Redundant Visual Coverage of Prioritized Targets in IoT Applications

Removed for double-blind review
Removed for double-blind review
Removed for double-blind review
Removed for double-blind review
Removedfordouble-blindreview

Removed for double-blind review
Removed for double-blind review
Removed for double-blind review
Removed for double-blind review
Removedfordouble-blindreview

Removed for double-blind review
Removed for double-blind review
Removed for double-blind review
Removed for double-blind review
Removedfordouble-blindreview

ABSTRACT

Visual monitoring is an important service for a lot of emerging and traditional applications in the Internet of Things (IoT) field, since still images and video streams can be gathered and processed for different kinds of multimedia-based tasks. Frequently, a set of targets may need to be covered by interconnected cameras, providing concurrent multiple views under different perspectives for the applications, besides enhanced resistance to camera failures. In this context, if targets have different monitoring priorities, the configuration of the cameras can be optimized, demanding proper algorithms to maximize redundant coverage over more relevant targets. This paper proposes a lightweight greedy algorithm and a costly but more efficient evolutionary algorithm to optimize redundant visual coverage by cameras, both aimed at reduction of uncovered targets and maximization of redundant coverage over the most relevant targets, which may improve the overall quality of different visual IoT applications.

CCS CONCEPTS

• **Computer systems organization → Sensor networks**; *Real-time systems*; • **Networks →** *Network reliability*; • **Computing methodologies → Camera calibration**;

KEYWORDS

Visual IoT applications, camera calibration, prioritized target viewing, greedy algorithms, evolutionary algorithms

ACM Reference format:

Removed for double-blind review, Removed for double-blind review, and Removed for double-blind review. 2018. Redundant Visual Coverage of Prioritized Targets in IoT Applications. In *Proceedings of Brazilian Symposium on Multimedia and the Web, Salvador, BA Brazil, October 2018 (WebMedia'2018),* 8 pages.
https://doi.org/XX.XXX/XXX_X

1 INTRODUCTION

The revolution started by the Internet of Things concept has put some important services at the spot, providing large amounts of data for uncountable applications [2, 21]. When IoT devices are able to gather visual data through cameras, monitoring applications may be significantly improved by the processing of image and video data. For emerging smart city platforms, vehicular networks, industry 4.0 systems and big data applications, visual data provided by IoT devices are of paramount relevance, but it brings some challenges when the employed camera units have to be adjusted for efficient viewing of the monitored field.

IoT devices equipped with cameras may be employed to monitor (cover) a set of static or moving targets. Concerning the coverage of those targets, such IoT devices may be modelled as cameras that may be carefully positioned in order to enhance coverage according to some metric, such as redundancy [12], multiple sensing perspectives [14], energy saving [25] and other aspects related to coverage efficiency in Wireless Visual Sensor Networks (WVSN) [20] and generic IoT systems. Although there are different metrics to assess and optimize visual coverage, critical applications will usually be guided by the need for increased redundancy, which means that a target should be covered by more than one camera in order to resist coverage and hardware failures [13]. For the broad scope of Internet of Things applications, industrial control, traffic management, rescue operations and medical assistance are some of the scenarios that may be assumed as critical and that will demand some level of redundant coverage.

When cameras are able to change their orientations, either due to some specific hardware that makes them rotatable or due to manual repositioning of the cameras, the viewing configuration of all considered cameras can be adjusted. In this context, the Redundant Coverage Maximization problem (RCM) is defined when there is a need for (active) multiple views over a set of targets, since the cameras should be adjusted to maximize redundant coverage over them. This problem, which has important practical applications, was initially addressed in [REMOVED FOR BLIND REVIEW] through a greedy algorithm, presenting some promising results.

Usually, if a target is being viewed by at least one visual sensor, the target is assumed as covered, while a target is assumed as R *Redundant* when it is being viewed by N cameras, for $N > 1$ and $R = N - 1$. Actually, a target may be any element (person, animal or object) that can be viewed by a camera. When all targets are equal for the applications, the objectives of the RCM is to assure that all targets is being viewed and that the average values of R are maximized. However, sometimes, targets may be different in the sense that they have different impacts on the quality of the visual IoT applications. For example, people at downtown area may be more relevant for a public security application than people at the

WebMedia'2018, October 2018, Salvador, BA Brazil

ACM ISBN XXX-XXXX-XX-XXX/XX/XX…$15.00
https://doi.org/XX.XXX/XXX_X

suburb, while cars may be more relevant in a parking system than people. For such cases, it is expected that optimization algorithms maximize coverage over the most relevant targets.

This paper defines the Redundant Coverage Maximization of Prioritized Targets problem (RCMPT), which aims to assure that all targets are covered and that the values of R of the most relevant targets are maximized, potentially enhancing the performance of visual IoT applications. For that, a greedy algorithm (extending the work in [REMOVED FOR BLIND REVIEW]) and an evolutive algorithm based on *a priori* decisions are proposed, and the results of both algorithms are presented and compared. These algorithms can then be used as a reference for further developments in this area.

The remainder of this article is organized as follows. Section 2 presents some related works. Fundamental concepts are presented in Section 3, followed by the proposed algorithms in Section 4. Numerical results are shown in Section 5 and then, conclusions and references are presented.

2 RELATED WORKS

The Internet of Things revolution is already on the go and the last years have seen increasing developments in this area [23]. Among the possible applications, the ones based on the acquisition of visual data are very promising, but there are still many relevant challenges that have to be addressed [1, 7]. Actually, for the particular problem of redundant coverage maximization, some works have influenced this paper in different ways, as presented in this subsection.

