

UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des Sciences
Département d'Informatique

Éclairage public

HEREMAN Nicolas,
VAN BRANDE Rodrigue,
HUBLET Magali,
VANBERGEN Julien.

Superviseurs : Labbe Martine, Porretta Luciano

Abstract

Ce rapport présente ...

Contents

1	Introduction	2
2	État de l'art	3
3	Méthodes implémentées	4
3.1	Modélisation du problème	4
3.2	Résolution à l'aide d'un solver	4
3.3	Test de toutes les solutions	5
3.4	Une solution moins optimale	5
4	Résultats expérimentaux	6
5	Discussion	7
6	Conclusion et perspectives	8
	Bibliographie	8

Chapter 1

Introduction

L'éclairage public est un terme général représentant l'ensemble des moyens mis en place afin d'illuminer les espaces publics. C'est un domaine qui ne doit pas être sous-estimé car il a de grandes conséquences sur notre vie à tous, que ce soit au niveau de sa qualité ou de la sécurité. [?]

Malheureusement, cela a un certain coût, aussi bien financier qu'énergétique, et les risques de pénurie d'électricité ne font qu'augmenter. Tout cela nous indique qu'il devient urgent de réaliser des économies dans ce domaine. En effet, en moyenne, en Belgique, ce coût s'élève à 53% de la consommation électrique à la charge d'une commune. De plus, selon l'Ademe [?], l'éclairage actuel pourrait être très coûteux pour le financement public. Ainsi, une amélioration de l'efficacité énergétique pourrait réduire la facture de moitié.

Depuis plusieurs années, les pouvoirs publics expérimentent des extinctions d'éclairage à certaines heures de la nuit afin de réduire ces coûts. Cependant, l'éclairage est un élément essentiel à notre sécurité, c'est pourquoi nous nous devons de trouver d'autres moyens d'économie.

Un éclairage possède un placement dit "optimisé" lorsqu'il s'adapte au contexte, c'est à dire au terrain dans lequel il se trouve. En effet, certains endroits demandent plus de lumière que d'autres. Par exemple, les chemins passants au bord de l'eau. Chaque année, plusieurs personnes meurent noyées car les endroits au bord de l'eau sont peu ou pas éclairés. Mais la pollution lumineuse a un impact très négatif sur les animaux. Par exemple, les abeilles sont fortement attirées par la lumière. Ces petites ouvrières s'activent à la première lueur aperçue. C'est l'une des nombreuses causes de leur disparition près des zones rurales car elles meurent de fatigue ! Il ne suffit pas de mettre des lampes partout ! Pour des soucis écologiques, on souhaiterait utiliser le moins de lampes possibles.

Le but de notre projet paraît donc simple ; nous devons placer un nombre minimum de lampadaires dans un espace donné afin d'éclairer la zone de manière optimale.

Nous utiliserons

Chapter 2

État de l'art

Chapter 3

Méthodes implémentées

Afin de pouvoir trouver des méthodes pour résoudre le problème, il est important de le définir correctement. Ce que nous voulons, c'est placer un certain nombre de lampadaires de différentes puissances afin d'obtenir une luminosité le plus proche possible de ce qui est demandé.

3.1 Modélisation du problème

Pour pouvoir résoudre le problème, nous devons d'abord le modéliser. Nous divisons l'endroit à éclairer sous forme de matrice dont nous remplissons chaque case d'un nombre représentant la luminosité voulue à cet emplacement.

La taille de cette matrice est en lien direct avec la précision des résultats et du temps de calcul. Lorsque le nombre de cases augmente, leur taille diminue. Chaque case représente donc une partie plus précise de l'endroit à éclairer. Le résultat sera donc lui aussi plus précis. Mais si le nombre de case augmente, le nombre de solution possible et donc de calcul à effectuer augmente aussi. Le temps d'exécution sera donc plus long.

Pour pouvoir trouver le meilleur positionnement possible de lampadaire, il faut savoir calculer la luminosité qu'ils apportent selon leurs emplacements. Pour cela, nous nous sommes servis du modèle présenté par Noor-E-Alama dans son article dans l'European Journal of Operational Research.

$$S_{in} = \frac{P_n}{r^2} \cos \left(\tan^{-1} \left(\frac{|i-x_n|}{r} \right) \right)$$
$$S_{jn} = \frac{P_n}{r^2} \cos \left(\tan^{-1} \left(\frac{|j-y_n|}{r} \right) \right)$$

La première formule calcule la luminosité apportée par le $n^{\text{ième}}$ lampadaire sur la $i^{\text{ième}}$ colonne. La deuxième est identique mais pour la $j^{\text{ième}}$ ligne. Les variables x_n et y_n sont les coordonnées du $n^{\text{ième}}$ lampadaire, P_n est sa puissance et r la hauteur de ces lampadaires.

