Optimum Multi-Camera Arrangement Using a Bee Colony Algorithm

Dimitrios Chrysostomou, *IEEE* member and Antonios Gasteratos, *IEEE* Senior member

Laboratory of Robotics and Automation, Department of Production and Management Engineering,
Democritus University of Thrace, Building I, Vasilissis Sophias 12, GR-671 00 Xanthi, Greece
E–mail: dimitris.chrysostomou@ieee.org, gasteratos@ieee.org

***Abstract*—This paper presents a new method for designing multi-camera arrangements with aim to maximize coverage with the minimum number of cameras. More specifically, the presented problem has three different components, namely *(a)* to maximize the coverage subject to a given number of cameras *(b)* to optimize the camera topology given fixed locations and *(c)* to minimize the cost of the arrangement, while the least required percentage of coverage is provided. In order to solve these problems, a bee colony algorithm is utilized as an optimization technique that is able to determine the minimum number of cameras needed to cover the given space completely while taking into consideration the minimum possible cost for the proposed arrangement as well. The algorithm employs several camera placement constraints referring to geometrical, optical as well as reconstructive limitations and delivers promising preliminary results.**

I. Introduction

Visual sensor arrangements are used in many novel applications such as video surveillance [1], sensing rooms [2], assisted living or immersive conference rooms [3]. Most of these applications require the layout of camera sensors to assure a minimum level of image quality or image resolution. Moreover, one of the most significant factors that determines the performance of a multi-camera arrangement is the amount of visibility of the observed space. Depending on the application, the visibility of a predefined space could be partial or complete. Thus, an important issue in designing visual sensor arrangements is the appropriate placement of the cameras such that they achieve the maximum amount of visibility as possible.

Currently most designers of multi-camera systems face difficulties concerning the visual sensor placement due to the lack of optimization techniques. Moreover, as multi-camera arrays are getting more popular and their cost tends to get reduced every day, efficient strategies for their topology need to be developed. In this paper, the main focus is on the topic of maximizing or achieving coverage with respect to a satisfying field-of-view, guaranteeing that an object in the space will be imaged at a minimum resolution.

The problem of multi-camera topology in a 3D monitoring space is based on the renowned Art Gallery Problem *(AGP)*. The main goal in the latter is to find the minimum number of guards that can monitor a fixed number of paintings in a gallery [4]. The layout of the art gallery is a close polygon and the covering points (vertices on the polygon) are the guards. In our case the guards are replaced by cameras and the goal is to find the minimum number of them that at the same time provide an optimum space coverage. The original art gallery theorem states that at most *(n/3)* guards are required for covering polygons with *n* edges [5] and is known to be a NP-hard problem [6], [7]. Many variations of AGP have been studied in previous works that address a variety of restrictions for the guards and a plethora of distinct polygons. More specifically, González-Baños and Latombe [8] established a 2D polygonal map where a robot visits polygons containing more expensive tasks with highest priority, whilst Urrutia [9] introduces new directions of research based on watchman routes and floodlight illumination problems in his extensive survey. Efrat and Har-Peled [10] presented randomized approximation algorithms for finding the smallest set of visible points of a polygon, while Ghosh [11] presents an $O(n^4)$ time approximation algorithm for simple polygons yielding solutions within a log n factor of the optimal. Recently, Bottino and Laurentini [12] proposed an incremental algorithm for interior and edge covering which produces results with nearly optimal performance or close to the lower bound of the polygonal environment and latest results presented from Couto [13] shown that an exact solution is possible by discretizing the examined polygon in $O(n^3)$ iterations.

Moreover, as the majority of these formulations result in NP-hard problems a profusion of optimization approaches have been studied. Standard Binary Linear Programming [14], Greedy approaches [15], Greedy heuristics, Dual Sampling techniques [16] and Monte Carlo simulations [17] are only some of the methods, researchers have employed to achieve the optimum solution. In this paper a nature-inspired solution based on the intelligent behavior of the honey bees during their foraging process and their waggle dance is used as the optimization technique to determine the optimal number of cameras needed to cover sufficiently the examined space.

Two major differences from the problem discussed in the paper in hand and the AGP are: (1) the described multi-camera topology problem has the restriction of the field-of-view of cameras in the sensor model due to resolution and sensor properties, while AGP considers guards with no such limitations and (2) multiple cameras with different fields-of-views at different levels of costs are used in the multi-camera

978-1-4577-1775-8/12/$26.00 ©2012 IEEE

topology problem, whilst in AGP all guards are assumed to have identical specifications.