Generally speaking, smart cameras can be positioned on a monitored scenario in different configurations, but usually the positioning approaches may allocate cameras in fixed positions, following moving patterns or even employing random positioning approaches [9]. Whatever is the chosen positioning strategy, cameras may be adjusted in order to enhance the coverage quality, either because the initial configuration is not efficient or because the target set is changed. In such ways, coverage optimization of connected cameras has been one relevant research topic for wireless visual sensor networks, smart cities and Internet of Things applications in past years, with different particularities.

Coverage optimization research has been influenced by the proposal of coverage metrics and assessment approaches. In short, coverage metrics are desired when assessing the visual sensing quality and how well the considered cameras are satisfactorily achieving the applications goals [4, 10, 26]. And coverage quality assessment has been focused on different parameters, assessing coverage based on viewed areas, viewed targets, barriers coverage, perspectives over targets and even perimeter viewing [6, 8, 14, 18]. In fact, all these particularities influence the way coverage maximization will be performed.

Critical visual applications will be concerned with the availability level of employed cameras [11]. Actually, general camera networks may be impacted by coverage and hardware failures, since both of those failure conditions may prejudice the quality of the performed visual monitoring tasks, decreasing the achieved availability level [13]. In such cases, a feasible mechanism to improve availability of any kind of camera network is increasing visual redundancy, since failed cameras can be replaced. In this context, some works have been concerned with redundant coverage optimization, achieving the minimum set of cameras that cover all considered targets [11, 15]. For example, the work in [3] proposed a mechanism that computes the minimum number of nodes that can view all targets in the monitored field, turning off redundant sensors for energy efficiency, while the work in [27] proposed a heuristic to compute the best configuration for visual sensor nodes to cover the greatest number of targets. Generally speaking, this objective can be defined as the Redundant Coverage Optimization (RCO) problem, which is out of the scope of this paper.

For visual IoT applications, the RCO problem is relevant for many scenarios, but sometimes applications may be not concerned with the minimum number of cameras to be activated and online. On the contrary, some applications may need to increase active visual redundancy over the targets, which is achieved maximizing the number of targets views by cameras when computing their orientations. That particular Redundant Coverage Maximization (RCM) problem was initially addressed by [REMOVED FOR BLIND REVIEW] through a greedy algorithm, presenting important results.

Considering that targets may have different priorities for the applications, this paper defines the Redundant Coverage Maximization of Prioritized Targets (RCMPT) problem, which is related to the maximization of targets views by cameras, but considering the targets' priorities as a decision parameter. At a certain point, this RCMPT problem can be considered as a multi-objective problem that takes different coverage metrics that have to be optimized, and thus it is suitable for evolutionary algorithms, which fostered the development of this work. Moreover, the greedy algorithm proposed in [REMOVED FOR BLIND REVIEW] can also be adapted to consider targets with different priorities. Together, both algorithms are expected to provide different results in terms of computing performance and visual coverage maximization, as discussed in next sections. Although greedy-based algorithms are easy to implement, the achieved results may be not satisfactory, and thus it is expected that evolutionary algorithms can perform much better in this scenario.

3 FUNDAMENTAL CONCEPTS

3.1 Cameras

A camera network may be part of a IoT system, a wireless sensor network or even it may be directly connected to the Internet using the TCP/IP protocol stack. Whatever the case, we focus the definition of a camera in the context of the Internet of Things, since it is expected to be the dominant platform for the upcoming years.

A visual IoT application will be composed of C homogeneous or heterogeneous cameras, which may be randomly or deterministically positioned over an area of interest. Each camera c, $c = 1, ..., C$, has a $(Ax_{(c)}, Ay_{(c)})$ location for a 2D modelling approach, assuming that the cameras are fixed and that their configurations cannot change after deployment, excepting their viewing orientations. Actually, even for randomly deployed cameras, when we do not know their initial positions, some localization mechanism can be used [22] and thus it is not a particular concern for redundant coverage maximization algorithms.

As rotatable cameras are considered, it is defined that the orientation $\alpha_{(c)}$ can change, ranging from 0° to 360°, but the viewing angle $\theta_{(c)}$ and the sensing range r are fixed and do not change. For simplification, the Field of View (FoV) of any camera is defined as the area of an isosceles triangle composed of three vertices, A, B and C. Vertex A is assumed as the position of the camera, ($Ax_{(c)}$, $Ay_{(c)}$), while the other vertices are computed considering the values of $\theta_{(c)}$, $\alpha_{(c)}$ and r.

Figure 1 shows a graphical representation of a typical camera's Field of View.

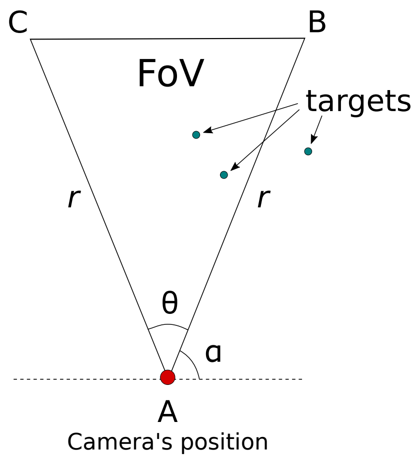

Figure 1: Field of View of a generic camera.

