal{B}$	boolean variable equal to 1 if the activity a_i overlaps the bucket $\mathcal{B}_l$

First, let us introduce the classical constraints ruling the start and end times of the activities.

$$\forall a_i \in A, \qquad \mathbf{et_i} - \mathbf{st_i} = pt_i \tag{3.11}$$
$$\forall (a_i, a_j) \in \mathcal{A}, \qquad \mathbf{st_j} - \mathbf{st_i} \geq dmin_i \tag{3.12}$$

Equation (3.11) ensures that an activity duration is exactly equal to its required processing time. While Equation (3.12) enforces a minimum delay between two activities linked by a precedence arc.

In order to ensure a machine is only used by one activity at a time, the following disjunctive constraints (coming from (Applegate and Cook, 1991)) are used.

$$\forall (a_i, a_j) \in A^2 | res_i = res_j \land i < j, \qquad \mathbf{et}_i \leq \mathbf{st}_j + M \times (1 - \mathbf{x}_{i,j}) \tag{3.13}$$
$$\mathbf{et}_j \leq \mathbf{st}_i + M \times \mathbf{x}_{i,j} \tag{3.14}$$

If two activities are supposed to be executed on the same machine, then one of them has to be scheduled sufficiently before the other one.

Each customer demand has an associated delivery tardiness that is the positive amount of time between the demand due date and the end of the last activity of the associated job.

$$\forall dem_d \in D, \qquad \mathbf{tard_d} \geq \mathbf{et_i} - duedate_d \mid a_i = a_6^d \tag{3.15}$$

$$\mathbf{tard}_d \geq 0 \tag{3.16}$$

Each material is stored after having been produced and before having been consumed. In this formulation, we suppose that all activities have the same production and consumption rate (which is not necessarily true but simplifies the modeling). Then the amount of time, during which one unit of material mat_m is stored, is modeled in the following way.

$$\forall mat_m \in M, \qquad \mathbf{stock_m} = \sum_{(a_i,a_j)\in\mathcal{A}|mat_i=mat_m} (\mathbf{st_j} - \mathbf{st_i}) \times q_i \tag{3.17}$$

Finally, the electricity cost is modeled according to (Erschler, Lopez, and Thuriot, 1991). The principle is to compute the amount of time each activity overlaps each bucket. With this information, it is possible to compute the electricity cost, if we suppose that the cost function is linear in each bucket. That does not model the whole complexity of the problem considered in this chapter, but allows this formulation to remain time-independent. The overlap of each activity on each time bucket is defined as: $overlap_{i,l} = max(0, min(\mathcal{B}_l^{sup} - \mathcal{B}_l^{inf}, \mathcal{B}_l^{sup} - \mathbf{st}_i, \mathbf{et}_i - \mathcal{B}_l^{inf}, \mathbf{et}_i - \mathbf{st}_i))$. This equation represents the four cases when the activity is processed outside the bucket (before or after), inside it and when they partially or totally overlap. This (non-linear) equation is modeled with the following linear equations.

$$\forall a_i \in A, \qquad \sum_{\mathcal{B}_l\in\mathcal{B}} \mathbf{overlap}_{i,l} = \mathbf{et}_i - \mathbf{st}_i \tag{3.18}$$

$$\forall a_i \in A, \forall \mathcal{B}_l \in \mathcal{B}, \qquad \mathbf{overlap}_{i,l} \leq \mathbf{pos}_{i,l} \times (\mathcal{B}_l^{sup} - \mathcal{B}_l^{inf}) \tag{3.19}$$

$$\mathbf{overlap}_{i,l} \leq \mathbf{overlapMax}_{i,l} + (t_H - t_0) \times (1 - \mathbf{pos}_{i,l}) \tag{3.20}$$

$$\mathbf{overlapMax}_{i,l} \leq \mathcal{B}_l^{sup} - \mathcal{B}_l^{inf} \tag{3.21}$$

$$\mathbf{overlapMax}_{i,l} \leq \mathcal{B}_l^{sup} - \mathbf{st}_i \tag{3.22}$$

$$\mathbf{overlapMax}_{i,l} \leq \mathbf{et}_i - \mathcal{B}_l^{inf} \tag{3.23}$$

$$\mathbf{overlapMax}_{i,l} \leq \mathbf{et}_i - \mathbf{st}_i \tag{3.24}$$

$$\mathbf{overlap}_{i,l} \geq 0 \tag{3.25}$$

Equation (3.18) ensures that the overlap of a given activity over all buckets is equal to the activity processing time. Equation (3.19) states that if Activity a_i does not overlap bucket $\mathcal{B}_l$, then $\mathbf{overlap}_{i,l} = 0$. Equation (3.20) bounds $\mathbf{overlap}_{i,l}$ by $\mathbf{overlapMax}_{i,l}$, iff Activity a_i overlaps bucket $\mathcal{B}_l$. Finally, Equations (3.21) - (3.24) bound $\mathbf{overlapMax}_{i,l}$ by the real (possibly negative) amount of time Activity a_i overlaps Bucket $\mathcal{B}_l$.

Figure 3.10 shows an example with one activity and three time buckets. Activity a_1 actually overlaps Buckets $\mathcal{B}_0$ and $\mathcal{B}_1$. As the activity begins during $\mathcal{B}_0$ and is completed at the end of $\mathcal{B}_1$, $overlapMax_{1,0}$ is positive and bounds $overlap_{1,0}$. In the same way, $overlapMax_{1,1}$ is positive and bounds $overlap_{1,1}$. On the other hand, as Activity a_1 does not cross Bucket $\mathcal{B}_2$, $overlapMax_{1,2}$ is negative and $pos_{1,2} = 0$ bounds $overlap_{1,2}$.

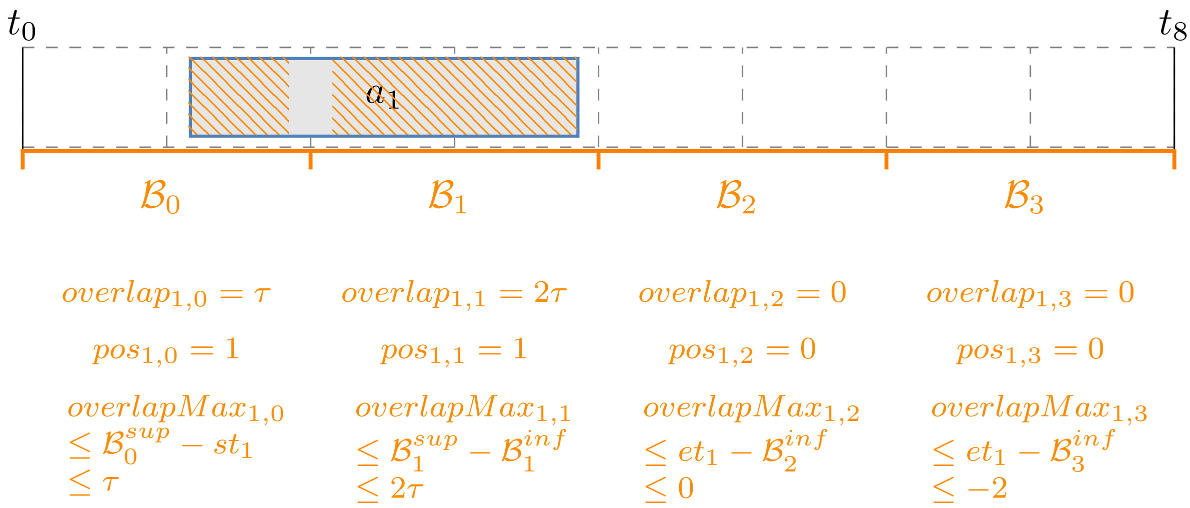


FIGURE 3.10 – Computation of $overlap$ between activity and time buckets.

Then, the objective function is given by:

$$\sum_{dem_d \in D} \mathbf{tard}_d \times tc_d + \sum_{mat_m \in M} \mathbf{stock}_m \times sc_m + \sum_{a_i \in A} \sum_{\mathcal{B}_l \in \mathcal{B}} p_i \times vc_l^1 \times \mathbf{overlap}_{i,l} \tag{3.26}$$

Note that we can compute the electricity cost looking independently at each activity only because of the simplifying assumption about the linear cost function.

The whole formulation is given by:

$$\begin{aligned} \min \quad & (3.26) \\ \text{s.t.} \quad & (3.11) - (3.20) \end{aligned}$$

This formulation has a big advantage: it is time-independent; but the storage and energy objectives are approximated (except, of course, respectively if all activities have the same production and consumption rates or if the tariff is truly linear over each bucket).

3.4.2 Time-Based Precedence-Oriented MILP

This formulation is inspired by a classical time-based scheduling formulation (see (Pritsker, Waiters, and Wolfe, 1969)) that uses time-indexed boolean variables giving the starting time of each activity. This formulation does not allow preemption and thus addresses Problem II. There are classical: precedence constraints with minimum delays, tardiness constraints and tardiness cost. Moreover, the storage constraints used give an approximation of the storage cost. Indeed, for each precedence arc (a_i, a_j), the whole material quantity is considered stored between the st_i and st_j. This is exact if and only if the production rate and consumption rate of both activities are the same. Otherwise this is an approximation of the storage costs that allows us to store material only if it is profitable regarding the others costs.

Finally, modeling piecewise-linear energy costs with a linear program was a challenge. Note that the electricity cost depends on the power consumption of all activities at the same time, thus this is a coupling constraint.

Let the decision variables of this formulation be:

$\mathbf{y_{i,t}}$	$\forall a_i \in A$, $\forall t \in T$	a boolean variable that is worth 1 if the activity a_i starts at t, 0 otherwise
$\mathbf{et_i}$	$\forall a_i \in A$	the completion time of the activity a_i
$\mathbf{stock_m}$	$\forall mat_m \in M$	the total amount of time during which one unit of material mat_m is stored
$\mathbf{tard_d}$	$\forall dem_d \in D$	the (positive) amount of time between the due date of demand dem_d and its delivery date
$\mathbf{power_t}$	$\forall t \in T$	the power consumption of the whole production plan at time t
$\boldsymbol{\beta_{t,h}}$	$\forall t \in T$ $\forall I^h \in I_l \vert$ $t \in \mathcal{B}_l$	a boolean variable equals to 1 if the power capacity interval I^h is used at time t
$\boldsymbol{\alpha_{t,h}}$	$\forall t \in T$ $\forall I^h \in I_l \vert$ $t \in \mathcal{B}_l$	a real variable equals to the power value used in the capacity interval I^h at time t

A feasible non-preemptive solution, regarding $\mathbf{y_{i,t}}$ variables, is defined by the following equations:

$$\sum_{t \in T} \mathbf{y_{i,t}} = 1 \qquad \forall a_i \in A \tag{3.27}$$

$$\sum_{t \in T} t \times (\mathbf{y_{j,t}} - \mathbf{y_{i,t}}) \geq dmin_i \qquad \forall (a_i, a_j) \in \mathcal{A} \tag{3.28}$$

$$\sum_{a_i \in A | res_i = r} \sum_{t' = t^{inf}}^{t} \mathbf{y_{i,t'}} \leq 1 \qquad \forall r \in R, \forall t \in T \tag{3.29}$$

$$\mathbf{et_i} - \sum_{t \in T} (\mathbf{y_{i,t}} \times t + \mathbf{y_{i,t}} \times pt_i) = 0 \qquad \forall a_i \in A \tag{3.30}$$

Equation (3.27) is a classical assignment constraint that ensures activities can start only once. Equation (3.28) defines start-to-start precedence constraints with minimum delay. Equation (3.29) is the disjunctive constraint that ensures every machine is used by at most one activity at the same time. In this constraint, $t^{inf} = \lfloor \frac{t - pt_i}{\tau} \rfloor \times \tau + 1$ is the first time step where a_i can have begun and still being running at time t. Finally, Equation (3.30) sets the completion time of each activity, regarding their starting time.

Then, the following constraint sets $\mathbf{stock_m}$ to the amount of storage used over the whole time horizon.

$$\mathbf{stock_m} = \sum_{(a_i, a_j) \in \mathcal{A} | mat_i = mat_m} \sum_{t} \left[t \times \tau \times (\mathbf{y_{j,t}} - \mathbf{y_{i,t}}) \times q_i \right] \qquad \forall mat_m \in M \tag{3.31}$$

Equation (3.31) allows us to recover: the idle time between the starting time of activity a_i (that produces material mat_m) and the starting time of activity a_i (that consumes it), multiplied by the quantity of material mat_m produced. Note that this is an approximation of the real storage duration of mat_m, since that supposes activities have the same consumption and production rate.

The following classical tardiness constraints ensure that $\mathbf{tard_d}$ is the positive tardiness of demand dem_d, regarding its due date.

$$\forall dem_d \in D \qquad \mathbf{tard_d} \geq \mathbf{et_i} - duedate_d \mid a_i = a_6^d \tag{3.32}$$

$$\mathbf{tard_d} \geq 0 \tag{3.33}$$

Equation (3.32) ensures that the tardiness of a given demand dem_d is greater than the amount of time between its due date and the completion time of the last activity of the job j_d. Equation (3.33) ensures that this tardiness is positive.

Equation (3.34) computes the instantaneous power used by the whole set of activities, at every time step.

$$\mathbf{power_t} = \sum_{a_i \in A} \sum_{t'=t^{inf}}^{t} p_i \times \mathbf{y_{i,t'}} \qquad \forall t \in T \tag{3.34}$$

Previous time-steps have to be looked over in order to determine if an activity is running at time t, following the same principles as Equation (3.29).

Now, let us introduce the constraints defining the piecewise-linear energy cost function, as described in subsection 3.3.1.

$$\forall t \in T, \forall I^h \in I_l | t \in \mathcal{B}_l,$$

$$\boldsymbol{\beta_{t,h}} \geq \frac{\mathbf{power_t} - cmin^h}{p_{max}} \tag{3.35}$$

$$\boldsymbol{\alpha_{t,h}} \leq (cmax^h - cmin^h) \times \boldsymbol{\beta_{t,h}} \tag{3.36}$$

$$\boldsymbol{\alpha_{t,h}} \geq (cmax^h - cmin^h) \times \boldsymbol{\beta_{t,h+1}} \tag{3.37}$$

$$\forall t \in T, \qquad \mathbf{power_t} = \sum_{I^h \in I_l | t \in \mathcal{B}_l} \boldsymbol{\alpha_{t,h}} \tag{3.38}$$

Equation (3.35) forces $\boldsymbol{\beta_{t,h}}$ to 1 when the power consumed at time t is greater than the minimal capacity of the interval. Equation (3.37) ensures that if the capacity interval I^{h+1} is (partially) covered, then the capacity interval I^h is entirely covered. Equation (3.36) ensures that if the capacity interval I^h is (partially) covered, then $\boldsymbol{\beta_{t,h}}$ is worth 1. Equation (3.38) ensures that the amount of capacity intervals covered at time t equals to the power consumed at the same time.

Figure 3.11 gives an example of how those variables behave while computing the electricity cost of a given schedule. In this example, there are three capacity intervals I^0, I^1, I^2, each of them with a capacity of two units of power. There are also four different activities with a power consumption represented on the y-axis. The x-axis represents time intervals during which the electricity cost is evaluated. We can see on the figure that between t_0 and t_1, three power units are consumed by activities a_1 and a_2. Thus, the two first capacity intervals I^0 and I^1 are (partially) covered and $\beta_{0,1} = \beta_{1,1} = 1$. The corresponding α variables contain the amount of the interval covered.

Finally, the objective function of this time-based precedence-oriented formulation is defined below.

Energy objective

$$\sum_{t \in T} \sum_{I^h \in I_l | t \in \mathcal{B}_l} fc^h \times \boldsymbol{\beta_{t,h}} + vc^h \times \boldsymbol{\alpha_{t,h}} \tag{3.39}$$

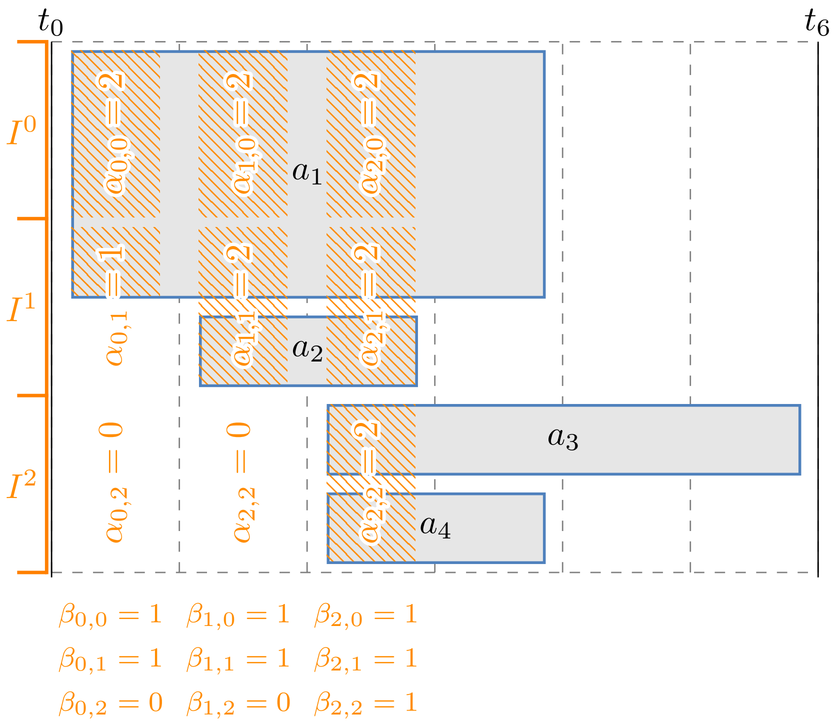


FIGURE 3.11 – Integer variables to compute the piecewise-linear electricity cost of a given schedule.