The remainder of this work is outlined as follows: In Section II, definitions and the statement of the problem is given, in Section III the proposed approach is discussed, explaining the constraints regarding the multi-camera topology in III-A, the objective function used for optimization in III-B and the bee colony algorithm used to converge to the optimum final solution in III-C. The experimental results of the approach are presented in Section IV, and finally the paper concludes with several points for discussion in the last section.

II. Problem Statement

In this paper the problem of optimum multi-camera arrangement for a given space and a specific vision task is discussed. The main focus is on the static, off-line camera topology problem, where the main goal is to manage the optimal positioning of a number of visual sensors for a region to be observed, given a set of task-specific constraints. In the following, the term $Volumetric\ Space$ denotes an arbitrary connected 3D volumetric physical room, which the arrangement of multiple cameras is designed to cover. $Coverage$ means that every point of a given space is sensed with a specified minimal spatial resolution.

Given this $Volumetric\ Space$ to be covered by multiple cameras, two broadened problems related to camera placement are of the highest interest and formulate it as an optimization problem:

- The MAX_{COVER} problem, where the maximum coverage of the given $Volumetric\ Space$ is sought, given the fixed number of cameras κ of a certain type and orientation ϕ.
- The MIN_{COST} problem, where the minimum total cost of the multi-camera arrangement is looked for, subject to a target coverage taking into account all other placement constraints.

In order to solve the aforementioned optimization problems a number of definitions should be established. Initially, every examined 3D $Volumetric\ Space$ is represented as an occupancy grid consisting of P control points, potentially covered by μ cameras, the respective costs for each one being K_μ. Therefore, a binary vector containing the covered points cp is defined as:

$$cp_i = \begin{cases} 1 & \text{if control point } i \text{ is covered by} \\ & \text{at least one camera } \mu \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Every element a_{ij} of a binary coefficient matrix A is then established as:

$$a_{ij} = \begin{cases} 1 & \text{if the } i_{th} \text{ grid point is covered by} \\ & \text{the } \mu_{th} \text{ camera} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

TABLE I
Camera Topology Constraints

Element	**Constraint**
E_1	Visibility
E_2	Viewing angle
E_3	Field of view
E_4	Resolution
E_5	Viewing distance
E_6	Occlusion

The following relation holds, subject to $cp' = Ax$:

$$cp_i = \begin{cases} 1 & \text{if } cp'_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Every possible solution vector λ is assumed as

$$\lambda_j = \begin{cases} 1 & \text{if the possible solution is chosen} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

while the maximum cost of every j_{th} solution is associated with K_{max}. Considering all the above elements, The MAX_{COVER} problem can be described by

$$\max \sum_i cp_i, \text{ subject to } \sum_j K_j \lambda_j \leq K_{max} \quad (5)$$

Given a required coverage vector $cp_{C,o}$ or a minimum overall coverage C_{min}, the MIN_{COST} problem can be modeled as

$$\min \sum_j k_j \lambda_j, \text{ s.t. } A\lambda \leq cp_{C,o} \text{ or } \sum_i cp_i \geq C_{min} \quad (6)$$

III. Proposed Approach

A. Topology Constraints

In this work, a common vision sensor is used for the acquisition of the 3-D surface information, whilst the method can be extended to other types of sensors as well. The parameters (positional and optical) of these off-the-shelf sensors are as follows [18]:

1) Three degrees of freedom of the sensor's position: (x, y, z);
2) Three degrees of freedom of the sensor's orientation: the pan, tilt, and swing angles: (α, β, γ) and
3) optical parameters including: the back principal point to image plane distance d; the entrance pupil diameter, a of the lens; and the focal length f of the lens.

Hence, the viewpoint of the sensor can be stated as a vector:

$$\upsilon = (x, y, z, \alpha, \beta, \gamma, d, f, a) \quad (7)$$

and all viewpoints must be planned in the nine-dimensional $Volumetric\ Space$:

$$V = \{\upsilon_i | \upsilon_i \in (x, y, z, \alpha, \beta, \gamma, d, f, a)\}. \quad (8)$$

Moreover, an acceptable viewpoint must satisfy multiple sensor placement constraints as those showing in Table I, including the geometrical (E_1, E_2), optical (E_3, E_5) and reconstructive (E_4, E_6) ones.