Using trigonometry, the vertices B and C (1), as well as the area of the FoV (2), can be easily computed [15].

$$\begin{aligned} Bx_{(c)} &= Ax_{(c)} + r_{(c)}.cos(\alpha_{(c)}) \\ By_{(c)} &= Ay_{(c)} + r_{(c)}.sin(\alpha_{(c)}) \\ Cx_{(c)} &= Ax_{(c)} + r_{(c)}.cos((\alpha_{(c)} + \theta_{(c)})mod2\pi) \\ Cy_{(c)} &= Ay_{(c)} + r_{(c)}.sin((\alpha_{(c)} + \theta_{(c)})mod2\pi) \end{aligned} \tag{1}$$

$$Area_{FoV_{(c)}} = \frac{r_{(c)}^2.sin(\theta_{(c)})}{2} \tag{2}$$

3.2 Targets viewing

Considering the particular problem of target viewing by a set of cameras, and that the objective is to maximize cameras views over targets, all targets are modeled equally for simplification purposes. In general, a target is any moving or static object that may have different formats and sizes. Actually, cameras may view just small parts of targets [10], but for the problem of redundant coverage maximization, they are modelled as points. Moreover, the problem of targets occluded by other targets are not considered in this work, for simplification purposes.

For a total of T targets, a target t, $t = 1, ..., T$, is defined as having position ($Tx_{(t)}$, $Ty_{(t)}$) as its center. Additionally, every target t has the priority $P_{(t)} = p$, with p as a positive integer number ranging from $Pmin$ to $Pmax$, $Pmin \geq p \geq Pmax$ and $Pmin < Pmax$, as defined by the considered IoT application.

The priority of targets may be established in different ways, e.g. applying algorithms for detection of visual patterns or even considering GPS coordinates of the targets as reference. Nevertheless, the way priorities will be assigned to the targets is out of the scope of this work, and it is expected that the proposed algorithms already know the targets' priority, which is a very realistic assumption.

There are different ways to verify if a target is being viewed by a camera, which is required for the RCMPT problem. In order to provide an easy mechanism to perform such verification in this work, initially the euclidean distance between the considered camera c and the target t has to be less than r_c. After that, a target t is viewed by a camera c if and only if the point ($Tx_{(t)}$, $Ty_{(t)}$) is completely inside the FoV of the camera c. Figure 2 presents a feasible way to perform such computation.

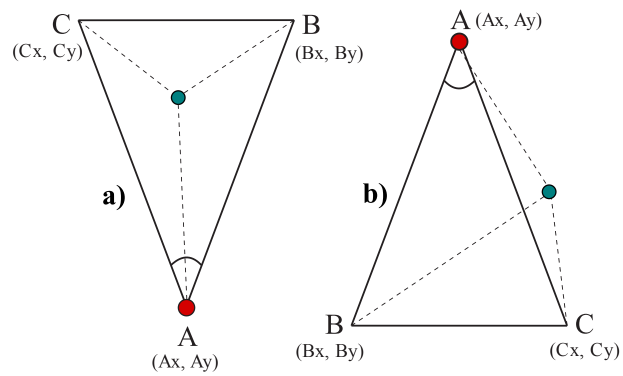

Figure 2: Checking if a target is inside a FoV. Target is inside in a), and outside in b).

The computation if a target is being viewed is then performed by the definition of three triangles, always taking two vertices of the cameras' FoV and the target position. The formulation in (3) computes the three triangles created when checking if a target is being properly viewed.

$$\begin{aligned} \triangle AtB = (&Ax_{(c)}.(By_{(c)} - Ty_{(t)}) + Bx_{(c)}.(Ty_{(t)} - Ay_{(c)}) + \\ &Tx_{(t)}.(Ay_{(c)} - By_{(c)})) \\ \triangle AtC = (&Ax_{(c)}.(Ty_{(t)} - Cy_{(c)}) + Tx_{(t)}.(Cy_{(c)} - Ay_{(c)}) + \\ &Cx_{(c)}.(Ay_{(c)} - Ty_{(t)})) \\ \triangle BtC = (&Tx_{(t)}.(By_{(c)} - Cy_{(c)}) + Bx_{(c)}.(Cy_{(c)} - Ty_{(t)}) + \\ &Cx_{(c)}.(Ty_{(t)} - By_{(c)})) \end{aligned} \tag{3}$$

Then the equality in (4) must be true if the target is being viewed by the considered visual sensor.

$$(\triangle AtB + \triangle AtC + \triangle BtC) = FoV_{(s)} \tag{4}$$

4 PROPOSED ALGORITHMS

For a lot of visual IoT applications, it may be desired that a set of targets have to be viewed by some or all considered cameras, achieving both visual redundancy (for increased resistance to failures) and active multiple viewing perspectives over targets. Although there is a wide scope of potential applications for that scenario, a special group of applications will be interested in the relevance of the targets and, thus, redundancy have to be maximized considering this optimization parameter.

Therefore, for the RCMPT problem, two different algorithms were proposed: a light-weighted greedy algorithm and a more complex but potentially more effective evolutionary algorithm. Both algorithms are designed to maximize coverage according to the priority of the targets, with the difference that the evolutionary algorithm also strives to avoid that targets that can be seen by at least one camera get uncovered even having very low priority, since it has a global perception of the camera network.

For both greedy and evolutionary algorithms, a centralized computing strategy was defined. In other words, they will be executed in some central unit that have knowledge of the configuration and positions of all visual sensors and all targets, and thus the final orientations can be transmitted further to the corresponding cameras. This is not necessarily a restriction, but only a implementation decision, but future works could consider distributed strategies.

The proposed algorithms are presented in next subsections.

4.1 Greedy algorithm

Generally speaking, a greedy algorithm performs taking decisions locally, which means that each camera try to do its best to maximize coverage. For the RCM problem, as presented in [REMOVED FOR BLIND REVIEW], the greedy algorithm maximizes the number of covered targets. However, as this work addresses the problem of coverage maximization over prioritized targets, cameras will try to maximize the sum of the priorities of viewed targets, and thus this is the optimization objective.

