

Évolution différentielle hybride pour le problème de placement optimal de caméras

Mathieu Brévilliers¹ Julien Lepagnot¹ Lhassane Idoumghar¹
Maher Rebai² Julien Kritter¹

¹ IRIMAS, Université de Haute-Alsace, Mulhouse, France

{mathieu.brevilliers, julien.lepagnot, lhassane.idoumghar, julien.kritter}@uha.fr

² École Supérieure d'Ingénieurs Léonard de Vinci, Paris, France

maher.rebai@devinci.fr

Mots-clés : *métaheuristiques, évolution différentielle, algorithmes hybrides, placement optimal de caméras, couverture à coûts unitaires*

Introduction

Les réseaux de caméras sont de plus en plus communs et stimulent de plus en plus d'intérêt à la fois dans le monde académique et auprès du public. Couplés à des systèmes d'analyse d'image et de comportement, ces réseaux peuvent permettre d'identifier certains événements ou risques efficacement. Cependant, en raison de leurs coûts de mise en place, il est devenu primordial de considérer ce problème du point de vue de l'optimisation. L'approche discrète à ce problème est assez courante dans la littérature [3] et implique plusieurs avantages. Elle permet notamment de transformer l'application en un problème combinatoire courant : celui de la couverture. Ainsi, dans notre approche, nos instances constitueront des graphes bipartis entre les caméras, représentées par leur positions et orientations (x, y, z, α, β) , et les points à surveiller dans l'espace 3D. Pour résoudre ce problème \mathcal{NP} -complet, nous présentons un algorithme hybride qui combine l'évolution différentielle (DE) et un Branch-and-Cut (CPLEX).

1 L'évolution différentielle

L'idée de DE [5] est d'exploiter une population de solutions et leurs *différences* pour explorer leur voisinage. Dans sa forme la plus simple, l'algorithme génère, pour chaque individu x_c , un *mutant* x_m à partir de trois individus x_1 , x_2 et x_3 distincts, selon l'équation (1), dans laquelle i indexe chaque variable de la solution et F est un paramètre de l'algorithme :

$$x_m^i = x_1^i + F(x_2^i - x_3^i) \quad (1)$$

Pour un problème discret, ces opérations doivent être ajustées afin de conserver un sens. Les approches courantes utilisent des opérations ensemblistes plutôt qu'arithmétiques [4]. Une fois la mutation appliquée, un croisement et une sélection ont lieu entre chaque mutant et son individu parent. L'individu dit *d'essai* x_e résultant est donné par l'équation (2), dans laquelle CR est un paramètre de l'algorithme et i_{bias} constitue un biais permettant de porter au moins une variable mutante à la génération suivante.

$$x_e^i = \begin{cases} x_m^i, & \text{si } rand(0, 1) \leq CR \text{ ou } i = i_{bias} \\ x_c^i, & \text{sinon} \end{cases} \quad (2)$$

2 DE et Branch-and-Cut pour le placement de caméras

Pour notre problème de placement de caméras, deux changements sont introduits. Tout d’abord, la mutation exploite les similarités entre caméras. Une fois les individus x_1 , x_2 et x_3 sélectionnés, le mutant est construit selon l’équation (3) suivante :

$$x_m^i = x_1^i + F(x_2^u - x_3^v) \quad (3)$$

où u (resp. v) représente l’indice de la caméra la plus similaire à x_1^i dans x_2 (resp. x_3). Cette mesure est donnée par la distance euclidienne des vecteurs de représentation des caméras. La seconde modification implique CPLEX pour la phase de croisement. Dans notre approche, l’équation (2) n’est plus utilisée. Une fois un mutant x_m sélectionné pour x_c , un sous-problème est défini par l’union des caméras de x_m et x_c puis résolu par CPLEX. La sélection est ensuite effectuée de façon habituelle entre x_c et x_e . Cette approche a été testée sur un jeu de 18 instances existantes [1] et comparée avec CPLEX seul et RWLS [2], un algorithme de l’état de l’art pour la couverture. Les instances et résultats qui permettent de discriminer entre les algorithmes sont présentés dans la Table 1. Les algorithmes ont tous été limités à 1000 secondes.

Instance	Points	Caméras	Densité	CPLEX	RWLS	DEsim
5	13 005	9 852	17.0	7	7.03	7.53
6	18 605	16 732	12.3	—	10.53	10.97
7	32 805	35 291	7.2	—	18.07	19.50
8	51 005	60 251	4.8	—	28.69	30.80
9	605	1 672	6.8	21	20.00	21.07
10	2 205	7 352	2.0	71	67.77	70.97
11	4 805	17 032	0.9	—	151.67	152.13
12	8 405	30 712	0.5	—	270.63	268.27
13	13 005	48 392	0.4	—	428.33	424.47
14	18 605	70 072	0.2	—	626.80	612.33
15	32 805	125 431	1.4e-3	—	1 108.70	1 061.93
16	51 005	193 791	9.1e-4	—	1 723.33	1 621.27
17	73 205	284 151	6.3e-4	—	2 482.93	2 299.47
18	99 405	387 511	4.7e-4	—	3 393.50	3 127.67

TAB. 1 – Instances et résultats pour le placement optimal de caméras

Conclusion

Les résultats obtenus rendent DEsim prometteur dès lors que la taille de l’instance devient importante. Pour ces instances CPLEX ne semble pas en mesure de fournir une solution réalisable, tandis que RWLS perd son avantage à l’instance 12. Ces résultats suggèrent que l’approche hybride présentée ici devient la plus efficace sur de grandes instances, comparée à l’état de l’art du problème de couverture à coûts unitaires.

Références

- [1] M. Brévilliers, J. Lepagnot, L. Idoumghar, M. Rebai, and J. Ritter. Hybrid differential evolution algorithms for the optimal camera placement problem. *JSIT*, nov 2018.
- [2] C. Gao, X. Yao, T. Weise, and J. Li. An efficient local search heuristic with row weighting for the unicost set covering problem. *EJOR*, 246(3) :750–761, nov 2015.
- [3] J. Ritter, M. Brévilliers, J. Lepagnot, and L. Idoumghar. On the optimal placement of cameras for surveillance and the underlying set cover problem. *Applied Soft Computing*, 74 :133–153, jan 2019.
- [4] A. L. Maravilha, J. A. Ramirez, and F. Campelo. A new algorithm based on differential evolution for combinatorial optimization. In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*. IEEE, sep 2013.
- [5] K. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution : A Practical Approach to Global Optimization*. Springer-Verlag, 2005.