Tardiness objective

$$\sum_{d} \mathbf{tard_d} \times tc_d \tag{3.40}$$

Storage objective

$$\sum_{m} \mathbf{stock_m} \times sc_m \tag{3.41}$$

Then, the overall MILP formulation is given by:

$$\begin{aligned} \text{min} \quad & (3.39) + (3.40) + (3.41) \\ \text{s.t.} \quad & (3.27) - (3.38) \end{aligned}$$

This formulation is time-based. That induces several difficulties: choosing an appropriate time-step for the instance and a big number of variables and constraints when the time-step is small and the time horizon is big. Conversely, choosing a large time step leads to a pessimistic formulation of the resource constraint and energy cost. But having a time-based formulation is the only way we found to allow the formulation of a piecewise-linear energy cost function. However, let us note that the linear relaxation of a time-based formulation is supposed to be better than the linear relaxation of a time-independent formulation. In this precedence-oriented formulation, the storage objective remains an approximation.

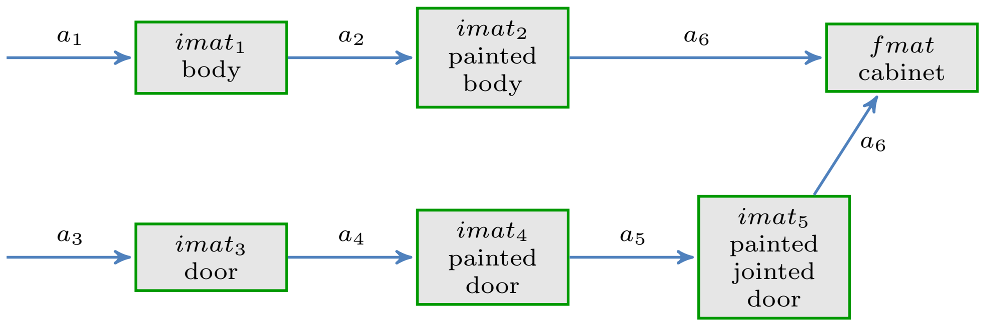


FIGURE 3.12 – Dual graph of the precedence graph from Figure 3.7.

3.4.3 Time-Based Storage-Oriented MILP

This formulation allows preemption and is driven by the amount of materials in storage. The key point is that a precedence link between activities is equivalent to a shared storage of material. Indeed, for each precedence arc $(a_i, a_j) \in \mathcal{A}$, a_i has to begin to fill the stock of mat_i before a_j could be scheduled. In fact, managing stock levels is somehow the dual point of view of ensuring precedence, as the materials graph is the dual graph of the precedence graph.

Definition 23. *Let G' be the dual graph of G where nodes are materials and arcs are activities.*

Figure 3.12 shows, as an example, the dual graph of the precedence graph of a job, shown on Figure 3.7. The nodes are stocks of material and the arcs are activities that produce and consume materials.

Now, let us show how precedence constraints are replaced by storage constraints in this formulation. The decision variables given below are the ones that are different from the previous formulation. Let:

$\mathbf{x_{i,t}}$	$\forall a_i \in A,$ $\forall t \in T$	be a boolean that is equal to 1 if a_i is running between t and $t + \tau$
$\mathbf{dur_{i,t}}$	$\forall a_i \in A,$ $\forall t \in T$	be the duration of a_i that covers $[t, t+\tau]$
$\mathbf{z_{d,t}}$	$\forall dem_d \in D,$ $\forall t \in T \cup \{H \times \tau\}$	be a boolean that is equal to 1 if demand dem_d is supplied at time t
$\mathbf{stock_{m,t}}$	$\forall mat_m \in M,$ $\forall t \in T \cup \{H \times \tau\}$	be the amount of material mat_m stored at time t

Moreover, $\mathbf{tard_d}$, $\mathbf{power_t}$, $\boldsymbol{\alpha_{t,h}}$ and $\boldsymbol{\beta_{t,h}}$ are defined in the same way as in Subsection 3.4.2.

As this formulation allows preemption, consistency between x boolean variables and dur amounts of time is ensured by the following equations.

$$\mathbf{dur_{i,t}} \leq \tau \times \mathbf{x_{i,t}} \qquad \forall a_i \in A, \forall t \in T \tag{3.42}$$

$$\mathbf{dur_{i,t}} \geq \mathbf{x_{i,t}} \qquad \forall a_i \in A, \forall t \in T \tag{3.43}$$

$$\sum_{t \in T} \mathbf{dur_{i,t}} = pt_i \qquad \forall a_i \in A \tag{3.44}$$

Equation (3.42) ensures that $\mathbf{x_{i,t}} = 0$ implies $\mathbf{dur_{i,t}} = 0$ and bounds $\mathbf{dur_{i,t}} = 0$ by τ, while Equation (3.43) ensures that $\mathbf{dur_{i,t}}$ is not null if $\mathbf{x_{i,t}}$ is not null. Equation (3.44) ensures that an activity is executed during its required processing time. Note that a solution of this formulation does not give starting and completion time of each activity: given the duration of an activity over a time-step, its starting time has to be decided. For that purpose, we implemented a simple heuristic, presented at the end of this subsection.

Disjunctive constraints relative to machines unitary capacity are stated in Equation (3.45).

$$\sum_{a_i \in A | res_i = r} \mathbf{x_{i,t}} \leq 1 \qquad \forall r \in R, \forall t \in T \tag{3.45}$$

Note that the disjunction concerns the whole time-step even if two activities could be scheduled one after the other in the same time-step. It would be worthwhile to consider a relaxation of this constraint using $\mathbf{dur_{i,t}}$ variables. But in that case, the heuristic to compute starting times from $\mathbf{dur_{i,t}}$ variables would be more complex.

Delivery and tardiness constraint are re-formulated as below.

$$\forall dem_d \in D,$$

$$\sum_{t \in T \cup \{H\}} \mathbf{z_{d,t}} = 1 \tag{3.46}$$

$$\sum_{\forall t \in T \cup \{H\}} (t \times \mathbf{z_{d,t}}) - \mathbf{tard_d} = duedate_d \tag{3.47}$$

$$\mathbf{tard_d} \geq 0 \tag{3.48}$$

Equation (3.46) ensures each demand is delivered once. Equations (3.47) and (3.48) ensures delivery is after due date and tardiness is consistent.

The key point of this formulation is the following flow-like equations that regulate material storage. The idea is that the quantity of a material mat_m stored at time t is given by the quantity of the same material stored at time $t - \tau$, minus the quantity consumed between $t - \tau$ and t, plus the quantity produced between $t - \tau$ and t. In order to compute those quantities, one has to look over each precedence arc related to mat_m. The time needed for activity a_i to produce one unit of material is given by vpt_i. Then the quantity of material produced by an activity a_i is given by: $\frac{\mathbf{dur_{i,t-\tau}}}{vpt_i}$.

Example: Figure 3.13 shows the evolution of the stock of the intermediate material $imat_2$ over six time periods. At time t_0, the stock is empty. Then, activity a_2 begins to produce $imat_2$. At time t_1, the stock grows of $\frac{\mathbf{dur_{2,0}}}{vpt_2} = 0.5 \times 4 = 2$. During the two next time periods, the stock grows of $1/vpt_2 = 4$ units. But at time t_3, activity a_6 begins to consume $imat_2$. Thus at time t_4, the stock has 6 units because it has been increased of $1/vpt_2 = 4$ units and decreased of $1/vpt_6 = 8$ units.

This reasoning works for every precedence arc without a minimum delay requested. But when there is a minimum delay $dmin_i$ on a precedence arc (a_i, a_j), the time delay has to be taken into account in the flow-like equation (with an additional time dimension). For that purpose, another virtual material mat_i^{fict} has to be created with a virtual producing activity of duration $dmin_i$. Indeed, for a given time-step t, the stock of mat_i (resp. mat_i^{fict}) is decreased (resp. increased) of what was produced by activity a_i at time-step $t^{delay} - \tau$, with $t^{delay} = \lfloor \frac{t - dmin_i}{\tau} \rfloor \times \tau$.

Example: An example of minimum delay is given in Figure 3.14, with activities a_2 and a_6 linked by a precedence arc with a minimum delay of 3τ. It means that each piece of material produced by a_2 has to wait at least 3τ time units before being consumed by a_6. Like in the previous example, the rounded rectangles at the bottom of the figure represent the evolution of the stock variables. The first row of stock variables corresponds to the stock of material $imat_2$ varying over time. The second row is for the stock of material $imat_2^{fict}$. The arrows that go

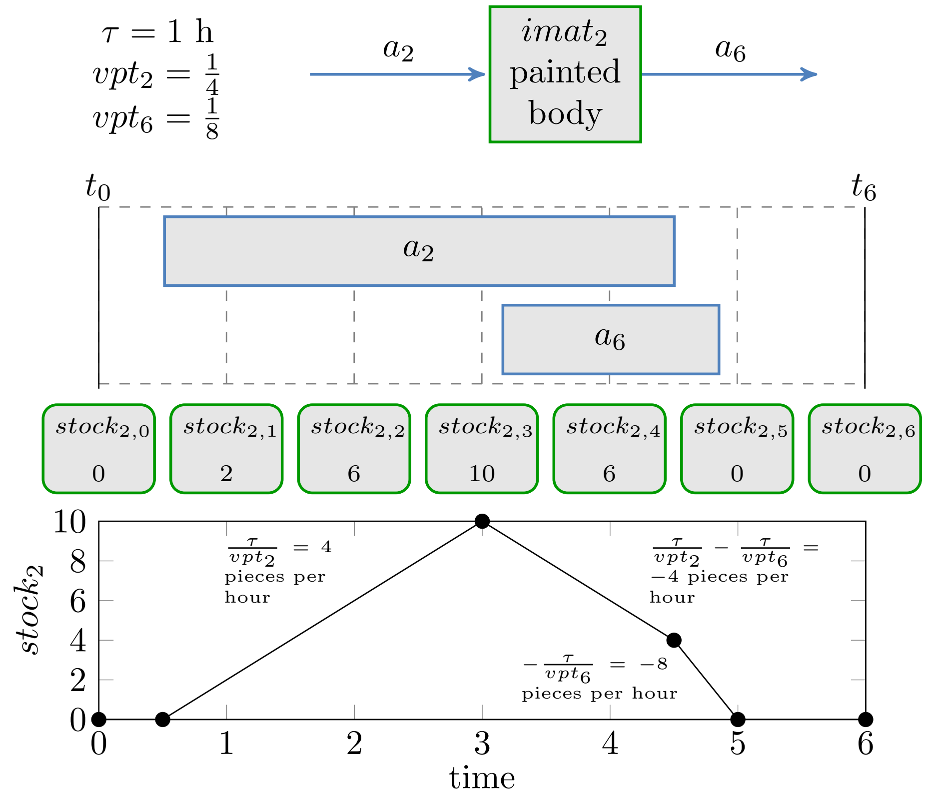


FIGURE 3.13 – Evolution of the stock, depending on consumption and production rates.

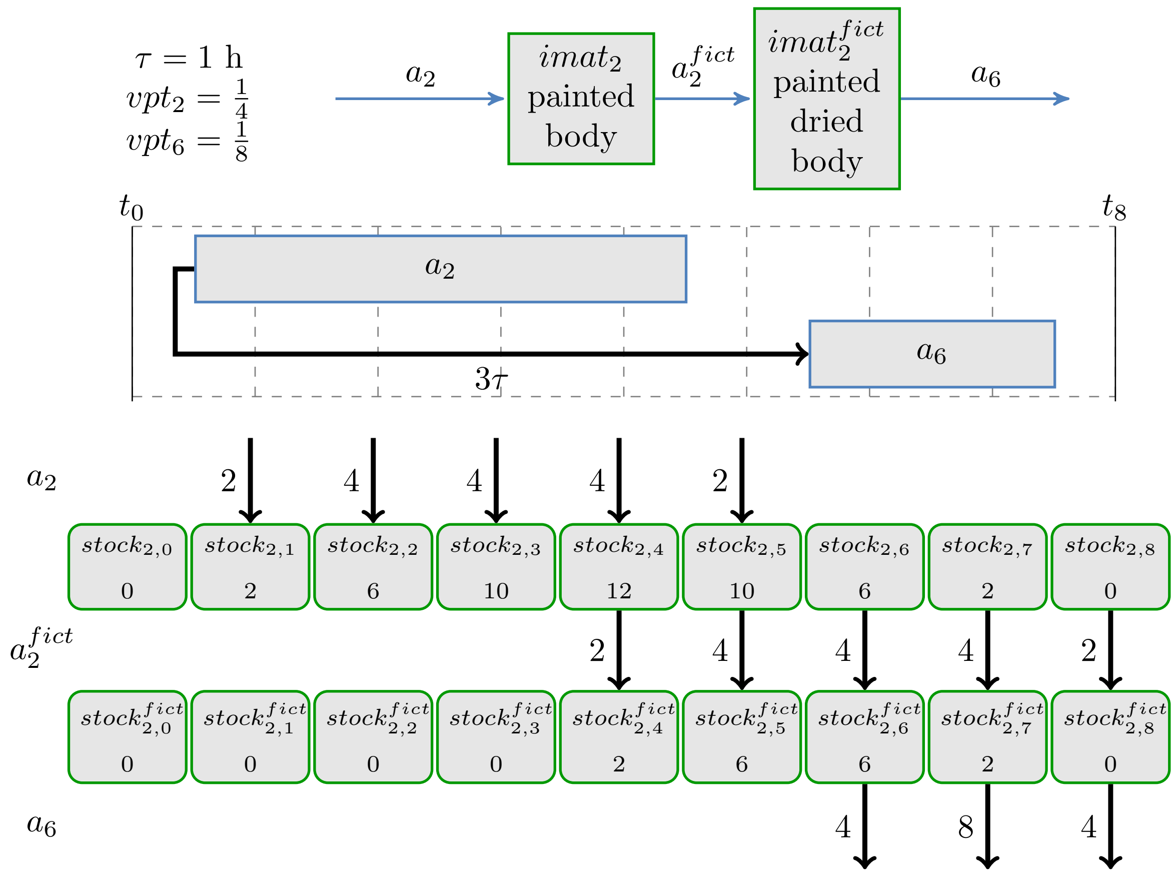


FIGURE 3.14 – Evolution of the stocks, when there is a minimum delay.

to the first raw of stock variables indicate the amount of material $imat_2$ produced by activity a_2. The arrows between stocks of $imat_2$ and stocks of $imat_2^{fict}$ correspond to pieces of material for which to delay $dmin_2$ has passed. Finally, the arrows coming from the second raw of stock variables indicate the amount of material $imat_2$ consumed by activity a_6. During the period $[t_0; t_1]$, a_2 produces two pieces of material $imat_2$, that fill the stock of $imat_2$ at time t_1. These two pieces are removed from the stock 3τ later, at time t_4, to fill the stock of material $imat_2^{fict}$. Since then, these two pieces are available to be consumed by activity a_6, and so on.

Now, let us introduce the previously described stock equations.

$$\begin{aligned}\forall mat_m \in M, \forall t \in T \cup \{H \times \tau\}, \quad \mathbf{stock_{m,t}} &= \mathbf{stock_{m,t-\tau}} \\ &+ \sum_{a_i \in A | mat_i = mat_m} \left(\frac{\mathbf{dur_{i,t-\tau}}}{vpt_i} \right) \\ &- \sum_{dem_d \in D | fmat_d = mat_m} (\mathbf{z_{i,t}} \times q_d) \\ &- \sum_{(a_i,a_j) \in \mathcal{A} | mat_i = mat_m \wedge dmin_i = 0} \left(\frac{\mathbf{dur_{j,t-\tau}}}{vpt_j} \right) \\ &- \sum_{a_i \in A | mat_i = mat_m \wedge dmin_i > 0} \left(\frac{\mathbf{dur_{i,t^{delay}-\tau}}}{vpt_i} \right) \end{aligned} \tag{3.49}$$

Equation (3.49) regulates the stock of each material mat_m. Every activity a_i increases $stock_{i,t}$ of the quantity of $imat_i$ it can produce in $\mathbf{dur_{i,t-\tau}}$. Every demand that consumes $fmat_d$ decreases $stock_{d,t}$ of q_d units if it is delivered at time t. Every activity a_j decreases $stock_{i,t}$ of the quantity of mat_i it can consume in $\mathbf{dur_{j,t-\tau}}$, if there is no minimum delay on the arc (a_i, a_j). If there is a minimal delay on arc (a_i, a_j), then the material mat_i is automatically consumed $dmin_i$ time units after having being produced. The quantity automatically consumed increases another virtual stock introduced below.

$$\forall (a_i, a_j) \in \mathcal{A} | dmin_i > 0, \forall t \in T \cup \{H \times \tau\},$$
$$\mathbf{stock^{fict}_{i,t}} = \mathbf{stock^{fict}_{i,t-\tau}} + \frac{\mathbf{dur_{i,t^{delay}-\tau}}}{vpt_i} - \frac{\mathbf{dur_{j,t-\tau}}}{vpt_j} \tag{3.50}$$

Equation (3.50) regulates the stock of each fictive material mat_i^{fict} induced by every precedence arc (a_i, a_j) producing mat_i with a not null minimum delay. This virtual stock at time t is automatically increased from the quantity of mat_m produced by activity a_i at $t^{delay} - \tau$. This virtual stock is consumed normally by activity a_j.