Considering a camera c in C, the algorithm must find the orientation $\alpha_{(c)}$ that maximizes the sum of the priorities of the viewed targets. For simplification and following a similar approach of the one adopted in [REMOVED FOR BLIND REVIEW], it is expected that rotatable cameras can assume one of a finite set of disjoint orientations, making this problem tractable, as depicted in Figure 3.

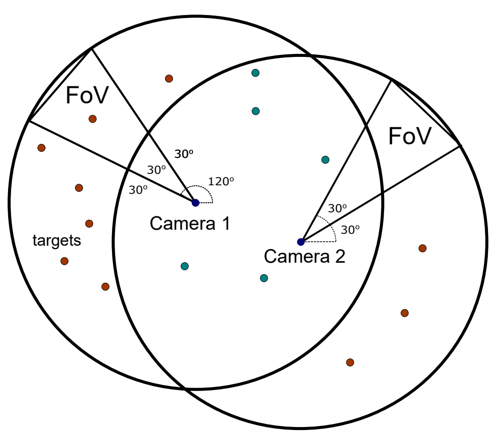

Figure 3: Disjoint positions of the orientations of the cameras, for $\alpha = 30^o$. There are 360/30 different positions for each of the cameras.

Assuming that only disjoint positions can be assumed, the number of possible positions for $\alpha_{(c)}$ is defined as $\lceil 2\pi/\alpha_{(c)} \rceil$. The optimization function of the proposed greedy algorithm for the RCMPT problem is then described in Equation (5). The function $FoV(\alpha_{(c)})$ returns a subset of targets $V_{(c)}$ of the set T that can be viewed by camera c.

$$\begin{aligned} \underset{\alpha_{(c)}}{\text{maximize}} \quad & FoV(\alpha_{(c)}) = (FoV(0), FoV(\alpha_{(c)}), ..., FoV(2\pi)) \\ \text{subject to} \quad & \sum_{t=0}^{V_{(c)}} P_{(t)} \end{aligned} \tag{5}$$

Actually, he greedy algorithm can be easily implemented, justing testing all possible disjoint positions and taking the value of α that maximizes $\sum_{t=0}^{V_{(c)}} P_{(t)}$, for $V_{(c)}$ as the set of viewed targets by camera c. However, as only local decisions are taken, always considering the best configuration for each camera individually, the coverage and redundancy results for the RCMPT problem could be further improved, leading to the development of an evolutionary algorithm, as described in next subsection.

4.2 Evolutionary algorithm

For many applications, the redundant coverage maximization of prioritized targets can be seen as a multi-objective problem. Besides the maximization of the sum of the priorities of the viewed targets, performed as the maximization of a *priority weighted sum* as defined in (5), it may be also required that all targets are viewed by at least one camera, defined as a *Coverage* objective, unless the uncovered targets could not be viewed by any camera (the target can not be placed inside the field of view of at least one camera or the computed configuration for the orientations can not attend all targets), assuring a minimum level of coverage with at least one camera. Moreover, it may be also desired that all targets have a minimum level of visual redundancy, meaning that at least two cameras can view each of the targets, defined as a *Redundancy* objective. Actually, all these three objectives can be optimized, even if they are conflicting, leading us to the proposition of a new evolutionary algorithm.

Generally speaking, evolutionary algorithms are a class of algorithms defining meta-heuristics to be applied according to the characteristics of the problem being modelled. They apply a population-based search approach to iteratively select and vary a set of candidate solutions through evaluation procedures, and thus they are well suited for multi-objective optimizations [17].

Some previous works have employed evolutionary algorithms to address the problem of coverage optimization in complex systems like wireless sensor networks and Internet of Things scenarios. Authors in [19] have proposed an evolutionary algorithm to enhance the total viewed area by visual sensors, minimizing sensing overlapping. The work in [16] employed an evolutionary algorithm to optimize the selection of IoT devices for more efficient communications. Camera calibration performed by evolutionary algorithms has also been investigated in some works [5, 24], with practical application in IoT systems. Actually, evolutionary algorithms have been used to solve coverage problems, but to the best of our knowledge, the redundant coverage maximization multi-objective problem has not been treated before by evolutionary algorithms, and the same is true when targets have different priorities.

Due to the way evolutionary algorithms operate, which is not deterministic, it is possible to achieve different results in independent executions and there is no guarantees that optimal results will be achieved, but "good" results may be frequently computed. For that, the proposed evolutionary algorithm will operate based on *a priori* approach, since there will be a definition of a preference information of the objectives before starting the optimization process. And this will be defined following a given priority order. In fact, such *a priori* evolutionary algorithm follows a lexicographic method, which was selected for the RCMPT problem due to its inherent characteristics. Actually, although evolutionary algorithms can be defined following different approaches, the *a priori* method was selected since it is simpler to implement, but with good results for the RCMPT problem.

Therefore, the proposed *a priori* evolutionary algorithm is a lexicographic algorithm that defines an order of preference for the objectives to be optimized and computes a single final solution to be considered. And as it is not deterministic, different executions of the algorithm may produce different solutions. The proposed algorithm processes a population (P) of individuals (possible solutions) and each individual is represented, considering the RCMPT problem, as orientations for each camera. Each possible solution is a numerical vector with size S with each position $\alpha_{(c)}$ being a value (from $0°$ to $360°$) indicating the orientation to be assumed by the corresponding c camera.

The proposed algorithm follows the basic steps expected from evolutionary algorithms, which are the definition of a initial initial population (P vectors with fixed size - number of cameras) and the definition of a "evaluation", a "selection", a "crossover" and a "mutation" function, allowing that a feasible solution "evolves" through different iterations. Actually, the evaluation function (also referred as "fitness function") is a key element of the algorithm, since it evaluates each candidate solution by computing the values for the three different optimization objectives: *Coverage* (*Cov*), *Priority Weighted Sum* (PWS) and *Redundancy* (*Red*). As a lexicographic approach, the objective of maximizing *Cov* has higher preference over maximizing PWS, which has higher preference over maximizing Red.