1) Visibility: A covered point is already denoted as cp_i and is defined as a point of the occupancy grid of the examined space, $\mathbf{n}$ is its normal, S as the vision sensor, $\mathbf{v}$ is its pose whilst $\mathbf{v_a}$ as the viewing direction from S to A. A point is considered visible if the dot product of its normal and sensor's viewing direction is negative. That is

$$E_1 : \mathbf{n} \cdot \mathbf{v}_\alpha = \|\mathbf{n}\| \times \|\mathbf{v}_\alpha\| cos(180 - \theta) < 0 \tag{9}$$

This means that the point is visible if the angle (θ) between its normal and the view direction is less than $90°$.

2) Viewing Angle: When this angle is close to $90°$ the resulting image is not reliable and thus a limit should be set that will be defined as:

$$E_2 : \theta = \pi - cos^{-1} \frac{\mathbf{n} \cdot \mathbf{v}_\alpha}{\|\mathbf{n}\| \times \|\mathbf{v}_\alpha\|} < \theta_{max} \tag{10}$$

3) Field of View: All detectable and covered points are useful only when are projected within the camera's field of view. The locus, which satisfies the field of view constraint is given by the following equation:

$$E_3 : \mathbf{v} \cdot \mathbf{v}_\alpha - \|\mathbf{v}\| \cdot \|\mathbf{v}_\alpha\| cos(\frac{\alpha}{2}) \geq 0 \tag{11}$$

where α stands for the field-of-view angle of the camera.

4) Resolution: The image resolution accounts for the size of the pixels on the camera's image plane, measured in *pixels per inch*. The resolution constraint ensures that the examined space is sampled with the minimum acceptable pixel size resolvable by the vision system, expressed as:

$$E_4 : \sigma_{resol} = \left(\frac{z}{Nf} - \frac{1}{N} \right) \frac{1}{cos\theta} < \sigma_{acceptable}, \text{(pixels/inch)} \tag{12}$$

where z is the distance from the lens to an object's surface, N is the number of total pixels in an sensors scanning line and θ the angle between the object's surface normal and the optical axis.

5) Viewing distance: A digital image is considered perfectly focused at a specific distance, measured along the optical axis given by $D = fd/(d - f)$. As this distance is limited due to optical constraints from the lenses, the image is projected back to the sensor plane through a blur circle of a given size c. In the meantime, the focus is maintained sufficiently for a range of depths from D_1, the far limit of the depth of field, to D_2, the near limit.

$$D_{12} = \frac{afd}{a(d - f) \pm cf} \tag{13}$$

where a is the entrance pupil diameter, f is the focal length of the lens and d is the focus distance. If d is adjustable from d_{min} to d_{max}, considering $f < d_{min} < d_{max} < 2f$, the object can be placed between z_{min} and z_{max}

$$E_5 : z_{min} < z < z_{max} \tag{14}$$

where

$$z_{min} = \frac{afd_{max}}{a(d_{max} - f) + cf}$$
$$z_{max} = \frac{afd_{min}}{a(d_{min} - f) + cf}$$

6) Occlusion: Occlusion is one of the most important elements to consider when a camera placement algorithm is designed. A target A is completely visible if nothing blocks its view into the camera frustum. Any geometrical element e_j such as a line, surface or solid object can be considered as an occlusion. Thus, the respective constraint will form as:

$$E_6 : O = \begin{cases} \text{visible :} & if \left(\left(L_{SA} \cap \left(\bigcup_{j=1}^{n} e_j \right) \right) = \phi \right) \\ \text{occluded :} & \text{otherwise} \end{cases} \tag{15}$$

where L_{SA} is the straight line connecting the sensor center and point A, and ϕ is an empty set of intersections between covered points.

B. Objective function

The camera placement problem can be expressed as a function optimization one taken into account the weighted sum of the proposed constraints, each of which characterizes the quality of the solution with respect to each associated requirement separately. Therefore, the optimization function is written as:

$$f = w_1(\mathbf{n} \cdot \mathbf{v}_\alpha) + w_2(\theta) + w_3 E_3 + w_4(\sigma_{resol}) + w_5(z) \tag{16}$$

where $w_i, i \in \{1 \ldots 5\}$ are predefined importance weights, applied to the application in hand and hence the MAX_{COVER} and MIN_{COST} problems can be obtained by

$$\max \sum_i f_i, \text{ subject to } \sum_j K_j \lambda_j \leq K_{max} \tag{17}$$

$$\min \sum_j k_j \lambda_j, \text{ then } \max \sum_i f_i,$$
$$\text{s.t. } A\lambda \leq cp_{C,o} \text{ or } \sum_i cp_i \geq C_{min} \tag{18}$$