Then, the total storage cost can be approximated in the following way.

$$\sum_{mat_m \in M} \sum_{t \in \{\tau, \ldots, H \times \tau\}} \mathbf{stock_{m,t}} \times \tau \times sc_m \tag{3.51}$$

The $stock_{m,t}$ variables correspond to the amount of stock at time t. In order to compute the storage cost between time t and time $t + \tau$, we have to consider that $stock_{m,t}$ pieces have been stored during the whole period $[t; t + \tau]$. This is not necessarily true and makes Equation (3.51) an approximation of the real storage cost.

Finally, Equation (3.52) computes the instantaneous power used by the whole set of activities, at every time step, while Equations (3.35) - (3.38) ensure the electricity cost is computed as in the previous formulation.

$$\mathbf{power_t} = \sum_{a_i \in A} p_i \times \mathbf{x_{i,t}} \qquad \forall t \in T \tag{3.52}$$

Note that this equation induces a pessimistic computation of the power consumed during a time-step as all activities are considered being simultaneous.

Then, the overall MILP formulation is given by:

$$\begin{aligned} \min \quad & (3.39) + (3.40) + (3.51) \\ \text{s.t.} \quad & (3.35) - (3.38) \ \&\ (3.42) - (3.52) \end{aligned}$$

With this formulation, we tried to find a way to compute storage costs in a more accurate way. However, there are many drawbacks with this formulation.

Firstly, a solution to this formulation does not give a schedule. Thus, a heuristic has to be used to compute starting and completion times of activities. We developed two heuristics: a simple greedy algorithm and a linear problem formulation. Moreover, precedence delays between activities are respected only when the time step used tends toward zero. In order to fix this issue, one time step (more) can be requested between two activities linked by a precedence arc. That would degrade solutions quality but would make them fully compliant with precedence constraints. Another solution would be to replace $x_{i,t}$ and $dur_{i,t}$ variables by $st_{i,t}$ and $et_{i,t}$ variables, stating for the starting and completion times of a_i on time step $[t, t+\tau]$. This solution would allow to remove integer variables but needs a lot more research time we do not currently have.

The second problem is there is no limit on activities preemption. Thus solutions found are not really usable in the plant. A solution could be to take into account machines setup costs, but new integer variables would be necessary.

The third problem is that electricity and storage costs are really approximated. Indeed for their computation, activities and stock behaviors are considered uniform over a given time step, which is not true.

For all those reasons, the storage-oriented formulation could not currently be used in a decision-aid tool. Thus, it is not included in the experiments.

3.4.4 Local Search Algorithm and Constraint Programming Problem

Since it is hard to compute an optimal solution in reasonable time, we implemented a hybrid local search algorithm coupled with the Overlap MILP and a Constraint Satisfaction Solver. The local search algorithm we implemented is illustrated on Figure 3.15. A first solution is computed using a Constraint Satisfaction solver, with a limited execution time ($\frac{1}{20}$ of the total allowed processing time). Then, a time frame that contains a very expensive activity is chosen, and optimized using a MILP operator, while there is execution time left.

In practice the use of binary variables $\mathbf{pos}_{i,l}$ for computing the energy cost seems to direct CPLEX away from the optimization of the tardiness and hence leads to poor solutions. To allow the Overlap formulation to run faster as a MILP operator, we have chosen to relax the $\mathbf{pos}_{i,l}$ variables to have a value in $[0, 1]$.

In order to get quickly a feasible solution, and bootstrap the Local Search Solver, we have implemented a naive Constraint Satisfaction Problem with the Choco 3 solver (cf (Prud'homme, Fages, and Lorca, 2016)). We modeled all variables as integers in order to improve the solver efficiency. As a branching strategy, we chose to look over start time/end time variables first because an "earliest possible solution" would be highly efficient to bootstrap the Local Search. We use also a "last conflict" composite heuristic that hacks the main strategy by forcing the use

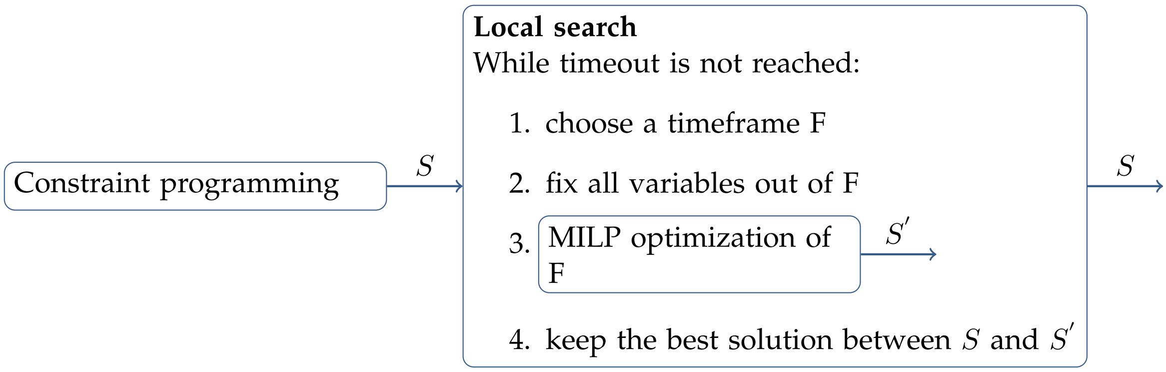


FIGURE 3.15 – Local search principles.

of variables involved in recent conflicts. As a search strategy in variables domain, we sorted the production orders according to the Earliest Due Date rule. Then, by subtracting the duration of each task and propagating the precedence constraints, we determined a partial order on the scheduled activities.

3.4.5 Complexity

Before presenting results, let us comment the size of the each formulation, depending on input data size. Table 3.2 gives the number of variables and constraints for each formulation. We can see that the storage-oriented formulation has more variables and constraints than the precedence-oriented formulation, while both have $O\left(|T| \times \left[|J| + \sum_{\mathcal{B}_l \in \mathcal{B}} |I_l|\right]\right)$ variables. On the other hand, the overlaps formulation has $O\left(|J|^2 + |J| \times |\mathcal{B}|\right)$ variables and constraints.

TABLE 3.2 – Dimension of the studied MILP formulations.

Formulation	Overlaps	Largest instance
Variables	$4\|J\|^2 + 16\|J\| + 18\|J\| \times \|\mathcal{B}\|$	191100
Including binaries	$4\|J\|^2 - 3\|J\| + 6\|J\| \times \|\mathcal{B}\|$	179550
Constraints	$8\|J\|^2 + 23\|J\| + 30\|J\| \times \|\mathcal{B}\|$	376530
Formulation	Precedence-oriented	Largest instance ($\tau = 10$mn)
Variables	$\|T\| \times (6\|J\| + 2\sum_{\mathcal{B}_l \in \mathcal{B}} \|I_l\| + 1) + 13\|J\|$	1285914
Including binaries	$\|T\| \times (6\|J\| + \sum_{\mathcal{B}_l \in \mathcal{B}} \|I_l\|)$	1276128
Constraints	$\|T\| \times (3\sum_{\mathcal{B}_l \in \mathcal{B}} \|I_l\| + 7) + 25\|J\|$	30450
Formulation	Storage-oriented	Largest instance ($\tau = 10$mn)
Variables	$\|T\| \times (19\|J\| + 2\sum_{\mathcal{B}_l \in \mathcal{B}} \|I_l\| + 1) + 8\|J\|$	4036704
Including binaries	$\|T\| \times (7\|J\| + \sum_{\mathcal{B}_l \in \mathcal{B}} \|I_l\|) + \|J\|$	1488018
Constraints	$\|T\| \times (22\|J\| + 3\sum_{\mathcal{B}_l \in \mathcal{B}} \|I_l\| + 7) + 19\|J\|$	4686150

Let us verify the impact of those dimensions on the experimental results.

3.5 Experiments

This chapter reports on a computational comparison of formulations described in Section 3.4. The implementation was done in Java using the Concert library of IBM ILOG Cplex optimizer 12.6.1. The experiments were launched on a Lenovo W540, with an Intel(R) Core(TM) i7-4810MQ CPU (2.80GHz) and 32Go RAM.

First, a small instance is solved in order to illustrate the differences in the solutions obtained by the three formulations. First, the methods presented in Section 3.4 are tested on Benchmark instances from the literature. Then, instances built from the real plant data are used to check how accurate a decision-aid tool could be.

3.5.1 How Do the Three Formulations Behave?

The electricity cost function considered has three time buckets and two intervals of capacity in each bucket. For the sake of the example, the second capacity interval costs less than the first one.

Figure 3.16 holds Gantt diagrams obtained with each method on the same given small 2-jobs instance. Each solution is optimal for its formulation and one-hour time-steps were chosen for the sake of the example. The X-axis represents the time horizon while the Y-axis represents the resources (electricity and machines). Considered resources are (top-down): electricity, body line, door line, a worker who hangs materials to be painted, another worker who unhangs those materials, door jointing machine, workers who assembly bodies and doors.

Figure 3.16a shows the overlaps formulation results. In this formulation, linear cost functions are taken into account instead of piecewise-linear cost functions. Indeed, only the first power capacity interval of each bucket is considered. Thus, scheduling energy consuming activities at the same time is not significant in this formulation. The door line activities are scheduled early to benefit of the cheap electricity tariff. The body activities are scheduled just before the mounting activities, when the electricity is cheap again, in order to pay a small storage cost.

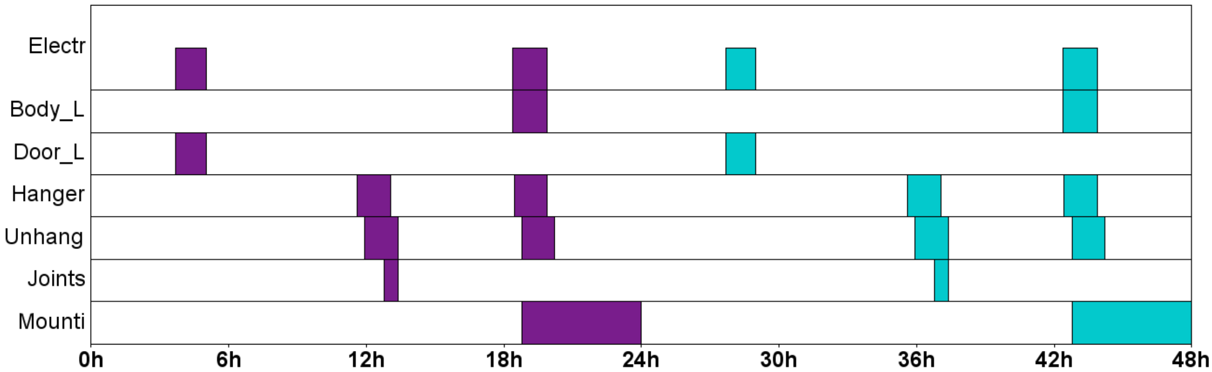
However, the precedence-oriented formulation schedules body activities at the same time as door activities. That allows the schedule to benefit from the second power capacity interval where the electricity cost is much lower than every other costs. In this case, paying more storage costs is interesting to gain even more on electricity cost.

On the other hand, the storage-oriented formulation uses preemption to refine the trade-off between storage costs and electricity costs. Indeed, body activities are preempted to benefit from the low electricity costs in the first bucket as well as reducing storage costs by scheduling the last part of body activities at the latest time possible. Moreover, hanging and unhanging tasks are preempted to save storage cost because of the low consumption rate of the mounting activities.

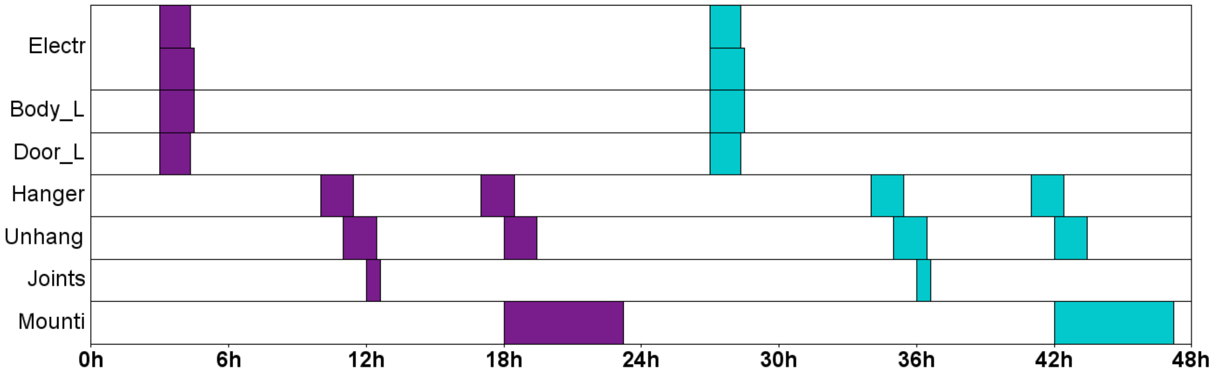
3.5.2 Results on Benchmark Instances

The MascLib Benchmark (cf (Le Pape and Robert, 2007)) is composed of scheduling problems from real business data and from the literature. These instances had initially no energy cost. Thus, appropriate energy consumption and tariff data have been added. In general, a day/night contract with an off-peak price 40% to 50% lower than the peak price. Piecewise-linear energy cost functions will be considered in the next subsection. The set of instances is composed of two different types of instances (NCGS and NCOS) that have from 12 to 200 operations. NCGS are instances of the No-Calendar-General-Shop problem. NCOS stands for No-Calendar-One-Step; in these instances all the recipes have only one activity and there is only one resource of unitary capacity. None of these benchmark instances have storage costs.

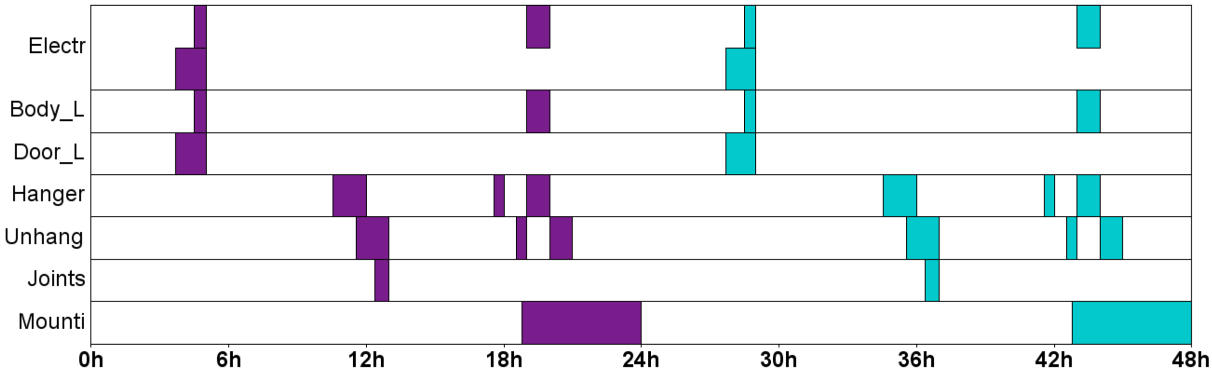
Four resolution methods were considered:



(a) Overlaps formulation



(b) Time-based precedence-oriented formulation



(c) Time-based storage-oriented formulation

FIGURE 3.16 – Gantt diagrams obtained on a two-days instance.

1. the local search algorithm,
2. the Overlap MILP without energy optimization,
3. the Overlap MILP,
4. the Precedence-Oriented MILP.

For all runs, a CPU time limit is imposed, depending on the number of activities in the problem instance. The three MILP formulations are warm started using a first solution computed by the naive Constraint Programming formulation.

We first compare methods 1, 3, and 4 on the 38 instances (without energy cost) to the best results known in the literature. Then, we try to establish how much energy cost can be saved without compromising (too much) the other objectives.

Optimization comparison with the best known results

We ran this experiment on the instances without energy cost, since there are no best solutions with the energy in the literature. Moreover, we had to adapt the time step of the Precedence-Oriented formulation to each instance. We proceeded automatically for each instance, by choosing as a time step the minimum duration needed to produce one unit of material. We tried iteratively to decrease the time step when the gap obtained was closed to zero, or to increase the time step when the gap was closed to 100%. Let us note that, knowing the context, a relevant time step could be determined easily.

TABLE 3.3 – Results on Benchmark instances of scheduling problems without energy cost.