Still considering the definitions of the proposed evolutionary algorithm, the selection ("fitness") function evaluates the effectiveness of the individuals, which are then processed in a selection method composed of two phases. Initially, the selection process gets the 2 (or 3, for an even number of individuals in the population) best individuals and directly copy them to the new population in the next generation. Such selection (*elitist selection*) is performed based on the greatest values for Cov, PWS and Red, considering this preference order. After that, a "tournament" process takes place, with K individuals being drawn randomly from current population and the best individual being selected for a temporary subpopulation. The tournament is repeated until this subpopulation reaches P individuals minus the number of individuals initially selected in the elitist selection. After that, this subpopulation set, which may even contain repeated individuals, will be considered in the next step of the algorithm.

Finally, "crossover" and then "mutation" take place as processing steps of the algorithm. These steps are necessary for the evolutive process, once each two selected parent individuals generates two new individuals (children), exchanging orientation values. After "crossover", a mutation method is used to produce genetic variation of the population. After this process, some orientations are changed in the individuals in the subpopulation set, generating variation for the evolutionary process. After the execution of the *mutation* step, the subpopulation is incorporated to the new population set for the next generation, which already had 2 (or 3) individuals from the elitist selection. This overall process is then repeated until a "stopping criterion" of maximum number of generations is reached.

As the proposed algorithm is parameterized, there are some configurations that have to be defined: population size, number of individuals (candidate solutions) to be processed in each iteration, number of generations (this is the stopping criterion of the algorithm), tournament size (the number of individuals selected to compete), crossover rate (the probability of two selected parents to generate children), mutation rate (the probability of an individual to be processed in the mutation step) and the number of positions for mutation (the percentage of positions in the individuals' vector that will be considered for mutation). All these parameters can be changed in order to "guide" the processing costs and quality of the results.

Differently of the greedy algorithm, the evolutive algorithm was not defined to work with disjoint positions (Figure 3), potentially achieving better results at the cost of a higher computational expense. Actually, the idea was to oppose two algorithms with very distinct costs, for comparison purposes, but a lightweight version of the evolutive algorithm could be designed with the adoption of disjoint orientations.

The execution steps of the proposed *a priori* evolutionary algorithm is presented in Figure 4.

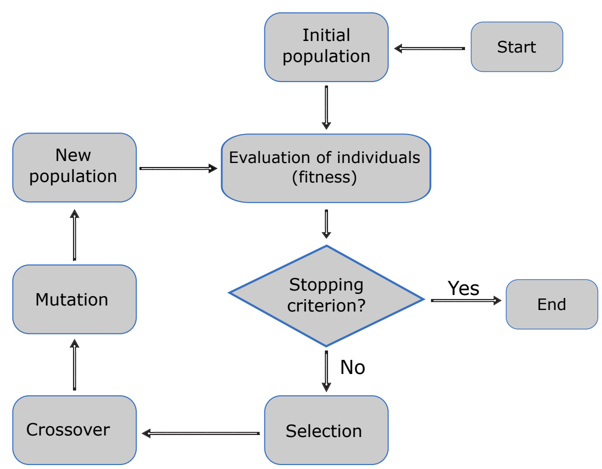

Figure 4: General operation of the proposed evolutionary algorithm.

Although there is a relatively high computational expense with evolutionary algorithms, specially due to the intense iterations over the search set, the computation of a possible (near) global optimum may be worth the cost, notably in large search space problems like in IoT optimizations.

5 NUMERICAL RESULTS

The proposed algorithms were executed and the results were compared, for different configuration parameters. The objective of the verifications was to compare the proposed algorithms between them, since they are the only available solutions for the RCMPT problem, and thus we could not make comparisons with algorithms on the literature. This section presents some of main results obtained after a series of execution tests.

A typical visual IoT application was defined on an area with 500m x 500m and assuming a uniform probability distribution of positions for the cameras and targets. It was also considered that all cameras are equal and that they have $\theta = 60°$ and $R = 70m$. Moreover, it is defined $Pmin = 1$ and $Pmax = 10$ for the targets. The results are then presented for average values after 20 consecutive executions with the same random positions of cameras and targets, better presenting an average behavior for the algorithms.

In order to visually demonstrate the execution of the algorithms, some *screenshots* were taken during the tests, since results were also plotted on graphical windows. As an example, Fig 5 presents some random initial configuration (cameras' orientations) before the execution of redundant coverage maximization, for 40 cameras and 50 targets with random priorities.

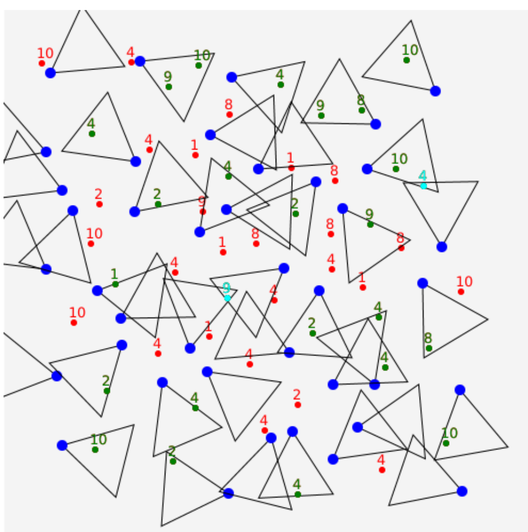

Figure 5: A random initial configuration for the cameras.