C. Bee Colony Algorithm

1) Bees in nature: Many studies in recent years stated that a colony of bees is able to travel over very long distances, sometimes even more than 10km. Another remarkable aspect of these colonies is their ability to choose multiple directions simultaneously in order to exploit a large number of possible food sources [19], [20], [21]. Fields containing quality food sources are essential for the survival of the colony and it is a common tactic that flower patches containing ample amounts of nectar that can be gathered will less effort are more favorable from the colony's foragers, whereas patches with less nectar will receive fewer visits. [22], [23].

The foraging process starts when the colony commend its scout bees to approach a promising field congested with flower patches. At first, scout bees pick random routes from one flower to another without following any specific pattern. After the selection of the most profitable flower patches the harvesting seasons begins, while a percentage of the colony's population remains as scout bees to continue its exploration [20]. All visited flower patches are considered a reliable food

Algorithm 1 Bee Colony Algorithm

Input: $Problem_{size}, Bees_{num}, Sites_{num}, EliteSites_{num}, PatchSize_{init}, EliteBees_{num}, OtherBees_{num}$
Output: Bee_{best}

$Population \leftarrow InitializePopulation(Bees_{sum}, Problem_{size})$
while $\neg(StopCondition())$ **do**
  $EvaluatePopulation(Population)$
  $Bees_{best} \leftarrow GetBestSolution(Population)$
  $NextGeneration \leftarrow 0$
  $Patch_{size} \leftarrow (PatchSize_{init} \times PatchDecrease_{factor})$
  $Sites_{best} \leftarrow SelectBestSites(Population, Sites_{num})$
  for $(Site_i \in Sites_{best})$ **do**
    $RecruitedBees_{num} \leftarrow 0$
    if $(i < EliteSites_{num})$ **then**
      $RecruitedBees_{num} \leftarrow EliteBees_{num}$
    else
      $RecruitedBees_{num} \leftarrow OtherBees_{num}$
    end if
    $Neighborhood \leftarrow 0$
    for (j **To** $RecruitedBees_{num}$) **do**
      $Neighborhood \leftarrow CreateNeighborhoodBee(Site_i, Patch_{size})$
    end for
    $NextGeneration \leftarrow GetBestSolution(Neighborhood)$
  end for
  $(RemainingBees_{num} \leftarrow (Bees_{num} - Sites_{num})$
  for (j **To** $RemainingBees_{num}$) **do**
    $NextGeneration \leftarrow CreateRandomBee()$
  end for
  $Population \leftarrow NextGeneration$
end while
Return (Bee_{best})

source only when their rate is above a certain quality threshold, such as the sugar content. Scout bees that return to the hive with such a successful finding will start to perform a dance known as the *waggle dance* [19].

This peculiar behavior is vital for the colony communication because it contains important information regarding every flower patch such as: the direction in which the remaining scout bees should travel in order to locate it, its distance from the hive and its quality rating according to the quality standards given by the colony. [19], [23]. The colony will then use the summary of these three pieces of information in order to send its bees precisely to the flower patches, without sacrificing energy following guides or maps. The mysterious waggle dance is the solely piece of information regarding each individual's knowledge of the outside environment. Moreover, this kind of dance allows the colony to evaluate the relative quality of different patches according to both the worth of the food they provide and the amount of energy needed to harvest it [23].

After waggle dancing on the *dance floor*, the dancer goes back to the flower patch followed by its fellow working bees. More follower bees are sent to more promising patches allowing the colony to gather food more quickly and efficiently. During the harvesting process from a patch, the working bees monitor its food level continuously, in order to decide when its the appropriate time to perform the next waggle dance when they return to the hive. [23]. If the patch is still resourceful as a food origin, then it will be announced in the waggle dance and more bees will be recruited to that source.