	Local Search	Overlap MILP	Precedence-Oriented MILP
ALL			
NB-RUN	38	38	38
NB-BEST (= best known)	29	34	26
GAP regarding best known solutions	5.6%	0.9%	16.4%
NCGS			
NB-RUN	20	20	20
NB-BEST	17	20	14
GAP regarding best known solutions	8.7%	0%	30.1%
NCOS			
NB-RUN	18	18	18
NB-BEST	12	14	12
GAP regarding best known solutions	2.1%	1.9%	1.2%

For all the 38 instances, and then for the NCOS and NCGS separately, we computed the number of instances where we found the same result as the best known result in the literature. Results are presented in Table 3.3, where NB-RUN denotes the number of instances used and NB-BEST denotes how many times the best-known solution was reached. Moreover, the gap between the total cost achieved and the total cost of the best known solution is used as a performance indicator. GAP denotes the average deviation from the best-known solution, i.e.: $GAP = \frac{1}{\text{NB-RUN}} \sum_{instances} \frac{\text{cost of computed solution}}{\text{cost of best-known solution}} - 1.$

We can see that, on these instances, our resolution methods are convincing. The Overlap formulation found the best known solution for 89% of the instances, with an average gap from the best known solution of 0.9%. The Local Search procedure and the Precedence-Oriented formulation were less successful, with respectively 76% and 68% of the instances solved with the best-known cost achieved. If we look more closely at the NCGS instances, the Overlap formulation always found the best known solution, and the Precedence-Oriented formulation performed badly with an average gap of 30.1%. This seems to indicate that the Precedence-Oriented formulation has some difficulties to deal with precedence. On the other hand, on NCOS instances, the Precedence-Oriented formulation obtained the best average gap of 1.2%. However, the Overlap formulation found the biggest number of best known solutions. In all the cases, the Local Search procedure performed moderately. It probably would have to be adapted to achieve better results. For example, a new local search operator using the Precedence-Oriented formulation could be implemented. Another first solution could also be computed with the Overlap MILP without energy optimization.

Comparison with and without energy

We wanted here to compare schedules computed with energy optimization to schedules computed without taken energy into account. Let us note that we considered here linear electricity cost functions, as peak/off-peaks tariffs were probably used in the real context of those Benchmark instances.

Table 3.4 shows the results obtained regarding energy optimization. NB-IMPROVED represents the number of instances (over NB-RUN) that were (strictly) improved when optimizing the electricity cost in addition to the other objectives. Some instances were not improved because taking the energy cost into account slowed down the solver and gets the tardiness cost to be really increased. MEAN-IMPROVEMENT represents the average gain on the total cost obtained when optimizing the energy cost compared to a solution obtained by the Overlap formulation without energy optimization. When MEAN-IMPROVEMENT is negative, that means the total cost have been increased. This is the case in average, because of the non-improved instances for which the tardiness cost has been far more increased than the decrease of energy cost on improved instances. This is why we also look at the MEAN-IMPROVEMENT-COND indicator, which takes only into account the improved instances to compute the gain.

We can see on Table 3.4 that the Local Search procedure improved the biggest number of instances ($\frac{26}{38} = 68\%$). But the Overlap formulation is the one that degrades the least the total cost in average (for improved and non-improved instances). On the other hand, the Precedence-Oriented formulation achieves a bigger gain for improved instances (1.1% in average, 9.6% in the best case). If we look only at the NCGS instances, precedence constraints induce worse performances for the three methods, but the trend is the same. If we look only at the NCOS instances, the MEAN-IMPROVEMENT of the Precedence-Oriented formulation is positive, meaning that an average gain is achieved, even if we take into account the non-improved instances.

Globally, savings are small. On the instances where the total cost could be improved in the given computation time, less than 1% of the total cost could be saved, up to 9% in the best

TABLE 3.4 – Results on Benchmark instances of scheduling problems with energy cost.

	Local Search	Overlap MILP	Precedence-Oriented MILP
ALL			
NB-RUN	38	38	38
NB-IMPROVED	26	24	19
MEAN-IMPROVEMENT	-12.4%	-7.5%	-11.9%
MEAN-IMPROVEMENT-COND	0.6%	0.6%	1.1%
NCGS			
NB-RUN	20	20	20
NB-IMPROVED	18	16	12
MEAN-IMPROVEMENT	-19.3%	-10.7%	-23.2%
MEAN-IMPROVEMENT-COND	0.5%	0.5%	0.5%
NCOS			
NB-RUN	18	18	18
NB-IMPROVED	8	8	7
MEAN-IMPROVEMENT	-4.7%	-4.0%	0.7%
MEAN-IMPROVEMENT-COND	0.8%	0.8%	2.2%

case. However, for those instances from the industry, 1% of the total cost could represent a big amount of money. Let us note also that most of the instances have too long activities compared to the buckets size, thus the margin to improve energy costs is really small. Different electricity cost functions would probably improve the gains, but they would have to be adapted to the time scale of the instance. If we had to propose a decision-aid tool in the real context of those instances, we probably would:

1. compute a first solution with the Overlap Formulation without energy cost, warm started by the Constraint Satisfaction Solver,
2. run in parallel both the MILP formulations warm started by this first solution and the Local Search Procedure using both formulations as local search operators,
3. keep those four solutions as choices for the decision maker.

3.5.3 Results on Instances Built from the Pilot Plant Data

The experiments have been conducted on four different instances, that have been built using the real plant data. The first one has two demands for a time horizon of two days. The second one has six demands for a time horizon of five days. Both are used to evaluate results of the formulations on small instances (far smaller than those of the real plant). The third one has two hundreds and ten demands for a time horizon of seven days and no precedence (there are only the body line data). This is a real instance, but without any precedence, in order to evaluate their impact on the results. The forth one is the same as the third one, but with all activities and precedence. This is the instance we have to solve, in order to be able to built a decision-aid tool for the real plant.

The electricity cost functions are the same in every instance: there are three time buckets for each day (0h - 5h ; 5h - 21h ; 21h - 0h) and two intervals of capacity in each bucket. This is equivalent to a peak/off-peak tariff with 0.002 euro cents per kWh in off-peak hours and 0.004 euro cents per kWh in peak hours, except that the second capacity interval (starting from 1kW) is a twice more expensive than the first one.

Our experimental study aims to compare resolution methods in term of solution quality and run time. For the time-indexed formulation, several time steps are tried in order to evaluate their impact on run time and solution quality. Maximum run time is also tuned to see how the methods behave through their resolution process. We use the naive Constraint Satisfaction Problem presented in Subsection 3.4.4, in order to compute a first feasible solution. We provide (or not) formulations with this first feasible solution to allow them to cut the branching tree faster. The impact of those warm starts is investigated during experiments. Finally for each instance, we ran the Overlaps formulation without the energy-related variables, in order to compare an optimal non-energy-aware solution with the others.

Now, let us present Table 3.5, giving numeric results. The first three columns give the formulation used, whether a warm start was performed or not, and the time-step τ (if relevant). The other columns give, for each instance, for each maximum run-time allowed: the gap between the best integer objective value and the best relaxed objective value, the energy cost, the storage cost and the tardiness cost of the best solution obtained. Instances are denoted by the couple: (number of jobs, number of days).

On the smallest instance (2,2), all formulations give optimal results in less than 1mn. As there are only two jobs, all the formulations schedule energy consuming activities at different times. Thus, the fact the Overlaps formulation does not take into account piecewise linear energy cost is not a problem for this instance. For the first two rows, energy is not part of the objective function and the total cost obtained if 0.79 €. All other formulations give the same optimal objective value of 0.68 € (but not the same solution).

On the second instance (6,5), all results are still optimal in less than 1mn. The Precedence-oriented formulation takes advantage of its knowledge of the piecewise linear cost function to achieve the best results (more than five times better than the Overlaps formulation). Indeed, the Overlaps formulation finds an optimal solution where consuming activities are scheduled at the same time. This is not a problem regarding its objective function, but it is when the electricity cost is evaluated with the piecewise linear functions. Even if the Local Search uses the Overlaps formulation as an operator, its solution is really close to the best solution found. That can be explained by the fact that Local Search evaluates solutions, to decide to keep them or not, using the real electricity cost.

On the third instance "(210,7) bodies", Local Search obtains good results, even in ten minutes (more than 11% better than Overlaps without energy optimization). The Precedence-Oriented formulation is not far away, but the warm start is required with a 10mn time-step. Let us note that increasing the time step decreases the run-time needed to find good solutions (with a small gap). On the other hand, an optimal solution with a bigger time-step is worse than (or equal to) an optimal solution with a smaller time-step. Finally, the Overlaps formulation has a hard time to reduce the gap and find good solutions, because of the size of the instance.

Finally, the biggest instance (one week of the real plant) is solved only by MILP with warm start and the local search procedure. The first solutions (found by the CSP solver) are not improved during a hour of computation by the MIP formulations. The local search procedure

TABLE 3.5 – Formulations results on realistic instances.

Formulations	warm start	τ (mn)	Instances ▶	(2,2)	(6,5)	(210, 7) bodies		(210, 7)	
▼	▼	▼	Run time ▶	1mn	1mn	10mn	1h	10mn	1h
Overlaps without energy optimization	✗		gap	0	0	92.2	91.6	∞	∞
			energy	0.72	2.04	505.41	503.9	∞	∞
			storage	0.07	0.82	5.35	5.53	∞	∞
			tardiness	0	0	0	0	∞	∞
Overlaps without energy optimization	✓		gap	"	"	97,8	91,3	99.7	99.7
			energy	"	"	481,13	504.65	1960.64	1960.64
			storage	"	"	15,72	5.38	30.38	30.38
			tardiness	"	"	0	0	0	0
Overlaps	✗		gap	0	0	∞	∞	∞	∞
			energy	0.62	12.98	∞	∞	∞	∞
			storage	0.06	1.23	∞	∞	∞	∞
			tardiness	0	0	∞	∞	∞	∞
Overlaps	✓		gap	"	"	99.2	98.9	100	99.9
			energy	"	"	471.56	456.54	1960.64	1960.64
			storage	"	"	32.05	24.28	30.38	30.38
			tardiness	"	"	0	0	0	0
Precedence-oriented	✗	10mn	gap	0	0	100	100	∞	∞
			energy	0.41	1.64	412.02	412.02	∞	∞
			storage	0.27	1.09	5.92	5.92	∞	∞
			tardiness	0	0	3×10^{10}	3×10^{10}	∞	∞
Precedence-oriented	✓	10mn	gap	"	"	75.4	0	100	70.7
			energy	"	"	476.72	448.89	2004.89	2004.89
			storage	"	"	27	11.9	20.99	20.99
			tardiness	"	"	0	0	0	0
Precedence-oriented	✗	15mn	gap	0	0	0.1	0.1	∞	∞
			energy	0.41	1.64	464.3	464.3	∞	∞
			storage	0.27	1.11	11.28	11.28	∞	∞
			tardiness	0	0	0	0	∞	∞
Precedence-oriented	✓	15mn	gap	"	0	0.1	0.1	100	67.1
			energy	"	1.64	464.36	464.36	2071.65	2071.65
			storage	"	1.07	11.21	11.21	14.46	14.46
			tardiness	"	0	0	0	0	0
Local search			gap						
			energy	0.62	1.61	429.23	423.82	1957.53	1745.98
			storage	0.06	1.16	12.54	12.09	27.78	16.77
			tardiness	0	0	0	0	0	0

gets better results, succeeding to improve greatly its solution during the last fifty minutes of computation. Savings of 10.9% on the electricity bill are finally achieved, compared to the solution found by the Overlaps formulation without energy optimization. Since the local search has for the moment a single MILP operator based on the Overlap formulation, better results would probably be achieved with a second local search operator based on the Precedence-oriented formulation.

To conclude, the size of the formulations has not the expected influence on the solution quality. As the Precedence-oriented formulation is time-based, a better linear relaxation seems to allow it to find results faster than the Overlap formulation for big instances. But this is not sufficient to handle a realistic instance efficiently. On the other hand, these first results show that 10.9% of the electricity bill can be saved by optimizing energy consumption and considering the chosen cost function. Extrapolated to the real plant, savings of 10.9% of the annual electricity bill would represent about 109000 euros. This result would have to be confirmed with a more accurate modeling of the activities duration and electricity cost.

3.6 Conclusion

In this chapter, a Schneider Electric manufacturing plant is studied. The goal is to determine if taking into account the electricity cost when scheduling production plans could generate saving on the total bill. In order to compute production plans, we had to know how much energy is consumed in producing each kind of reference. Thus, we measured energy consumed by each machine, over a long time period. Then we synchronized energy measures with production log, and used a regression algorithm to compute approximated energy consumption of each activity, as well as the baseload. The approach has been validated on clean datasets from benchmark files. In addition, we confirmed for the studied plant, that energy consumption varies, depending on the kind of reference produced. This potentially enables savings on the electricity bill, by scheduling most energy-consuming activities when the electricity is cheaper. However, contrarily to what we expected, we did not find out any correlation between the size of the cabinet produced and the energy consumption.

On the other hand, the scheduling problem taken from the real manufacturing plant is formalized. Three MILP formulations are given in order to solve the problem efficiently with some modeling approximations. The emphasis is put on the electricity cost modeling and a set of piecewise-linear cost functions is considered. The number of variables stays polynomial, contrarily to a formulation found in the literature for a slightly different problem with the same kind of cost functions (cf (Ngueveu, Artigues, and Lopez, 2016)). We show that good solutions can be found, either for Benchmark instances and for instances built from the real plant data, with the proposed formulations. Savings of about 10.9% of the electricity bill are achieved for the most realistic instance built from the real plant data.

In the real manufacturing plant context, those formulations are aimed to be integrated in a decision-aid tool. Such a tool could be used by the plant manager to generate and compare possible production plans, as well as their impact on each of the cost functions. A good way to build a decision-aid tool could be to use all the formulations as local search operators, on accurately chosen time-windows. A first solution computed without any energy optimization could feed this local search procedure. Moreover, the electricity tariff should be chosen among a set of possible ones, regarding the flexibility of the plant. Such a decision-aid tool could also assist a plant manager for participating in the demand-response market.

Finally, from a fundamental research point of view, the following tracks are considered. The energy models of activities should be compared to models based on finite-state machines, in order to evaluate how much precision is lost. The Storage-oriented formulation should be further investigated. Decision variables would have to be changed in order to generate feasible solution in the first place. Adding setup costs on activities would discourage preempting too many times activities if the gain is not so big. An event-based formulation, taken into account piecewise-linear energy cost functions should be possible to find. That would need to model in an exact way the instantaneous power consumed, using (Artigues, Lopez, and Haït, 2013). On the other hand, a fundamental question concerns the strength of linear relaxations of the three formulations proposed. This question of interest requires to be dug into. A multi-objective approach, providing Pareto-optimal schedules could also be interesting. Finally, decomposition methods could be investigated.

Chapter 4

Control and Planning of Prosumer Systems

Contents

In order to handle our three use-cases described previously, we had to imagine a relevant software architecture. A generic object-oriented model was implemented in Matlab, and web services were implemented in other languages. In this chapter, we inspired of what we learned when building a software architecture relevant for our three use-case, to give practical guidelines for control, planning and design of prosumer systems.

4.1 How to Control Electricity Purchasing?

There are two ways to control electricity purchasing: limiting the maximum power peak purchased, or minimizing the electricity bill. Minimizing the electricity bill is a long-term objective, as a sufficiently long time-frame has to be considered to benefit of the energy tariff changes. On the other hand, limiting the maximum power peak purchased is only possible with a real-time controller. Although this can be facilitated by considering piecewise-linear electricity cost functions in a long-term optimization. Furthermore, when an energy manager wants to act on his electricity bill, he has two directions to investigate: shift consumption when electricity is cheap, or smooth consumption by storing electricity.

There are cases, like a manufacturing plant or a water network, where the electricity consumption can be shifted of a noticeable duration. In the studied manufacturing plant, the most consuming activities can be scheduled at times the electricity is cheap. Indeed, intermediate products (doors or bodies) can be produced and stored when electricity is cheap, and used to produce cabinets when electricity is expensive. In a water network, pumps can be used to fill in water towers mostly when electricity is cheap. In both cases, something is transformed using electricity, and stored to be used later. A specific scheduling algorithm has to be developed to take into account constraints and specific objectives of the system, as well as to minimize the electricity bill. Input data of such an algorithm are composed of (but not reduced to): energy models, forecasted (or well-known) customer demands, physical constraints (like delays, or yields). In most of the cases, the considered system is subject to (machine) failures, and if forecasts are used, there cannot be fully accurate. Thus, long-term plans computed by this scheduling algorithm should be used by a low-level controller to cope with real-time aspects. This low-level controller can be a manager, or an automated control system.