Figure 6 e Figure 7 present the same scenario after redundant coverage maximization, for the greedy and the evolutionary algorithms, respectively.

As can be seen in the configurations of the cameras, both algorithms achieved better results in terms of redundant coverage maximization, when compared with the initial scenario (Figure 5). However, it can be noted that the evolutionary algorithm performed better, leaving fewer targets uncovered (which is one of the defined optimization objectives).

Numerical results were obtained considering a varying number of cameras, from 10 to 80, and for scenarios with 50 and 100 targets. The results for *Coverage* is presented in Figure 8, initially for 50 targets. The results for the initial scenario before the use of the proposed algorithms are also displayed, for comparisons purposes. And although the proposed greedy algorithm was not designed to

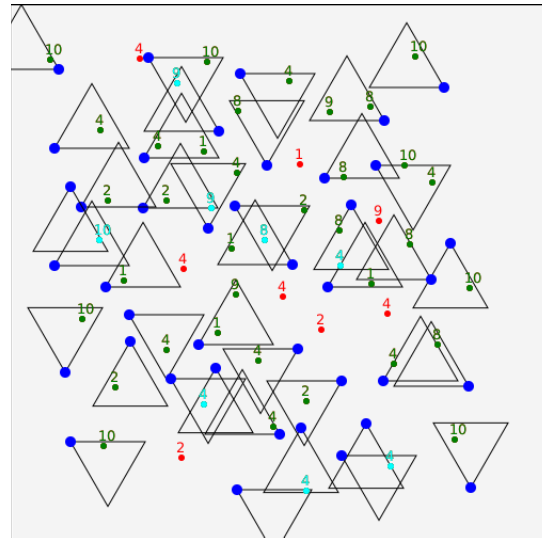

Figure 6: Camera scenario after execution of the greedy algorithm.

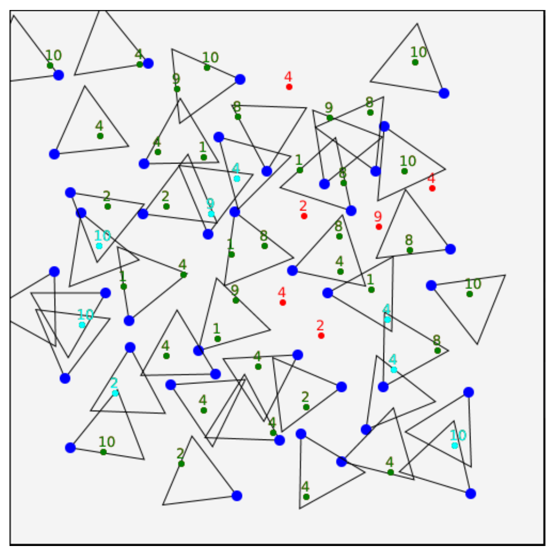

Figure 7: Camera scenario after execution of the evolutionary algorithm.

optimize *Cov*, the obtained results are also displayed in order to allow comparisons with the evolutionary algorithm.

For the results in Figure 8, *Coverage* will be higher for the evolutionary algorithm, meaning that fewer targets will get uncovered, when compared with the greedy algorithm. The vertical lines show the minimum and maximum values, since average results are presented.

Results for *Priority Weighted Sum* and *Redundancy* are presented in Figure 9 and Figure 10, respectively.

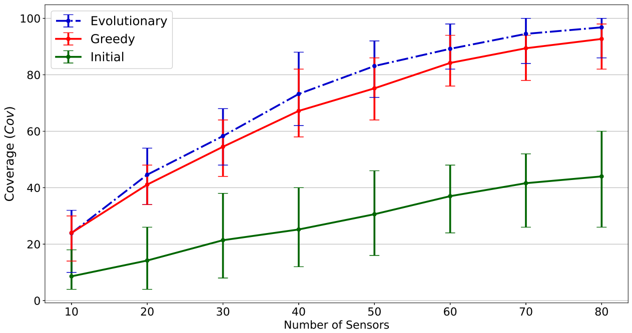

Figure 8: Values for *Cov* for the proposed algorithms and 50 targets.

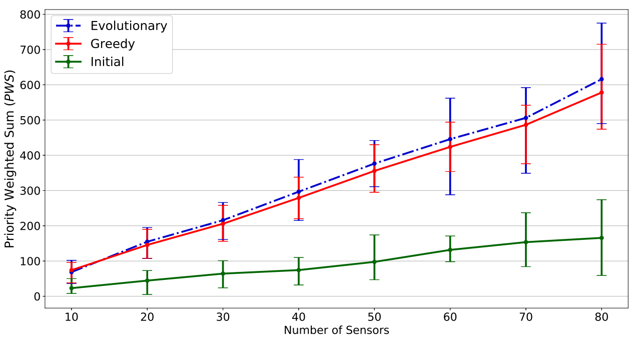

Figure 9: Values for *PWS* for the proposed algorithms and 50 targets.

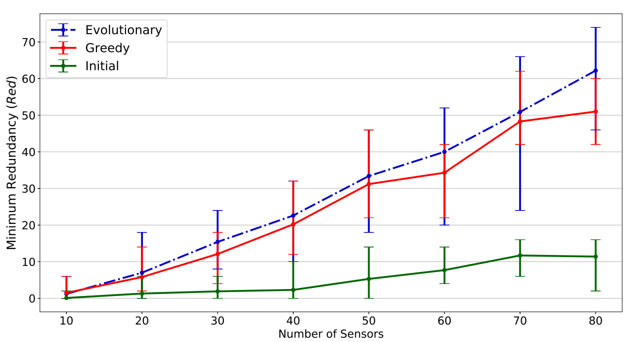

Figure 10: Values for *Red* for the proposed algorithms and 50 targets.