La luminosité apportée par le $n^{\text{ième}}$ lampadaire sur une case est égal à la plus petite variable entre S_{in} et S_{jn} . La luminosité totale d'une case est la somme des luminosités apportées par les différents lampadaires. On a donc:

$$S_{ijn} = \min(S_{in}, S_{jn})$$
$$S_{ij} = \sum S_{ijn}$$

3.2 Résolution à l'aide d'un solver

Notre première implémentation a été de faire appel à un solver. Pour ce projet, Knitro a été utilisé. Il permet de résoudre des problèmes non linéaire utilisant des variables entières et continues.

Afin de faire appel à ce solver, le problème a été formulé dans un langage informatique nommé AMPL. Ce langage utilise un fichier data définissant les paramètres tel que la taille de la matrice, la demande, la puissance maximale des lampadaires, ... Le problème lui même est décrit dans un fichier modèle (.mod). Il y est décrit l'objectif du problème ainsi que ces contraintes.

Pour le résoudre, knitro utilise un algorithme branch and bound (Séparation et évaluation en français).

3.3 Test de toutes les solutions

Une autre méthode de résolution est d'essayer toutes les solutions possibles et de garder la meilleure. Pour cela un algorithme de backtraking est utilisé. Il va placer des lampadaires aux différents emplacements possibles et garder la solution qui apportera la luminosité la plus proche de la demande. Cette façon de faire étant très gourmande, il a été décidé de ne pas tenir compte des différentes puissances et d'en définir une fixe pour cette algorithme. Afin d'apporter un autre intérêt à cette méthode, certaines cases sont bloquées et on ne peut pas y placer de lampadaire. Cela s'avère utile pour éviter d'en placer au milieu d'un lac par exemple.

3.4 Une solution moins optimale

Le problème avec la méthode précédente c'est que lorsque le problème grandit, le nombre de solution augmente énormément et donc le temps de calcul aussi. Il y a des cas où on ne veut pas spécialement la meilleure solution et qu'il suffit qu'elle soit assez proche de l'optimale. Cela permet de gagner du temps de calcul.

C'est ce que fait le dernier algorithme présenté ici. Il est très proche du deuxième sauf qu'il s'arrête dès qu'il trouve une solution acceptable. Ce qu'il trouve acceptable dépend d'un paramètre qu'on lui donne qui est un pourcentage d'écart avec la demande.

Si jamais il n'existe aucune solution dans cet écart, l'algorithme va donner la meilleur solution qu'il aura trouver. Dans ce cas, il agit exactement comme le deuxième.

Chapter 4

Résultats expérimentaux

Chapter 5

Discussion

Les résultats obtenus dans la partie AMPL sont les meilleurs car la puissance des lampadaires est variable et on s'approche très près de la matrice de demande. Ces résultats sont dus aux équations décrites dans l'article scientifique et surtout à la puissance variable des lampadaires. Cette variabilité des puissances permet de couvrir au mieux la zone et de s'adapter à ces besoins.

Dans les algorithmes d'énumération, la puissance fixe permet de trouver une solution plus rapidement car on ne doit pas chercher quelle puissance est la meilleure. Pour des tailles de tableaux plus grands, l'énumération devient vite impossible car il y a beaucoup trop de cas à prendre en compte¹. C'est pour cela qu'un algorithme d'énumération s'arrêtant à un pourcentage donné de satisfaction a été donné. Cela permet d'avoir un résultat correct sans prendre trop de temps.

¹Total = largeur*hauteur*nombre de lampadaire

Chapter 6

Conclusion et perspectives

En conclusion, nous avons un logiciel qui permet de trouver l'emplacement optimal des lampadaires. D'une part, via un modèle mathématique qui prend en compte des puissances variables; et d'autre part, via une énumération exhaustive qui a une puissance fixe mais des zones bloquantes.

Dans le futur, avec plus de contraintes et certaines optimisations pour gagner du temps de calcul, on pourrait utiliser ce genre de logiciel pour éclairer les routes, parc, villes, etc. . . afin de permettre aux gens de circuler en sécurité dans les lieux publics. Pour cela, il suffirait d'ajouter des contraintes pour bloquer des zones, prendre en compte les heures les plus fréquentées.