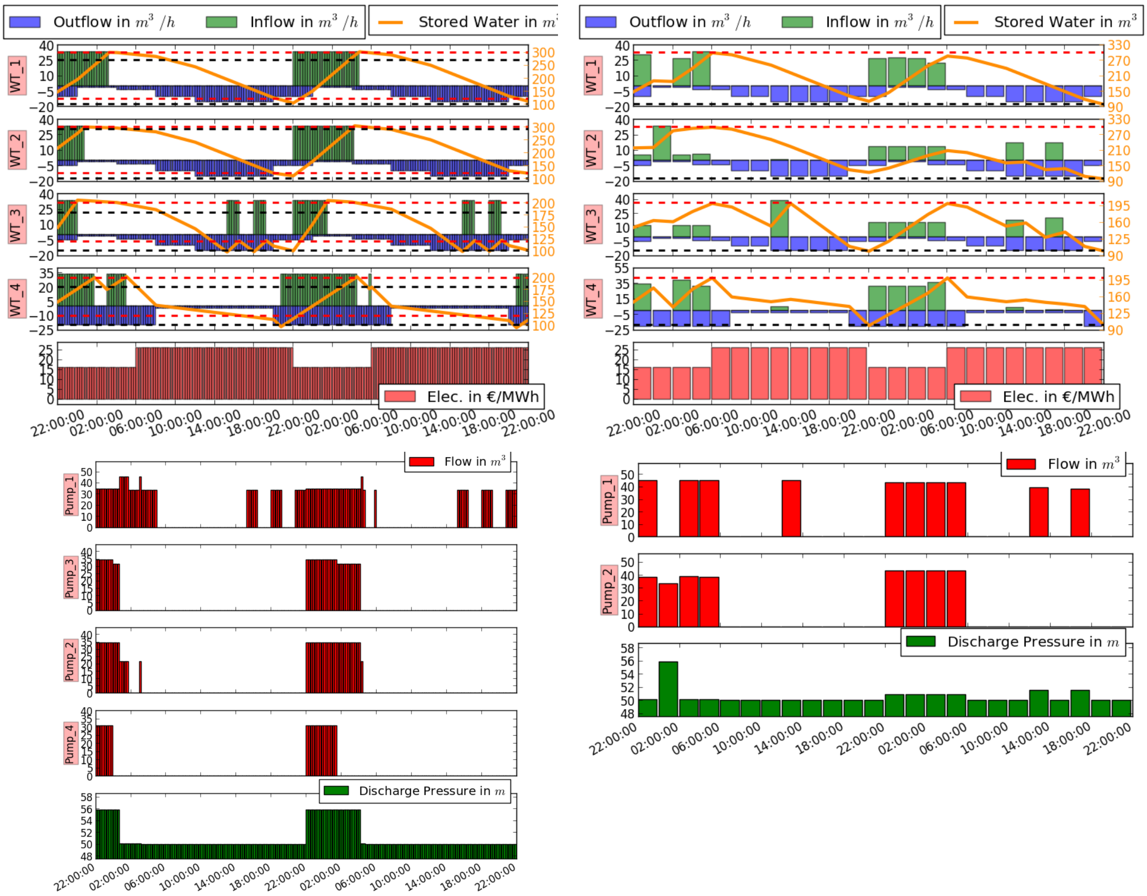


FIGURE 4.1 – Pumping plans without and with optimization.

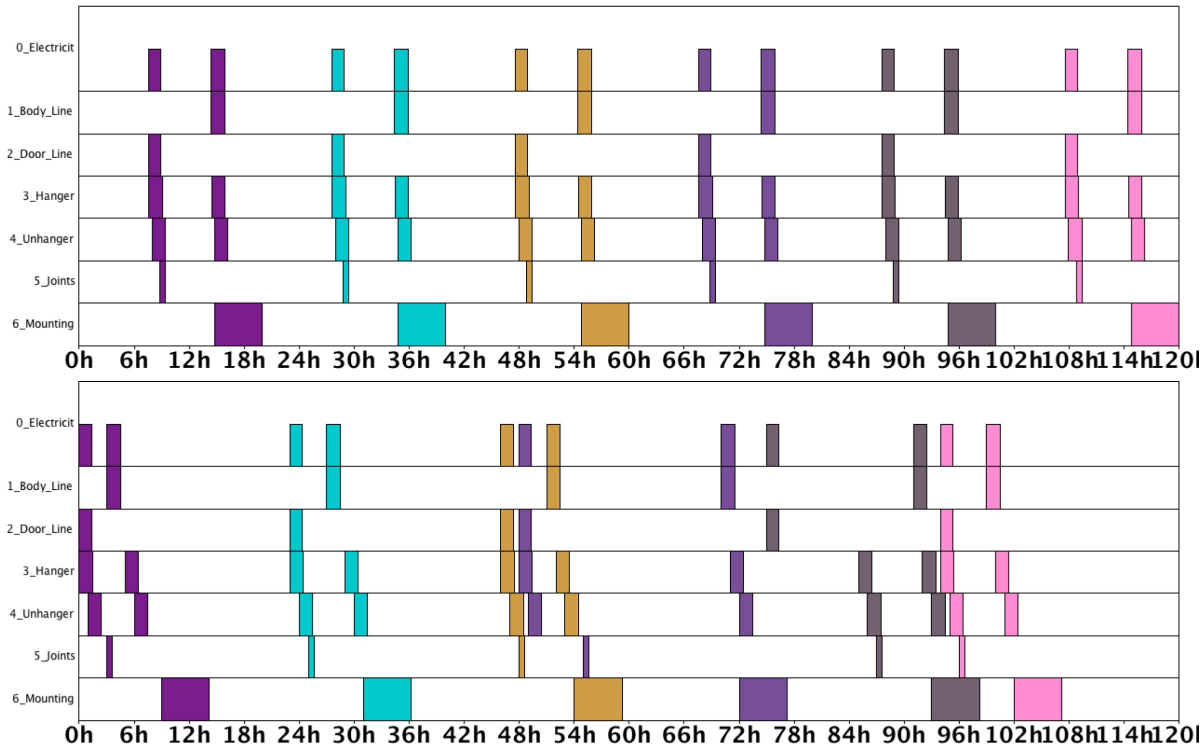


FIGURE 4.2 – Production plans without and with optimization.

Examples of pumping and production plans are given on Figures 4.1 and 4.2. On Figure 4.1, we can see Business As Usual (BAU) pumping plans and optimized pumping plans (resp. on the left side and on the right side). Graphs on the top represent the evolution of the water level in water towers, compared to the water flows and the electricity tariff over a given day. Graphs on the bottom represent water flow coming from each of the pumps and the corresponding discharge pressure. Optimized pumping plans are more stable than BAU ones, regarding the flow coming from the pumps. Pumps are also used in a more efficient way regarding the yield. This is possible because of water demand prediction, and a model of pumps efficiency. On Figure 4.2, two production plans for the same voluntary under-loaded week are presented. The first one does not optimize the energy cost, and the second one does. We can see that when the electricity bill is minimized, activities are scheduling sooner, even if that increases the storage cost. Whatever is the context, one of the core question is: how to use those strategic plans during real-time control?

In other cases, shifting electricity consumption is complicated, or even impossible. For example, the electricity consumption of an elevator cannot be shifted more than five or six seconds (the admissible waiting duration). Thus, electricity itself has to be stored into batteries, in order to limit the impact of an uncontrollable energy consumption. In that case, coupled-controllers proposed in Chapter 2 can be used as they are, or enriched with new prosumers (like gensets).

If the system considered can shift its consumption and has multiple sources of energy, optimal plans would address both energy consumption scheduling and sourcing plans. But such a system would probably be too complex for both problems at the same time. Thus, interactions between scheduling algorithm and multisource optimization has to be defined. I would recommend to first compute an optimal consumption schedule, then to use it as a consumption forecast for the multisource optimization. The low-level multisource controller proposed in Chapter 2 is then capable to exploit real-time feasibilities of each prosumer, depending on its priority level. A more detailed view on the proposed software architecture is given in Section 4.2.

The same high-level features are needed to control every prosumer system. There are introduced in Figure 4.3.

First, data have to be collected to get to know the system. They can be:

- customer demand data (water demand at every water tower, cabinet production demand, or user call to the elevator for example),
- energy data (electricity consumption of machines, of the elevator, of the pumps),
- external data (like weather data) that influence energy consumption or customer demands of the system,
- or even the real-time state of the system (like storage units state of charge, weight of the cabin of the elevator, production machines state).

These data can then be used to build an energy model of the system. A simple model can be built from energy data, external data and state data, with simple machine learning algorithms. Such a model could have been used for an elevator and has been used for the manufacturing plant. An energy model of the activities was built based on energy measurements, and production data (state of the system). A more accurate model could have been based on precise machines state data. Or a more complex model can be built based on physical knowledge of the system. Such a model has to be fed with more data than the previous one. This is what we did for the elevator, and our model is fed by the altitude of the starting and destination floors, the weight of the cabin (including passengers), the nominal speed, and the cable length. Let us note that this model computes energy consumption or production but suppose constant power,

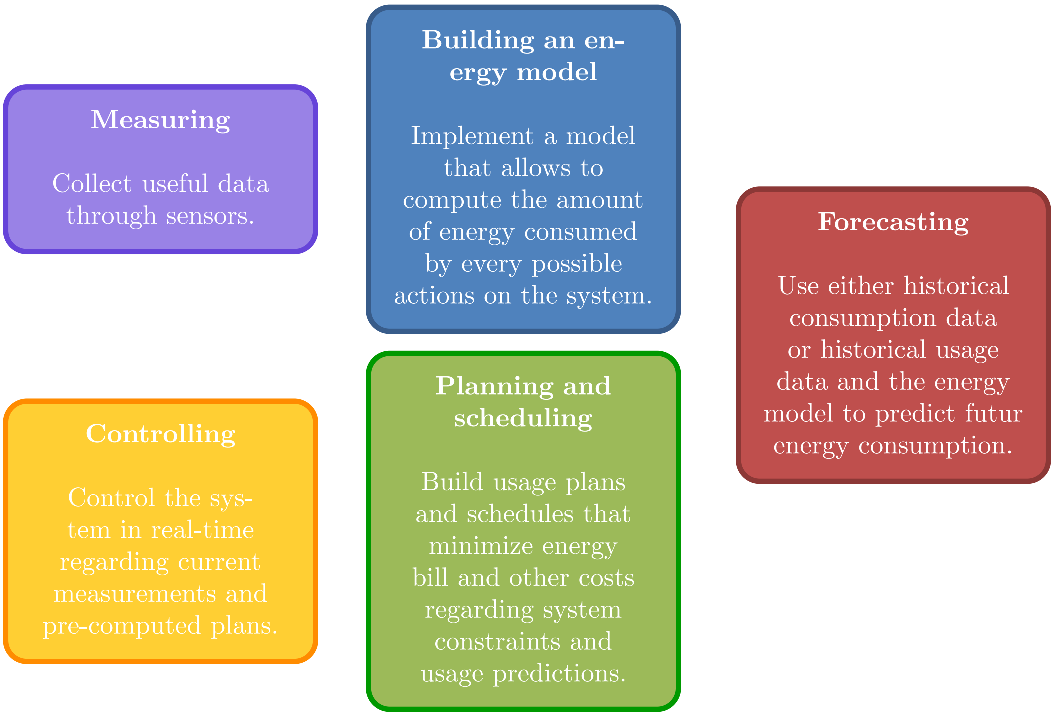


FIGURE 4.3 – Needed features to optimize a prosumers system.

that is not true at all in reality. An energy model of the water network pumps was also built from physical data, and is fed by the needed water flow.

Using data collected and energy models, future customer demands and/or energy consumption of the system have to be forecasted, in order to allow computing optimal schedules. For example, in the elevator use-case, future energy consumption of the elevator is forecasted, as well as solar panels energy production. The former is built using a statistical model of user calls and an energy model; the latter is based on weather forecasts and an energy model. These forecasts are then used to compute a sourcing strategy. In the manufacturing use case, customer demands are well-known. Thus, no forecasts are needed. Finally, in the water network use case, water demand is forecasted using historical weather and water demand data, and a machine-learning algorithm.

Once all that information is available, context specific scheduling algorithms can be used to compute optimal energy-aware schedules. Schedules can either be sourcing plans, consumption schedules or both, as described in the beginning of this section. Finally, every prosumers system has to be controlled in real-time, taking into account high-level scheduled previously computed.

Now, let us introduce the principles of the proposed software architecture.

4.2 Proposed Software Implementation

As explained in the previous section, each prosumer has to be modeled in order to be controlled. This model, in blue on Figure 4.4, is composed of an energy model and a behavioral and physical model, depending on the measured data. Depending on the local controller decisions, the energy model is used to compute how much energy would be used. Moreover, the behavioral and physical model keeps the system state up to date.

For example, in the elevator use-case, the system states are: current floor, current cabin weight and whether the elevator is moving or waiting or doing nothing. The current cabin

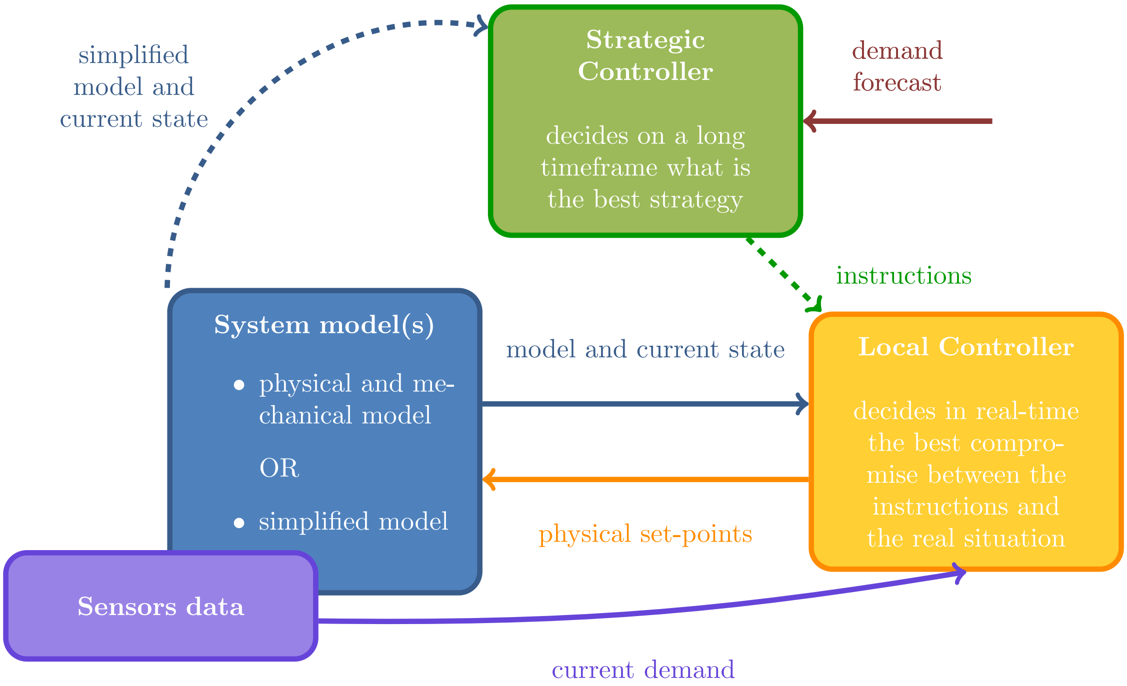


FIGURE 4.4 – Proposed software architecture for a prosumer system.

weight is measured and the data comes from a (virtual) sensor, but the other states are maintained by the model. A more accurate model, taking into account current speed and power consumption of the elevator, would be needed for a real elevator. But for the simulations purpose, we use a simplified model, knowing only the nominal speed and the global energy consumption of an elevator travel.

The second software component needed by a prosumer system is the local controller, in yellow on Figure 4.4. The local controller takes real time decisions in order to meet customer demands. For a prosumer to be controlled, the local controller has to know what are its flexibilities at a given time step, based on customer demands and current state. Each flexibility is linked to its energy consumption. Then the local controller has to decide, at every time step, which flexibility to use. In most of the cases, this controller can be a simple rule-based one. Once the decision is taken, the corresponding set point is sent to the system model in order to update the system state.

For example, for the elevator, the flexibilities can be: stay at the current floor and wait, or go up because of a user call, or go down because of a user call. A simple state machine is used to decide the order of user calls to meet. Then the destination of the first user call is sent to the model to update the system states.

The last software component part of a prosumer system is the strategic controller, in green on Figure 4.4. It is optional because the system can be controlled only by a local controller, using a model. But the strategic controller allows a prosumer system to be optimally controlled, based on forecasts. Indeed, a local controller has to be real-time and accurate, thus long-term objectives cannot be considered. Long-term objectives (like minimizing energy consumption, or minimizing the economical cost of the system) are optimized by a strategic controller. This component uses forecasts of customer demands, as well as forecasts of other data influencing the prosumer system. A simplified model of the system is used in the strategic controller, in order to keep the optimization problem as simple as possible.

For example, the manufacturing plant strategic controller uses customer demands and simple activities energy model to optimize the energy consumption of the plant (as well as other KPIs). For this purpose, an optimal production plan is built, composed of the starting times of each production activity. In that case, the local controller would be the plant manager and the workers themselves. From the pre-computed production plan, they would have to deal with the machine failures and other unexpected events, in real time.