As can be seen in the presented results for 50 targets, the proposed evolutionary algorithm performed better in most configurations, when compared to the greedy algorithm. It is due to the fact that (among other factors) the evolutionary algorithm considers the three objectives to optimize, with a "global" perception of the cameras, thus better optimizing the IoT camera network. However, as there are more iterations, the evolutionary algorithm has a higher computational cost than the greedy algorithm. Table 1 presents the computational costs of both algorithms, considering the number of times a target coverage test is performed (verification if a target is inside the FoV of a camera).

Table 1: Average computational cost when covering 50 targets.

Cameras	Evolutionary	Greedy
10	495,207.25	4,000.00
20	1,974,129.70	8,000.00
30	4,442,972.70	12,000.00
40	7,894,369.00	16,000.00
50	12,337,390.65	20,000.00
60	17,793,075.90	24,000.00
70	24,218,967.95	28,000.00
80	31,610,100.85	32,000.00

As expected, the computational costs of the evolutionary algorithm is much higher than the costs of the greedy algorithm, thus requiring more computational power and demanding more execution time. Therefore, although the achieved results are better for the RCMPT problem, the computational costs have to be properly considered.

Results for *Coverage*, *Priority Weighted Sum* and *Redundancy* are presented in Figure 8, Figure 9 and Figure 10, respectively, for the coverage of 100 targets and considering the same configuration parameters for the camera.

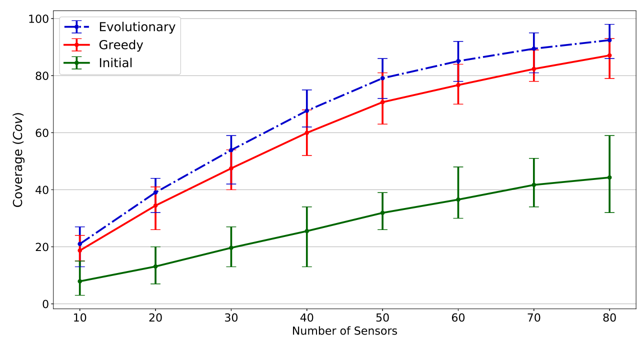

Figure 11: Values for *Cov* for the proposed algorithms and 100 targets.

After execution of the algorithms, one can note that there is considerable improvement of redundant coverage over prioritized targets, which is desired for many applications. The two proposed algorithms are reasonable solutions to address the RCMPT problem, with different performance results, opening new possibilities for future research in this field.

6 CONCLUSION

The problem of redundant coverage maximization considering the use of cameras to cover a set of targets is of paramount importance for many applications, fostering the development of innovative algorithms in this area. When targets have different priorities, this problem becomes a bit more complex, but the possibilities for

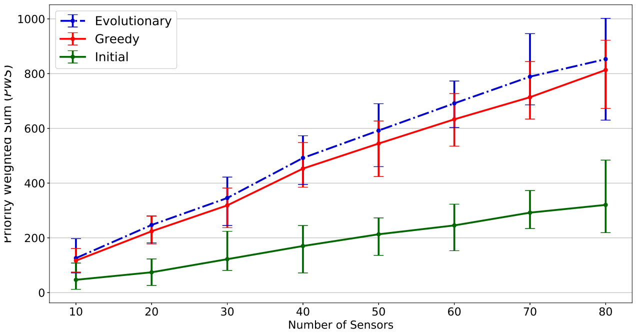

Figure 12: Values for *PWS* for the proposed algorithms and 100 targets.

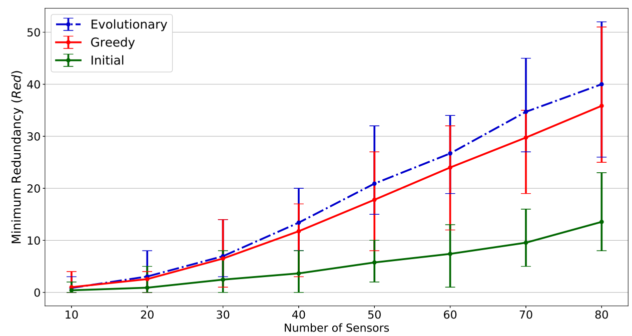

Figure 13: Values for *Red* for the proposed algorithms and 100 targets.

camera-based applications are also increased, with direct benefits in the IoT world.

The proposed lexicographic algorithm is an appropriate approach when we can favor some optimization objectives over the others and it is expected that it will adequately perform well for the RCMPT problem. The greedy algorithm is simpler to implement and execute, but the results are worse. Together, both algorithms are good references for new developments inf this area.

As future works, additional verification will be performed, considering the impact of applying the proposed algorithms in real IoT scenarios. The communication costs will be evaluated and the overall assignment time of computed orientations will be assessed. At last, new variations of the evolutionary algorithm, for example considering *a posteriori* decisions, will be developed.

REFERENCES

[1] Kevin Abas, Katia Obraczka, and Leland Miller. 2018. Solar-powered, wireless smart camera network: An IoT solution for outdoor video monitoring. *Computer Communications* 118 (2018), 217 – 233.

[2] Ejaz Ahmed, Ibrar Yaqoob, Abdullah Gani, Muhammad Imran, and Mohsen Guizani. 2016. Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges. *IEEE Wireless Communications* 23 (2016), 10–16.

[3] Jing Ai and Alhussein A Abouzeid. 2006. Coverage by directional sensors in randomly deployed wireless sensors networks. *Journal of Combinatorial Optimization* 11 (2006), 21–41.