2) Proposed Bees Algorithm: As mentioned in [22], the Bee Colony algorithm is an optimization technique inspired by the natural behavior of honey bees, particularly designed to find optimal solution to continuous optimization problems. Thus, it is an ideal framework for the camera placement problem and is presented in pseudocode format above. In order for the algorithm to start, a number of parameters need to be set, such as: the $Problem_{size}$ derived from the number of available κ cameras and the $Bees_{num}$ standing for the initial grid points for κ cameras. An initial population is then created by placing randomly the scout bees into the 3D space. An initial solution for the objective function (eq.16) is estimated and the $Sites_{best}$ parameter is established as a token of feasible solution sets defined by the constraints. Advancing to the next step of the algorithm, bees that have the highest fitnesses are chosen as $EliteBees_{num}$ and sites visited by them are chosen as $EliteSites_{num}$ for neighborhood search.

Then, the algorithm conducts searches in the neighborhood of the selected sites, assigning more bees to search near to the $EliteSites_{num}$. The bees can be chosen directly according to their performance associated with the sites they are visiting. Searches in the neighborhood of the best sites, which represent more promising solutions, are made more detailed by recruiting more bees to follow them than the other selected bees. However, for each visited patch only the bee with the highest fitness will be selected to form the next bee population. In nature, there is no such a restriction but in our case this is applied in order to reduce the number of points to be explored. Finally, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until the optimal solution is found, a stopping criterion is met or the number of permitted iterations has expired. At the end of each iteration, the colony will have two parts to its new population, which are the representatives from each selected patch and other scout bees assigned to conduct random searches.

IV. Experimental results

Along with the proposed Bee Colony algorithm, two other optimization approaches are utilized to produce an optimum solution for both the MAX_{COVER} and the MIN_{COST} problems. The *Branch and Bound* [24] method is able to solve complex optimization problems such as the ones discussed in this work by dividing the basic problem in a summary of simpler sub-problems while covering the needs of the root problem. This capability of limiting the exploration for the optimum solution by the progressive evaluation of the sub-solutions makes it possible to develop or not a branch of the tree representing the total research. The second optimization technique used to produce a candidate solution is a genetic algorithm by formulating an initial population using the topology constraints as genetic operators and the objective function as an initial input. In the following experiments a common test scenario is set using an arbitrary floor plan in order to measure the performance of the proposed solutions.

A. MAX_{COVER} solutions

As mentioned in Section II, in MAX_{COVER} problems the maximum coverage is expected, given a fixed number of cameras of a certain type and orientation. Thus, in Figure 1, the performance of the proposed algorithm along with the other two optimization techniques is illustrated. All three methods were set to use 10 cameras with the same high resolution lens, attached to a corner of the room or to the midpoint of a wall. The Bee Colony (Fig. 1(a)) achieved maximum coverage of 91.2% while the use of Genetic Algorithm (Fig. 1(b)) produced a solution of 89.2% coverage of the room. Moreover, the Branch & Bound method (Fig. 1(c)) achieved 88.2% of maximum coverage of the floor plan.

B. MIN_{COST} solutions

In this particular MIN_{COST} problem the minimum total cost of the final solution is the optimum solution, subject to a target coverage $> 95\%$, taking into account that all points inside the camera frustum are 100% visible, the *Viewing angle* is $45°$, the *Field of View* and the *Resolution* of the lens are $60°$ and $150ppi$ respectively while the *Viewing distance* is set at $2m$. The cost of each camera with a lens attached is \$1300. The Bee Colony (Fig. 2(a)) achieved the maximum coverage of 96.84% using 12 cameras while the Genetic Algorithm (Fig. 2(b)) used 11 cameras to cover the 95.88% of the room. Lastly, the Branch & Bound method (Fig. 2(c)) covered 95.42% of the room using 11 cameras as well. While our method achieved the maximum coverage of the room the cost of the final multi-camera arrangement is more than the other solutions. Generally, the final decision depends on the application in hand to obtain the most balanced solution between coverage of the room and the final cost of the proposed method.

Conclusions

In this paper, the problem of finding the optimal placement of multiple cameras placed in 3D space is presented. The main focus of this research was two broadened problems in the field of camera placement problems: maximizing coverage of the 3D space, given a fixed number of cameras and minimizing the total cost of the proposed arrangement, taking into account several geometrical, optical and reconstructive constraints. The combination of these limitations, formed the examined problem as a continuous optimization one and suitable for a benchmark for the presented optimization technique. The *Bee Colony algorithm* is utilized as it is capable to determine the number of cameras needed to adequately cover the given space while taking into consideration the minimum possible cost for the proposed arrangement as well.