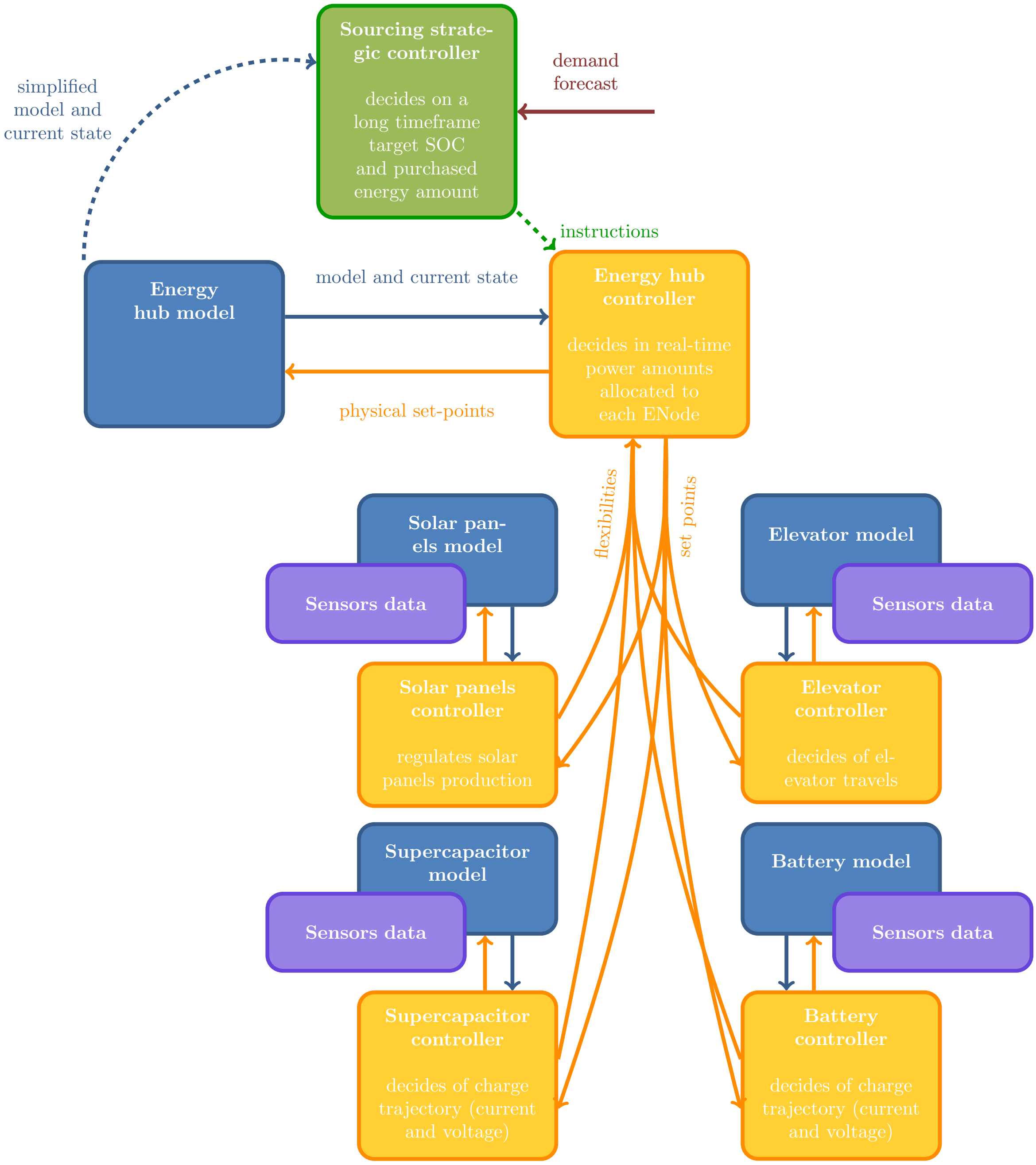


FIGURE 4.5 – A more complex software architecture with an energy hub.

We can take another example of strategic controller with the elevator. We did not use any strategic controller for the elevator itself, because the quality of service is a priority in this field and is ensured by the local controller. But we could have considered a set of several elevators in the same building. In that case, a strategic controller could have dispatched between elevators the different user calls, in order to optimize the energy consumption, and the quality of service (waiting time for example). A reduced set of user calls to meet would have been sent as an instruction to each local controller.

But we did not consider this track in the elevator use-case, because we focused on the sourcing problem. That led us to consider a special kind of prosumer: the energy hub. An energy hub is the root of a star graph of prosumers. The corresponding software architecture, applied to the multisource elevator use-case, is illustrated on Figure 4.5. The energy hub controller aims to maintain the power equilibrium between all prosumers, by degrading the quality of service of prosumers of low priority. For that purpose, flexibilities of each prosumer are requested by the energy hub controller at every time step, as power intervals or discrete power alternatives. Then, a decision is taken using a depth-first search in a tree; and power set-points are sent to each prosumer. Algorithms 2 and 3 show how we implemented the energy hub controller.

Algorithm 2 Energy hub's Local Controller algorithm for time step t and period τ.

Require: $\mathcal{P}$ the set of n_p prosumers
Require: $instruction$ the relevant strategic instruction
▷ **check flexibilities of every prosumer during** $[t, t+\tau[$
  $flex \leftarrow [\,]$ ▷ let $flex$ be the ordered list of the n_p prosumers flexibilities
  for $\pi_i \in \mathcal{P}$ **do**
    $flex[i] \leftarrow getFlexibilities(i, t+\tau)$ ▷ let $flex[i]$ be a list of pairs where: the first member represents an instant in $[t, t+\tau[$, the second member represents different flexibilities for Prosumer π_i at this instant
  end for
▷ **make the time steps match between prosumer flexibilities**
  $times \leftarrow \{\,\}$ ▷ let $times$ be a set of instants considered in $[t, t+\tau[$
  for $\pi_i \in \mathcal{P}$ **do**
    for $(t', flex_{i,t'}) \in flex[i]$ **do**
      $times \leftarrow times \cup \{t'\}$ ▷ build the set of all different instants in $[t, t+\tau[$
    end for
  end for
  for $\pi_i \in \mathcal{P}$ **do**
    for $t' \in times$ **do** ▷ add a value for every time-step in every prosumer flexibility list
      if $flex_{i,t'}$ exists **then**
        do nothing
      else
        $flex_{i,t'} \leftarrow flex_{i,t''}$ with t'' the last instant in $flex[i]$ lower than t'
      end if
    end for
  end for

  $pList \leftarrow getPriorityList()$ ▷ **get the priority list depending on the current instruction**
▷ **choose the best combination of flexibilities depending on the priority list**
  $choice \leftarrow [\,]$ ▷ let $choice$ be the array of the n_p chosen power values for each instant in $[t, t+\tau[$
  for $t' \in times$ **do**
    $choice_{t'} \leftarrow choosePower(t', flex, instruction, pList, 0, [0 \ldots 0])$
  end for

Algorithm 2 ensures that all events in a given time-step $[t, t+\tau[$ will be considered. For example, if a given prosumer is supposed to stop what it was doing at $t+0.5\tau$, then $[t, t+0.5\tau[$ and $[t+0.5\tau, t+\tau[$ will be considered separately. Another example could be when a battery is going to be fully discharged at $t+0.3\tau$, then during the interval $[t, t+0.3\tau[$ the battery could be

Algorithm 3 Depth-first search algorithm of the energy hub's Local Controller.

Require: t
Require: $flex$ ▷ the ordered list of the n_p prosumers flexibilities
Require: $instruction$ ▷ the relevant strategic instruction
Require: $pList$ ▷ a list of prosumers, sorted by decreasing priority number
Require: $p \in \{0, \dots, n_p - 1\}$ ▷ number of the current prosumer in $pList$
Require: $choice$ ▷ power choices made by previous prosumers
  $i \leftarrow pList[p]$
  ▷ **choose the preferred flexibility for Prosumer** π_i
  if $instruction \neq \emptyset$ **then**
    $choice[i] \leftarrow$ the value closest to $instruction[i]$ in the flexibility range $flex_{i,t}$
  else
    if Prosumer π_i has a discrete list of flexibilities **then**
      $choice[i] \leftarrow flex_{i,t}[0]$
    else
      $choice[i] \leftarrow$ the value closest to the default instruction in the flexibility range $flex_{i,t}$
    end if
  end if
  ▷ compute choices of other prosumers with a smallest priority number
  if $p < n_p - 1$ **then**
    $choice \leftarrow choosePower(t, flex, instruction, pList, p + 1, choice)$
  end if
  ▷ **choose a less desired flexibility of Prosumer** π_i **if needed**
  if $\sum_{c \in choice} (c) \neq 0$ **then**
    if Prosumer π_i has a discrete list of flexibilities **then**
      $updatedChoiceNb \leftarrow 0$
      while $updatedChoiceNb < |flex_{i,t}| \wedge \sum_{c \in choice} (c) \neq 0$ **do**
        $updatedChoiceNb \leftarrow updatedChoiceNb + 1$
        $choice[i] \leftarrow flex_{i,t}[updatedChoiceNb]$
        ▷ update choices of other prosumers with a smallest priority number
        if $p < n_p - 1$ **then**
          $choice \leftarrow choosePower(t, flex, instruction, pList, p + 1, choice)$
        end if
      end while
    else
      $choice[i] \leftarrow$ the value in flexibility range $flex_{i,t}$ closest to $-\Big(\sum_{c \in choice} (c) - choice[i]\Big)$
    end if
  end if

discharged but during $[t + 0.3\tau, t + \tau[$ it has to be unused or recharged. However, if charging or discharging the battery is possible over the whole interval $[t, t + \tau[$, the interval is not split. Complexity of Algorithm 2 is in $O(n_p \times \zeta + \zeta \times c(\text{Algorithm } 3))$ with ζ the number of times the interval $[t, t + \tau[$ should have to be split to consider all possible events from all prosumers, and $c(\text{Algorithm } 3)$ the complexity of Algorithm 3. That allows simulations to be accurate but would probably not be done in practice. Indeed, choosing a sufficiently small time step and updating regularly the system state with sensors data should be sufficient to keep the model accurate.

Algorithm 3 is a recursive algorithm that illustrates the depth-first search in the tree of flexibilities. The complexity of this algorithm is in $O(\sigma^{n_p})$, with σ the maximum number of flexibilities of a prosumer at the given time step. The application to the multisource elevator use-case is explained with more details in Subsection 2.4.3. Several parametrizations of the prosumers priority order and default instructions are given. Regarding the complexity of the proposed rule-based LC, a simple Model-Based Predictive Controller could be preferable in some cases.

TABLE 4.1 – Linear Program addressing the sourcing problem.

Minimize

$$\sum_{l=0}^{H-1}\Bigg[\sum_{\pi_i\in\mathbb{P}}\Bigg(-c_i^{sold}[t_l]\times e_i^{sold}[t_l]+c_i^{purch}[t_l]\times e_i^{purch}[t_l]$$

$$+\min\left[\frac{\min\limits_{\pi_j\in\mathbb{P}}(c_j^{purch}[t_l])}{10},\frac{\min\limits_{\pi_j\in\mathbb{S}}(c_j^{aging})}{2}\right]\times\rho_i^{stab}[t_l]\Bigg)$$

$$+\sum_{\pi_i\in\mathbb{S}}\Bigg(\frac{c_i^{aging}}{2}\times e_i^{ch}[t_l]+\frac{c_i^{aging}}{2}\times e_i^{dis}[t_l]$$

$$+2\times\max_{\pi_j\in\mathbb{P}}(c_j^{purch}[t_l])\times\rho_i^{minSOC}[t_l]\Bigg)\Bigg]$$

Such that:

$\forall l \in \{0, \ldots, H-1\}$,

$$\sum_{i\in\mathcal{P}}(e_i[t_l]) = 0$$

$$\forall\pi_i\in\mathbb{P},\quad e_i[t_{l-1}]-e_i[t_l]-\rho_i^{stab}[t_l]\le 0$$

$$e_i[t_l]-e_i[t_{l-1}]-\rho_i^{stab}[t_l]\le 0$$

$$\forall\pi_i\in\mathbb{S},\quad x_i[t_{l+1}]=x_i[t_l]+\frac{c_i^{cy}}{c_i^{ce}}\times e_i^{ch}[t_l]-\frac{1}{c_i^{dy}\times c_i^{ce}}\times e_i^{dis}[t_l]$$

$\forall l \in \{0, \ldots, H\}$,

$$\forall\pi_i\in\mathbb{S},\quad x_i[t_l]+\rho_i^{minSOC}[t_l]\ge c_i^{minSOC}$$

The corresponding sourcing strategic controller decides, on a long timeframe, the target state of charge of storage units and the target power to purchase from controllable prosumers. The reason why the instruction for storage units is a target state of charge, rather than a target mean power, is that the LC can adapt the power needed in real-time to reach to target state of charge at the desired time. The linear program is summarize in Table 4.1 and described with more details in Subsection 2.4.2.

This particular strategic controller needs a forecast of the energy consumption and production of each non-controllable prosumer. These forecasts are obtained from prosumers's local controllers, based on other forecasted data. For example, the forecast of energy consumption of the elevator is obtained from the elevator local controller. The elevator local controller computes, based on forecasted user calls, its call answering tactic and the corresponding energy consumption. The time-series obtained is re-sampled and sent to the strategic controller. The solar panels forecasted production is computed based on weather forecasts and an energy model. More than those forecasted data, the strategic controller has to build an optimization problem with constraint coming from every prosumer. A centralized optimization problem is built, asking each prosumer its own energetic and power constraints. To keep the problem linear, constraints have to stay simple, and simplified prosumer models are used. A decentralized approach has not been investigated because of the simplicity and the small number of prosumers considered. In cases where a centralized approach does not make sense, the interested reader can refer to (Pflaum, Alamir, and Lamoudi, 2014) for the introduction of a hierarchical control structure (decentralized but supervised) that addresses a sourcing problem at the district level.

The last point we want to address about software engineering is Service Oriented Architecture. In order to optimize the development effort needed to handle the three Arrowhead pilots (elevator, plant and water network), shared web services were developed. The architecture we imagined is presented on Figure 4.6. These services would be called by deployed appli-

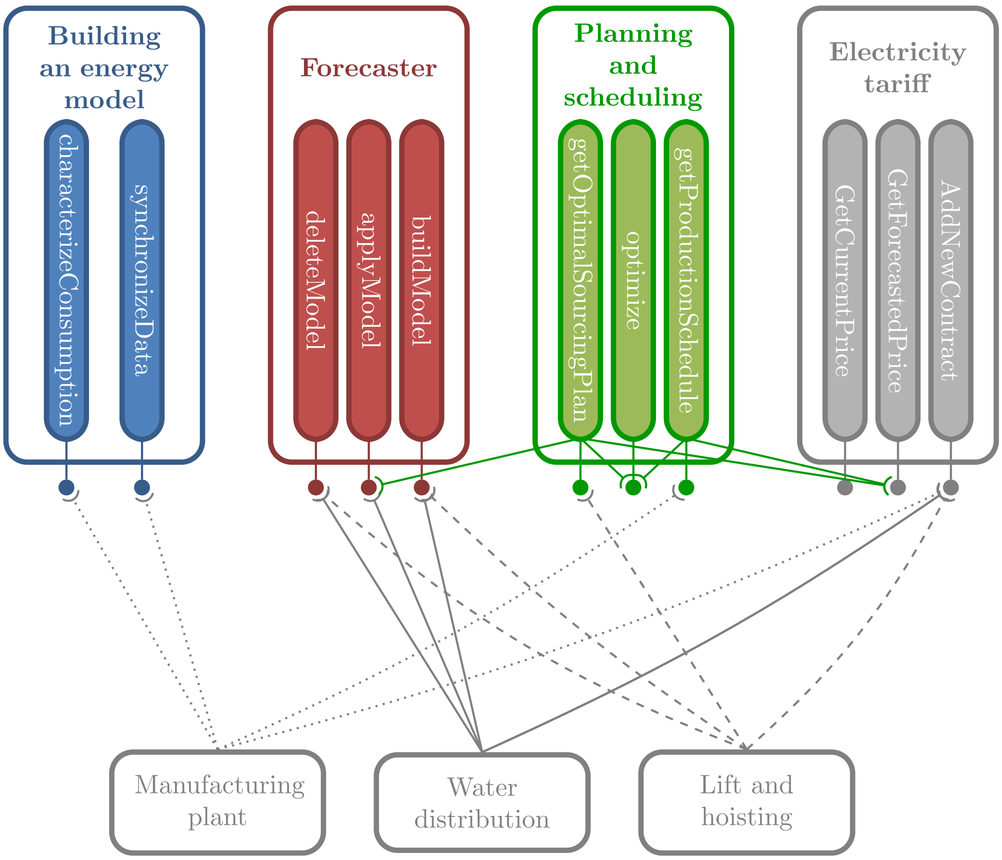


FIGURE 4.6 – Service oriented architecture of the three Arrowhead pilots.

cations in order to deport expansive computing. Moreover, a Service Oriented Architecture allows a viable business model. Indeed, the simplest part of the software solutions can be sold to customers at a fair price. For some customers who have small needs, that would be sufficient. For those who want to control and understand more finely their prosumer system, the most complex (and expansive part) of the algorithms can be rented as web services. Another advantage of this architecture is that the components are kept modular, and thus more easily maintainable. Finally, that allows applications and services to be developed in any preferred programming language (and not necessarily the same). On the other hand, the main drawback is that network lags and failures can impact the quality of service. This is why local controllers are never implemented as services but nested in on-site applications.

On Figure 4.6, we can see that four kinds of services were developed: those related to energy modeling, forecasting, scheduling, and electricity tariff. The energy modeling services are those explained in Subsection 3.2.2. There are quite specialized for the manufacturing domain. The forecast services are fully generic and can be used to build a forecasting model from any kind of time-series. They can actually be used by the three pilots. Among the scheduling services, there are two specialized strategic controllers (one solving sourcing problems, and one building production plans), and one generic service called "optimize". This generic service is composed of several optimization engines, and manages internally the optimization engines available for handling optimization demands. That allows a deported management of licenses and gives a unified optimization interface to all applications and scheduling services. Finally, the electricity tariff services maintain tariffs up to date and give a unified interface to request them. The electricity bill of power time-series can also be computed.