[4] M. Alaei and J. M. Barcelo-Ordinas. 2010. Node clustering based on overlapping FoVs for wireless multimedia sensor networks. In *IEEE Wireless Communications and Network Conference*. 1–6.

[5] D. V. Arnold and A. Lu. 2016. An evolutionary algorithm for depth image based camera pose estimation in indoor environments. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. 3871–3878. https://doi.org/10.1109/CEC.2016.7744280

[6] C. Y. Chang, C. Y. Hsiao, and C. T. Chang. 2012. The k-barrier coverage mechanism in Wireless Visual Sensor Networks. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. 2318–2322.

[7] L. W. Chen, Chi-Ren Chen, and Da-En Chen. 2017. VIPS: A video-based indoor positioning system with centimeter-grade accuracy for the IoT. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 63–65. https://doi.org/10.1109/PERCOMW.2017.7917523

[8] C. F. Cheng and K. T. Tsai. 2012. Distributed Barrier Coverage in Wireless Visual Sensor Networks With $beta$-QoM. *IEEE Sensors Journal* 12, 6 (June 2012), 1726–1735.

[9] Daniel Costa and Luiz Guedes. 2010. The coverage problem in video-based wireless sensor networks: a survey. *Sensors* 10 (2010), 8215–8247.

[10] Daniel G. Costa and Cristian Duran-Faundez. 2016. Assessing Availability in Wireless Visual Sensor Networks Based on TargetsâĂŹ Perimeters Coverage. *Journal of Electrical and Computer Engineering* 2016 (2016), Article ID 9312439, 14 pages.

[11] D. G. Costa, I. Silva, L. A. Guedes, P. Portugal, and F. Vasques. 2014. Availability assessment of wireless visual sensor networks for target coverage. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. 1–8.

[12] Daniel G. Costa, Ivanovitch Silva, Luiz Affonso Guedes, Paulo Portugal, and Francisco Vasques. 2014. Enhancing Redundancy in Wireless Visual Sensor Networks for Target Coverage. In *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web (WebMedia '14)*. ACM, New York, NY, USA, 31–38.

[13] Daniel G. Costa, Ivanovitch Silva, Luiz Affonso Guedes, Francisco Vasques, and Paulo Portugal. 2014. Availability Issues in Wireless Visual Sensor Networks. *Sensors* 14 (2014), 2795–2821.

[14] D. G. Costa, I. Silva, L. A. Guedes, F. Vasques, and P. Portugal. 2015. Optimal sensing redundancy for multiple perspectives of targets in wireless visual sensor networks. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*. 185–190.

[15] Daniel G. Costa, Francisco Vasques, and Paulo Portugal. 2017. Enhancing the availability of wireless visual sensor networks: Selecting redundant nodes in networks with occlusion. *Applied Mathematical Modelling* 42 (2017), 223 – 243. https://doi.org/10.1016/j.apm.2016.10.008

[16] M. Cuka, D. Elmazi, R. Obukata, K. Ozera, T. Oda, and L. Barolli. 2017. An Integrated Intelligent System for IoT Device Selection and Placement in Opportunistic Networks Using Fuzzy Logic and Genetic Algorithm. In *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. 201–207. https://doi.org/10.1109/WAINA.2017.178

[17] Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. 2015. Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)* 48, 1 (2015), 13.

[18] Vikram P. Munishwar and Nael B. Abu-Ghazaleh. 2013. Coverage Algorithms for Visual Sensor Networks. *ACM Transactions on Sensor Networks* 9, 4, Article 45 (July 2013), 36 pages.

[19] A Habibizad Navin, B Asadi, S Hassan Pour, and M Mirnia. 2010. Solving coverage problem in wireless camera-based sensor networks by using genetic algorithm. In *Computational Intelligence and Communication Networks (CICN), 2010 International Conference on*. IEEE, 226–229.

[20] Joao Paulo Just Peixoto and Daniel G. Costa. 2017. Wireless visual sensor networks for smart city applications: A relevance-based approach for multiple sinks mobility. *Future Generation Computer Systems* 76, Supplement C (2017), 51–62.

[21] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. 2014. Sensing as a service model for smart cities supported by internet of things. *Transactions on Emerging Telecommunications Technologies* 25, 1 (2014), 81–93.

[22] Dan Pescaru and Daniel-Ioan Curiac. 2014. Anchor Node Localization for Wireless Sensor Networks Using Video and Compass Information Fusion. *Sensors* 14, 3 (2014), 4211–4224.

[23] S. H. Shah and I. Yaqoob. 2016. A survey: Internet of Things (IOT) technologies, applications and challenges. In *2016 IEEE Smart Energy Grid Engineering (SEGE)*. 381–385.

[24] D. Strubel, O. Morel, N. M. Saad, and D. Fofi. 2017. Evolutionary algorithm for positioning cameras networks mounted on UAV. In *2017 IEEE Intelligent Vehicles Symposium (IV)*. 1758–1763. https://doi.org/10.1109/IVS.2017.7995961

[25] Florence G. H. Yap and Hong-Hsu Yen. 2014. A Survey on Sensor Coverage and Visual Data Capturing/Processing/Transmission in Wireless Visual Sensor Networks. *Sensors* 14 (2014), 3506–3527.

[26] Florence G. H. Yap and Hong-Hsu Yen. 2014. A Survey on Sensor Coverage and Visual Data Capturing/Processing/Transmission in Wireless Visual Sensor Networks. *Sensors* 14, 2 (2014), 3506–3527. http://www.mdpi.com/1424-8220/14/2/3506

[27] H. H. Yen. 2013. Efficient visual sensor coverage algorithm in Wireless Visual Sensor Networks. In *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. 1516–1521.