The software architecture described in this section, applied to the multisource elevator use-case, is presented on Figure 4.7. We can see that Local Controllers interact between them, while

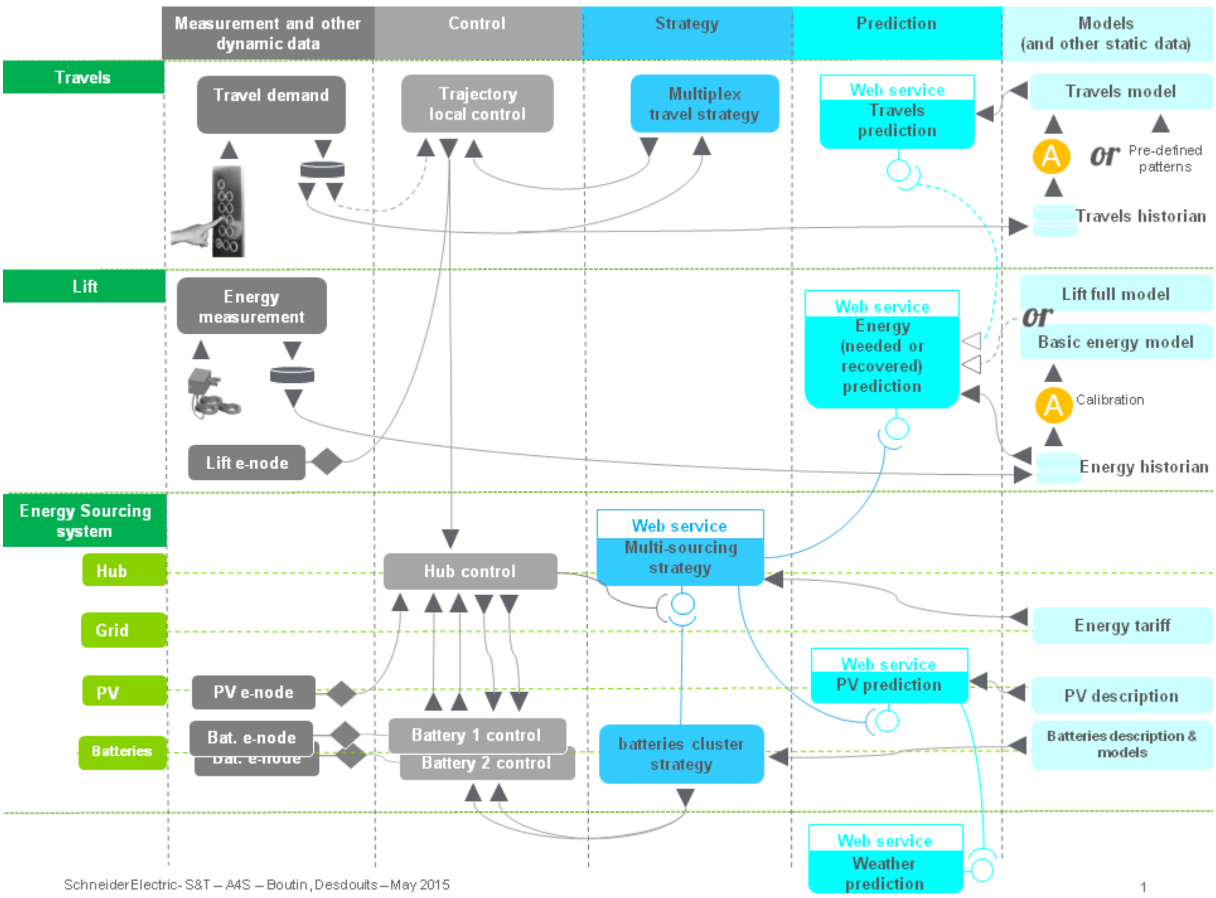


FIGURE 4.7 – Lift application software architecture.

each Local Controller also interact with its own Strategic Controller (possibly implemented as a web-service). Strategic Controllers use predictions coming from web-services, themselves fed by historical data.

4.3 Robustness and Performance Guarantee

The previous section explains how to control a prosumer system. In order for a prosumer system to be efficient, it has to be appropriately controlled (at the tactic and strategic levels), and appropriately designed. For example, designing the multisource elevator system consists in choosing the best storage units, electricity tariff, and controller. The elevator itself could also be chosen in case of design of a brand new system. Designing the manufacturing plant regarding its electricity consumption would be to choose the most appropriate electricity tariff, or even the best places and sizes of storage areas. Designing the water network in the energetic point of view would be to choose groups of pumps regarding their efficiency and the most profitable electricity tariff.

The notion of robustness can be defined in the following way at the different levels. The robustness of a Strategic Optimizer depends on the amount of disturbances its strategies can endure before being invalidated. The robustness of a local controller is its capacity to balance the prosumer system whatever the situation. Finally, a robust prosumer system design is a design that has always the expected performances whatever the situation, even in improbable use-cases.

The key point of designing a system is to be able to guarantee the performances of the proposed design. For that purpose, two questions arise: which Key Performance Indicators should be used to evaluate the design performances and how to guarantee, in an efficient way, those performances? The first idea coming to mind is to simulate the chosen system design over a representative number of days and to compute the average or the worst case value of the chosen KPI. But what is a representative number of days? This is why we propose to use a randomized algorithm and its associated number of samples needed to guarantee performances with a given probability. This algorithm, described in Section 2.5, is fed by a set of design parameters and a set of drawn samples and gives the best design among the combinations of design parameters value. Along with this best design, the associated KPI value obtained in the worst case over the drawn samples is given, as well as the guarantee with a given probability that the performances will not be worse than that. From that probability, the mean number of days over a year, when the design KPI will be worse than the certified value, can be computed and given to a customer.

Among the possible KPIs, we can cite the maximum power peak purchased from the grid, the economical (operational and investment) cost of the prosumer system, or even the capacity of a prosumer system to exploit and respect offers from a demand-response aggregator. On the other hand, the same randomized algorithm can be used to evaluate the robustness of a strategic optimizer. In that case, the KPI could be whether the strategy has to be repaired after some disruptions or not and the design parameters could be the amplitude of the disruptions considered.

4.4 Example of Demand-Response Application

In this section, an example of demand-response applied to the multisource elevator communicating with an aggregator is presented.

For an elevator in an office building, the strategic optimization results generally show multiple battery charging/discharging "*trend periods*" during the day. These trend periods are illustrated on Figure 4.8, on which they are separated by dashed black lines. A trend period

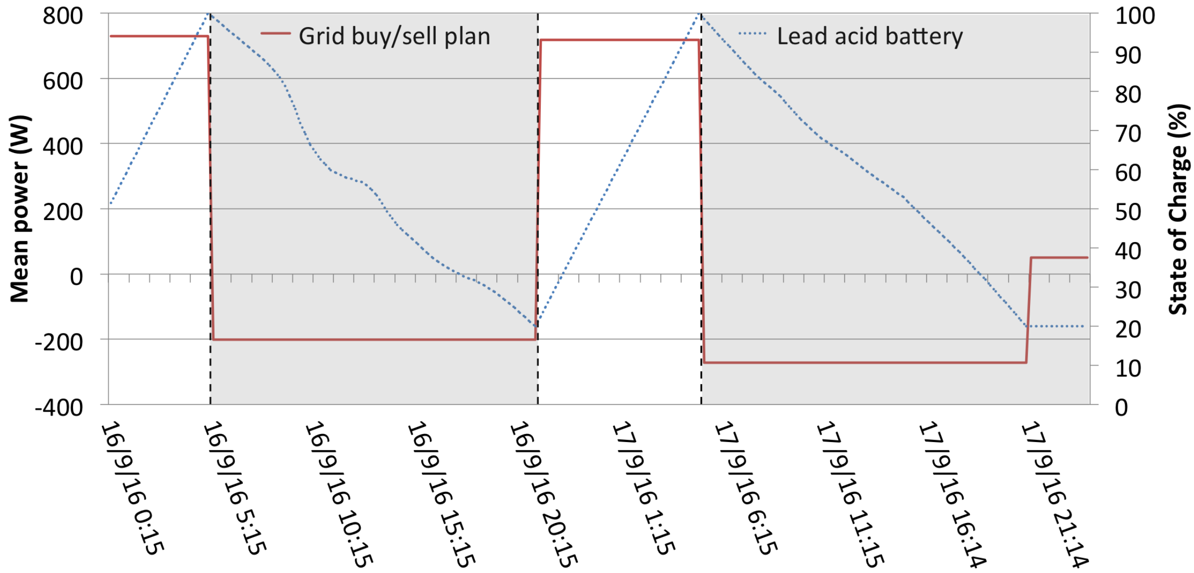


FIGURE 4.8 – Trend periods of the battery during a two days period.

is defined as a period over which the battery state of charge (looked at the time granularity of the SO) monotonically increases or monotonically decreases. In the late evening and early morning (off-peak), electricity is bought to charge the battery; this is the first trend period of the first day. During daytime, the battery is discharged as the elevator requires a lot of energy and the electricity tariff is high (on-peak); this is the second trend period of the first day, during which power power can even be sold to the grid (see Figure 4.8). The next day also has two trend periods following the same logic. Let us note that at the end of the second day, the state of charge of the battery is constant. This is considered being part of the previous "discharging" trend period.

If we look at each of these trend periods from the elevator point of view, we see that the exact profile followed to recharge or discharge the battery is not very important. The important point is to keep the trend (either increasing or decreasing) and reach the right level at the end of the period. In the SO model, a term of the objective function favors smoothing of the purchases and sales of energy. In many cases, the amount of energy purchased or sold is constant or almost constant over the identified trend periods. In reality, the grid operator or an aggregator could prefer to sell or buy more or less energy during the trend period, depending on global production and consumption of many actors.

For example, imagine the weather is such that the utility is expecting a production of a lot of Aeolian energy between 10pm and midnight. In such as case, it is more interesting to incite consumers to consume between 10pm and midnight: the batteries of the elevator could be fully recharged at that time rather than on the early morning. This could be done in two ways: a change of tariff decided on the day, or by asking flexible consumers how they could adapt, deciding how much they shall adapt, and rewarding them for their flexibility.

As part of the Arrowhead project, we collaborated with a team developing a system usable by an aggregator to receive and manage flexible energy consumption and production offers Šikšnys et al., 2015. We first identify trend periods, based on the plan generated by the SO. For each of these periods, we compute an envelope of the battery state of charge and grid purchase/sale curves, while guaranteeing:

1. that the battery charging/discharging trend is respected throughout the trend period;

2. that the planned value of the battery state of charge at the end of the trend period is attained;

3. that the maximal power of the battery and grid connection are respected.

The envelope of the possible grid power constitutes a flexible offer. To build a flexible offer object, we iterate over the τ intervals composing the trend period under consideration. For each such interval, we determine lower and upper bounds on grid consumption, taking into account the trend, the charging or discharging yield of the battery, and the power limits of the battery and grid connection. The flex-offer object is composed of these bounds for each τ interval and a global constraint enforcing that the total energy bought or sold will remain equal to the initial plan. Let us note that constraints on the minimal and maximal states of charge of the battery do not need to be taken into account as, in any solution, (i) the states of charge at the beginning and end of the trend period will be the same as in the initial SO plan, and (ii) the state of charge will monotonically increase or decrease between these two points.

Figure 4.9 illustrates a plan computed by the SO, the corresponding flexible offer that can be proposed to the aggregator, and the plan updated with the aggregator response. The red curve

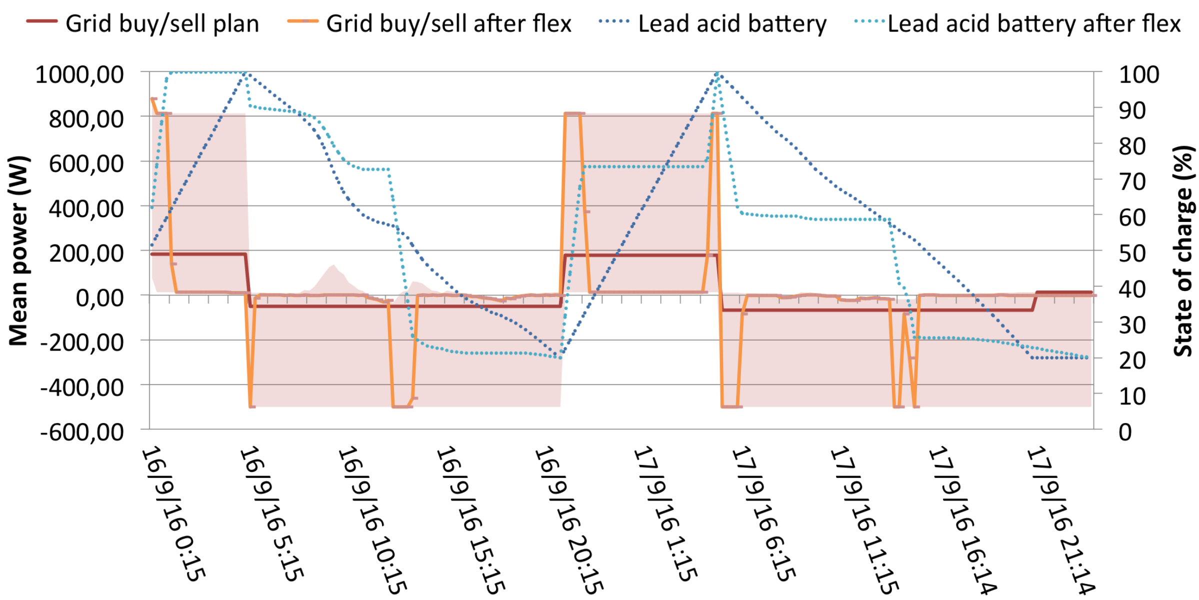


FIGURE 4.9 – Initial plan versus flex-offer of the aggregator.

(Grid buy/sell plan) and the dark blue dotted curve (Lead acid battery) constitute the initial plan computed by the SO. The red shadowed area represents the flexible offer: the envelop within which the grid power should stay to respect trend periods. Then, the orange curve (Grid buy/sell plan after flex) is the response of the aggregator regarding its preferred purchasing curve. The light blue dotted curve (Lead acid battery after flex) is the evolution of the state of charge of the battery, computed from the aggregator response. First, let us note that the initial plan is much smoother than the one proposed by the aggregator. Indeed, the aggregator has to deal with several consumers. Their consumption have to be scheduled at different times to minimize the global consumption peak. In this example the initial plan had an electricity bill of -1.01€ while the plan amended after the flexible offer has an electricity bill of -0.98€. However, a retribution of $-3,47$€ for our flexibility comes with the aggregator plan. That would fully compensate storage units usage cost and make the multisource elevator profitable, even with a simple peak/off-peak tariff.

Let us note that in practice, computing a flexible offer is equivalent to build the statement of a sub-problem of the SO problem, (i) limited to the given trend period, (ii) where all variables except the charging/discharging quantities, states of charge and purchase/sale quantities have been fixed, and (iii) with no optimization criterion. This statement can then be sent to the aggregator which uses it as a sub-problem of a more global problem, and optimizes its global problem using its own objective function. The solution of the sub-problem can then be sent back to the SO and used to update the plan. In practice, the aggregator from Šikšnys et al., 2015 is not expecting a formal sub-problem description, but is using an object-oriented representation of a flexible offer that is equivalent to a sub-problem of the SO problem.

Let us conclude that such a demand-response scenario can be put in place for any energy system including a SO, a battery and a connection to the grid.

Chapter 5

Conclusion

During this PhD thesis, we proposed efficient resolution methods for three optimization problems from the industry, two of them are reported in this manuscript. These problems have different scale, different context and different constraints, but they have a common point: the energy consumption has to be controlled.

The multisource elevator problem was formalized, then solved using two coupled controllers to handle real-time and economical dimensions of the problem. The first controller is a linear problem which, once solved, gives a sourcing strategy for the next twenty-four hours, for example. This sourcing strategy is composed of target state of charge for storage units and mean power for controllable prosumers, and minimizes the electricity bill as well as the storage units aging cost. The low-level controller is a real-time rule-based controller using a depth-first search in a tree of flexibilities to compute the best trade-off between the strategic instruction received and the current situation. This two-levels approach was generalized to allow its application to a wide class of small sourcing problems, with a few additions in case of slightly different prosumers. Experiments were conducted to highlight the impact of the two-level strategy on the accuracy of the computed savings. Moreover, a randomized algorithm was proposed to design the system by choosing storage units, electricity tariff and controllers, and to be able to guarantee the performances of this design.

It would have been worthy to experiment the coupled controllers on a real elevator. This would have allowed to get more sensor data, to refine the statistical model built and to check the accuracy of the call answering tactic implemented. Experiments on multiple elevators sharing the same storage units would also have been interesting. A strategic controller would have been needed to dispatch travels between the elevators and the interactions between this strategic controller and the sourcing strategic controller would have been of a great interest. The aging model of the battery could have been improved with a more complex model, or with a more accurate coefficient computed after a great number of battery aging experiments. Finally, different elevator models (varying counterweight for example) could have been compared to evaluate their impact on the savings. We did not have time nor means to perform all the experiments mentioned above, but they are valuable research tracks for future work.

The manufacturing plant use-case allowed us to put in place the whole process needed to optimize a real prosumer, from sensors to optimization strategy. From the energy measurements, data mining, and energy modeling processes, we learned that in case production data are not accurate, it is probably better to use finite state machines to model energy consumption of production machines. But in case accurate production data are available, our method to model activities energy consumption is really faster to put in place. This remark is also valid in other contexts to choose between simplified energy model and more complex process-based energy model. On the other hand, the scheduling problem was formalized as a generalized (preemptive) flow-shop with precedence, due dates, and a trade-off between tardiness, inventory and energy objectives. In particular the electricity cost functions considered are a set of piecewise-linear functions of the power used; one function per time bucket. Three MILP formulations were proposed to model the scheduling problem, each of them with several modeling approximations to keep the problem linear. The first formulation is event-based but considers only linear electricity cost function and approximates the storage cost. The second formulation models piecewise-linear energy cost functions, but is time-based and thus is pessimistic regarding resource constraints (including power). The third formulation proposed is also time-based, allows preemption and considers precedence constraints as constraints on the amount of stock of the intermediate material. Although this approach could be interesting, we did not have time to handle all the related issues and we did not use this formulation in the experiments. Finally, a local search algorithm and a naive Constraint Satisfaction Problem have also been implemented in order to facilitate the resolution of the complex scheduling problem considered. Experiments have been conducted on Benchmark instances from the literature, as well as on four instances built with data from the real plant. Results on the Benchmark show that only a small percentage of the total cost can be saved by optimizing energy, and not for the biggest instances. Fortunately, better results are achieved on the instances built from the real plant data, and savings are made on the electricity cost, without any degradation of the tardiness cost.

There are four main points we wish to have been able to dig into. The first one is comparing finite state machines energy models of production machines to our regression-based energy models of activities. The second one is an event-based formulation including our linear modeling of piecewise-linear energy cost functions. The third one is a functional preemptive storage-oriented formulation, based on the principles given in Subsection 3.4.3, but solving the issues discussed in the same subsection. The last one concerns improvement of the local search procedure. Indeed, the first solution feeding the local search could be optimized briefly by the Overlap formulation without energy cost. Moreover, new local search operators, based on the time-based formulations could be implemented.

The third use-case studied is about computing optimized pumping plans for a drinking water network. We did not investigate much this use-case, but a quadratically constrained formulation was used to solve the optimization problem, taking into account hydraulic constraints. Two simulation models of the water network were proposed and experiments were run to evaluate the feasibility and potential gains of the optimized pumping plans compared to the Business As Usual plans. No outlook are considered regarding this use-case, as Gratien Bonvin is currently doing his PhD thesis on this subject (cf Bonvin, Samperio, et al., 2015; Bonvin, Demassey, et al., 2016). But research work on this use-case helped us to better understand how to manage the coupling between optimization and simulation engines.

The savings achieved on the electricity bill go from 10.9% to 114%, depending on the use-case considered. The gain in euros is small for the multisource elevator (114% $\times$ 0.59 € per day $\simeq$ 245 € per year) but potentially multiplied by a big number of elevators. Moreover, the optimization proposed in this case can be handled in a fully automatic way. For the manufacturing plant, potential gains are bigger (10.9% $\times$ 1M€ = 109000 € per year) but the specific modeling to be done for each plant is significant. Furthermore, production plans have to be applied by human resources, or a fully automatic production system has to be put in place (with local controllers) and that complicates the optimized production plans application. In a water network, the application of pumping plans can be (quite) easily put in place, and could bring 15% savings. But the interaction between pumping plans and local control of pumps has to be carefully investigated.

From those three use-cases, we extracted guidelines on software architecture and algorithms that shall be used when considering a (reasonably small) problem of prosumers and their electricity bill in a centralized way. In case there are too many prosumers to solve a global problem, either the problem should be split at different levels and the interactions between these levels should be defined, or decentralized problems should be defined in each prosumer and the convergence of the global system should be studied. See Pflaum, Alamir, and Lamoudi, 2014 for an example of hierarchical architecture for a district. Whatever the optimization method used, optimizing the electricity bill at the consumer side allows aggregators and electricity transmission system operators to minimize electricity consumption peaks. That could profit to all the actors by saving money, and to the earth by reducing the CO2 emission. Depending on the evolution of electricity prices, demand-response incentives, and regulations, the use-cases considered in this thesis might become worth implementing.

Bibliography

Alamo, Teodoro, Roberto Tempo, Amalia Luque, and Daniel R Ramirez (2015). "Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms". In: *Automatica* 52, pp. 160–172.

Albadi, Mohamed H and EF El-Saadany (2008). "A summary of demand response in electricity markets". In: *Electric power systems research* 78.11, pp. 1989–1996.

Applegate, David and William Cook (1991). "A computational study of the job-shop scheduling problem". In: *ORSA Journal on computing* 3.2, pp. 149–156.

Artigues, Christian, Pierre Lopez, and Alain Haït (2009). "Scheduling under energy constraints". In: *Proceedings of IESM 2009 in Montreal, Canada*.

— (2013). "The energy scheduling problem: Industrial case-study and constraint propagation techniques". In: *International Journal of Production Economics* 143.1, pp. 13–23.

Baptiste, Philippe, Claude Le Pape, and Wim Nuijten (2001). *Constraint-based scheduling: applying constraint programming to scheduling problems*. Springer.

Based on IEA data from IEA Statistics © OECD/IEA 2014 (www.iea.org/statistics), Licence: www.iea.org/t&c; as modified by The World Bank. (2014a). *Electric power consumption (kWh per capita)*. The World Bank.

— (2014b). *Electricity production sources*. The World Bank.

Benoist, Thierry, Eric Bourreau, Yves Caseau, and Benoît Rottembourg (2001). "Towards stochastic constraint programming: A study of onine multi-choice knapsack with deadlines". In: *Seventh International Conference on Principles and Practice of Constraint Programming*. Springer, pp. 61–76.

Ben-Tal, Aharon, Laurent El Ghaoui, and Arkadi Nemirovski (2009). *Robust optimization*. Princeton University Press.

Bilbao, Endika and Philippe Barrade (2012). "Optimal Energy Management of an improved elevator with energy storage capacity based on Dynamic Programming". In: *Energy Conversion Congress and Exposition (ECCE)*. Raleigh, USA: IEEE, pp. 3479–3484.

Bonvin, Gratien, Sophie Demassey, Claude Le Pape, Nadia Maïzi, Vincent Mazauric, and Alfredo Samperio (2016). "A convex mathematical program for pump scheduling in a class of branched water networks". In: *Applied Energy*.

Bonvin, Gratien, Alfredo Samperio, Claude Le Pape, Vincent Mazauric, Sophie Demassey, and Nadia Maïzi (2015). "A heuristic approach to the water networks pumping scheduling issue". In: *Energy Procedia* 75, pp. 2846–2851.

Boutin, Veronique and Jean-Louis Bergerand (2013). "Water networks contribution to demand response". In: *PowerTech 2013 Conference*. Grenoble: IEEE.

Boutin, Véronique, Chloé Desdouits, Maxime Louvel, Francois Pacull, Maria Isabel Vergara-Gallego, Oussama Yaakoubi, Cédric Chomel, Quentin Crignon, Christophe Duhoux, Denis Genon-Catalot, Laurent Lefevre, Thanh Hung Pham, and Van Thang Pham (2014). "Energy optimisation using analytics and coordination, the example of lifts". In: *Emerging Technology and Factory Automation (ETFA)*. IEEE. Barcelona, Spain, pp. 1–8.

Brucker, Peter (2007). *Scheduling Algorithms*. Springer-Verlag Berlin Heidelberg.

Brucker, Peter, Andreas Drexl, Rolf H. Möhring, Klaus Neumann, and Erwin Pesch (1999). "Resource-constrained project scheduling: Notation, classification, models, and methods". In: *European Journal of Operational Research* 112.1, pp. 3–41.

Brucker, Peter and Sigrid Knust (1999). “Complexity results for single-machine problems with positive finish-start time-lags”. In: *Computing* 63.4, pp. 299–316.

Castro, Pedro M, Iiro Harjunkoski, and Ignacio E Grossmann (2011). “Optimal scheduling of continuous plants with energy constraints”. In: *Computers & chemical engineering* 35.2, pp. 372–387.

Danna, Emilie (2004). “Intégration des techniques de recherche locale à la programmation linéaire en nombres entiers”. PhD thesis. Université d’Avignon et des pays du Vaucluse.

Danna, Emilie and Laurent Perron (2003). “Structured vs. unstructured large neighborhood search: A case study on job-shop scheduling problems with earliness and tardiness costs”. In: *International Conference on Principles and Practice of Constraint Programming*. Springer, pp. 817–821.

Danna, Emilie, Edward Rothberg, and Claude Le Pape (2003). “Integrating mixed integer programming and local search: A case study on job-shop scheduling problems”. In: *Fifth International Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimisation Problems (CP-AI-OR’2003)*. Citeseer, pp. 65–79.

Desdouits, Chloé, Mazen Alamir, Veronique Boutin, and Claude Le Pape (2015). “Multi-source elevator energy optimization and control”. In: *European Control Conference (ECC)*. EUCA. Linz, Austria.

Desdouits, Chloé, Mazen Alamir, Rodolphe Giroudeau, and Claude Le Pape (in press). “Certification under Uncertainties of Control Methods for Multisource Elevators”. In: *16th International Conference on Intelligent Systems Design and Applications (ISDA)*.

Desdouits, Chloé, Mazen Alamir, Rodolphe Giroudeau, and Claude Le Pape (in press). “The Sourcing Problem, Energy Optimization of a Multisource Elevator”. In: *13th International Conference on INformatics in COntrol, automation and robotics (ICINCO 2016)*. Lisbon, Portugal.

Desdouits, Chloé, Jean-Louis Bergerand, Pierre-Alexis Berseneff, Claude Le Pape, and Dimitri Yanculovici (in press). “Energy study of a manufacturing plant”. In: *ECEEE Industrial Efficiency Summer Study*. Berlin, Germany.

Dietmair, A and A Verl (2009). “Energy consumption forecasting and optimisation for tool machines”. In: *Energy* 62, p. 63.

Drummond, Mark, Keith Swanson, and John Bresina (1994). “Robust scheduling and execution for automatic telescopes”. In: *Zweben and Fox (eds.) Intelligent Scheduling, Morgan Kaufmann*, pp. 629–654.

Erschler, Jacques, Pierre Lopez, and Catherine Thuriot (1991). “Raisonnement temporel sous contraintes de ressource et problèmes d’ordonnancement”. In: *Revue d’intelligence artificielle* 5.3, pp. 7–32.

Fysikopoulos, Apostolos, Georgios Pastras, Theocharis Alexopoulos, and George Chryssolouris (2014). “On a generalized approach to manufacturing energy efficiency”. In: *The International Journal of Advanced Manufacturing Technology* 73.9-12, pp. 1437–1452.

Garey, Michael R and David S Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*.

German, Grigori, Chloé Desdouits, and Claude Le Pape (2015). “Energy Optimization in a Manufacturing Plant”. In: *ROADEF annual conference*. ROADEF.

GNU Linear Programming Kit, Version 4.54 (2014). GNU.

Gonzalez, Teofilo and Sartaj Sahni (1978). “Flowshop and jobshop schedules: complexity and approximation”. In: *Operations research* 26.1, pp. 36–52.

Graham, Ronald L, Eugene L Lawler, Jan Karel Lenstra, and AHG Rinnooy Kan (1979). “Optimization and approximation in deterministic sequencing and scheduling: a survey”. In: *Annals of discrete mathematics* 5, pp. 287–326.

Graves, Stephen C (1981). “A review of production scheduling”. In: *Operations research* 29.4, pp. 646–675.

Haït, Alain and Christian Artigues (2011). "Un modèle PLNE à temps continu pour l'ordonnancement d'une aciérie avec régulation de consommation d'énergie". In: *12ème congrès annuel de la Société française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2011)*, p. 271.

Haouassi, Mustapha, Chloé Desdouits, Rodolphe Giroudeau, and Claude Le Pape (in press). "Production scheduling with a piecewise-linear energy cost function". In: *2016 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2016)*.

Herroelen, Willy, Bert De Reyck, and Erik Demeulemeester (1998). "Resource-constrained project scheduling: a survey of recent developments". In: *Computers & Operations Research* 25.4, pp. 279–302.

Hofmann, Markus and Ralf Klinkenberg (2013). *RapidMiner: Data mining use cases and business analytics applications*. CRC Press.

Johnson, Thomas Joel Russell (1967). "An algorithm for the resource constrained project scheduling problem". PhD thesis. Massachusetts Institute of Technology.

Lamoudi, Mohamed Yacine, Mazen Alamir, and Patrick Béguery (2011). "Unified NMPC for multi-variable control in smart buildings". In: *IFAC Proceedings Volumes* 44.1, pp. 11024–11029.

Le Pape, Claude (1994). "Implementation of resource constraints in Ilog Schedule: a library for the development of constraint-based scheduling systems". In: *Intelligent Systems Engineering* 3.2, pp. 55–66.

Le Pape, Claude and Anna Robert (2007). "Jeux de données pour l'évaluation d'algorithmes de planification et ordonnancement". In: *Livre des résumés séléctionnés lors de la conférence conjointe FRANCORO V/ROADEF*. Vol. 20, p. 23.

Le, Cao Vinh, Chee Khiang Pang, Oon Peen Gan, Xiang Min Chee, Dan Hong Zhang, Ming Luo, Hian Leng Chan, and Frank L Lewis (2013). "Classification of energy consumption patterns for energy audit and machine scheduling in industrial manufacturing systems". In: *Transactions of the Institute of Measurement and Control* 35.5, pp. 583–592.

Louvel, Maxime and François Pacull (2014). "LINC: A Compact Yet Powerful Coordination Environment". In: *Coordination Models and Languages: 16th IFIP WG 6.1 International Conference, COORDINATION 2014, Held as Part of the 9th International Federated Conferences on Distributed Computing Techniques, DisCoTec 2014, Berlin, Germany, June 3-5, 2014, Proceedings*. Ed. by Eva Kühn and Rosario Pugliese. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 83–98.

Maravelias, Christos T and Ignacio E Grossmann (2004). "A hybrid MILP/CP decomposition approach for the continuous time scheduling of multipurpose batch plants". In: *Computers & chemical engineering* 28.10, pp. 1921–1949.

Masmoudi, Oussama, Alice Yalaoui, Yassine Ouazene, and Hicham Chehade (2015). "Lot-sizing in flow-shop with energy consideration for sustainable manufacturing systems". In: *IFAC-PapersOnLine* 48.3, pp. 727–732.

Nattaf, Margaux, Christian Artigues, and Pierre Lopez (2015). "A hybrid exact method for a scheduling problem with a continuous resource and energy constraints". In: *Constraints* 20.3, pp. 304–324.

Nattaf, Margaux, Christian Artigues, Pierre Lopez, and David Rivreau (2016). "Energetic reasoning and mixed-integer linear programming for scheduling with a continuous resource and linear efficiency functions". In: *OR Spectrum* 38.2, pp. 459–492.

Ngueveu, Sandra Ulrich, Christian Artigues, and Pierre Lopez (2016). "Scheduling under a non-reversible energy source: An application of piecewise linear bounding of non-linear demand/cost functions". In: *Discrete Applied Mathematics* 208, pp. 98–113.

Paire, Damien, Marcelo Godoy Simoes, Jérémy Lagorse, and Abdellatif Miraoui (2010). "A real-time sharing reference voltage for hybrid generation power system". In: *Industry Applications Society Annual Meeting (IAS)*. Houston, USA: IEEE, pp. 1–8.

Palpant, Mireille, Christian Artigues, and Philippe Michelon (2004). “LSSPER: solving the resource-constrained project scheduling problem with large neighbourhood search”. In: *Annals of Operations Research* 131.1-4, pp. 237–257.

Pecero, Johnatan E., Héctor Joaquin Fraire Huacuja, Pascal Bouvry, Alejandro Santiago Pineda, Mario César López Locés, and Juan Javier González Barbosa (2012). “On the energy optimization for precedence constrained applications using local search algorithms.” In: *HPCS*. Ed. by Waleed W. Smari and Vesna Zeljkovic. IEEE High Performance Computing and Simulation (HPCS), 2012 International Conference in Madrid, pp. 133–139.

Pesek, Igor, Andrea Schaerf, and Janez Žerovnik (2007). *Hybrid local search techniques for the resource-constrained project scheduling problem*. Springer.

Pflaum, Peter, Mazen Alamir, and Mohamed Yacine Lamoudi (2014). “Comparison of a primal and a dual decomposition for distributed MPC in smart districts”. In: *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*. IEEE, pp. 55–60.

Pritsker, A Alan B, Lawrence J Waiters, and Philip M Wolfe (1969). “Multiproject scheduling with limited resources: A zero-one programming approach”. In: *Management science* 16.1, pp. 93–108.

Prud’homme, Charles, Jean-Guillaume Fages, and Xavier Lorca (2016). *Choco Documentation*. TASC, INRIA Rennes, LINA CNRS UMR 6241, COSLING S.A.S. URL: http://www.choco-solver.org.

Sachs, Harvey M (2005). *Opportunities for elevator energy efficiency improvements*. Tech. rep. American Council for an Energy-Efficient Economy Washington, DC.

Sampson, Scott E and Elliott N Weiss (1993). “Local search techniques for the generalized resource constrained project scheduling problem”. In: *Naval Research Logistics (NRL)* 40.5, pp. 665–675.

Šikšnys, Laurynas, Emmanouil Valsomatzis, Katja Hose, and Torben Bach Pedersen (2015). “Aggregating and disaggregating flexibility objects”. In: *IEEE Transactions on Knowledge and Data Engineering* 27.11, pp. 2893–2906.

Sugimori, Y., K. Kusunoki, F. Cho, and S. Uchikawa (1977). “Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system”. In: *International Journal of Production Research* 15.6, pp. 553–564.

The MathWorks Inc. (2015). *MATLAB 2015b*. The MathWorks Inc., Natick, Massachusetts, United States.

Tominaga, Shinji, Ikuro Suga, Hiroshi Araki, Hiroyuki Ikejima, Makoto Kusuma, and Kazuyuki Kobayashi (2002). “Development of energy-saving elevator using regenerated power storage system”. In: *Proceedings of the Power Conversion Conference (PCC)*. Vol. 2. IEEE. Osaka, Japan, pp. 890–895.

Zhang, Xiandong (2010). “Scheduling with Time Lags”. PhD thesis. Erasmus University Rotterdam.