References

[1] J. Ren, M. Xu, J. Orwell, and G. A. Jones, "Multi-camera video surveillance for real-time analysis and reconstruction of soccer games," *Machine Vision and Applications*, vol. 21, no. 6, pp. 855–863, Aug. 2009.

[2] B. Schauerte, J. Richarz, T. Plötz, C. Thurau, and G. Fink, "Multi-modal and multi-camera attention in smart environments," in *Proceedings of the 2009 international conference on Multimodal interfaces*, 2009, pp. 261–268.

[3] J.-M. Lien, G. Kurillo, and R. Bajcsy, "Multi-camera tele-immersion system with real-time model driven data compression," *The Visual Computer*, vol. 26, no. 1, pp. 3–15, May 2009.

[4] V. Chvatal, "A combinatorial theorem in plane geometry," *J. Combin. Theory Ser. B*, vol. 18, no. 39-41, p. 32, 1975.

[5] J. O'Rourke, *Art Gallery Theorems and Algorithms (International Series of Monographs on Computer Science)*. Oxford University Press Inc, Dec. 1987.

[6] J. O'Rourke and K. Supowit, "Some np-hard polygon decomposition problems," *Information Theory, IEEE Transactions on*, vol. 29, no. 2, pp. 181 – 190, mar 1983.

[7] J. Culberson and R. Reckhow, "Covering polygons is hard," *Journal of Algorithms*, vol. 17, no. 1, pp. 2–44, 1994.

[8] H. González-Baños and J.-C. Latombe, "A Randomized Art-Gallery Algorithm for Sensor Placement," in *Proceedings of the seventeenth annual symposium on Computational geometry*, 2001, pp. 232–240.

[9] Urrutia, Jorge, *Art gallery and illumination problems*. Handbook of computational geometry, 2004.

[10] A. Efrat and S. Har-Peled, "Guarding galleries and terrains," *Information processing letters*, vol. 100, no. 6, pp. 238–245, 2006.

[11] S. Ghosh, "Approximation algorithms for art gallery problems in polygons," *Discrete Applied Mathematics*, vol. 158, no. 6, pp. 718–722, 2010.

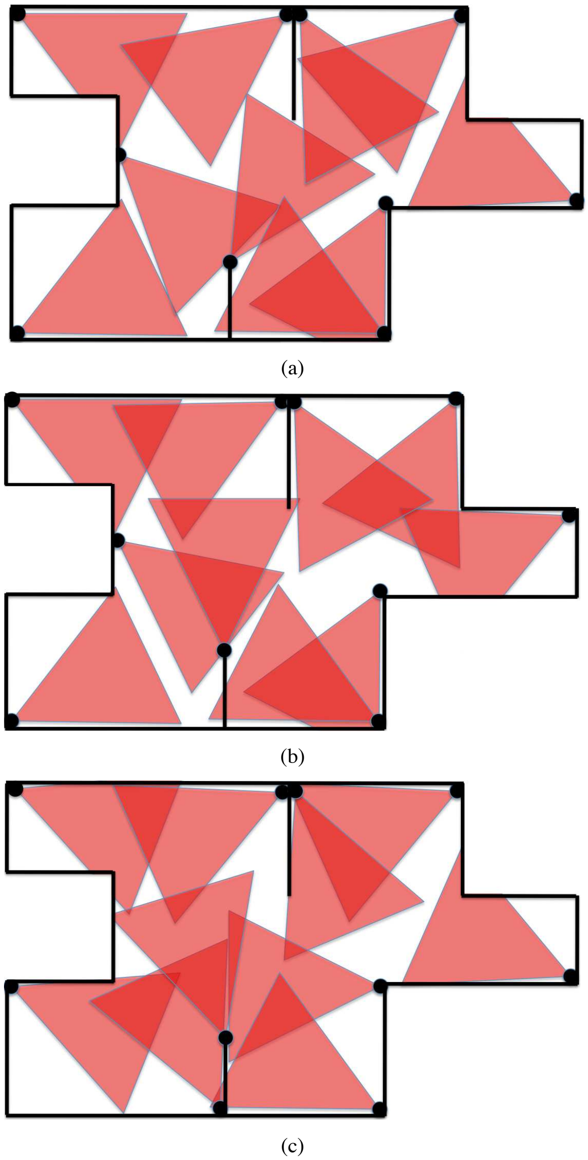

Fig. 1. From the top to bottom, the performance of Bee Colony, Genetic Algorithm and Branch & Bound algorithms for the MAX_{COVER} problem is depicted

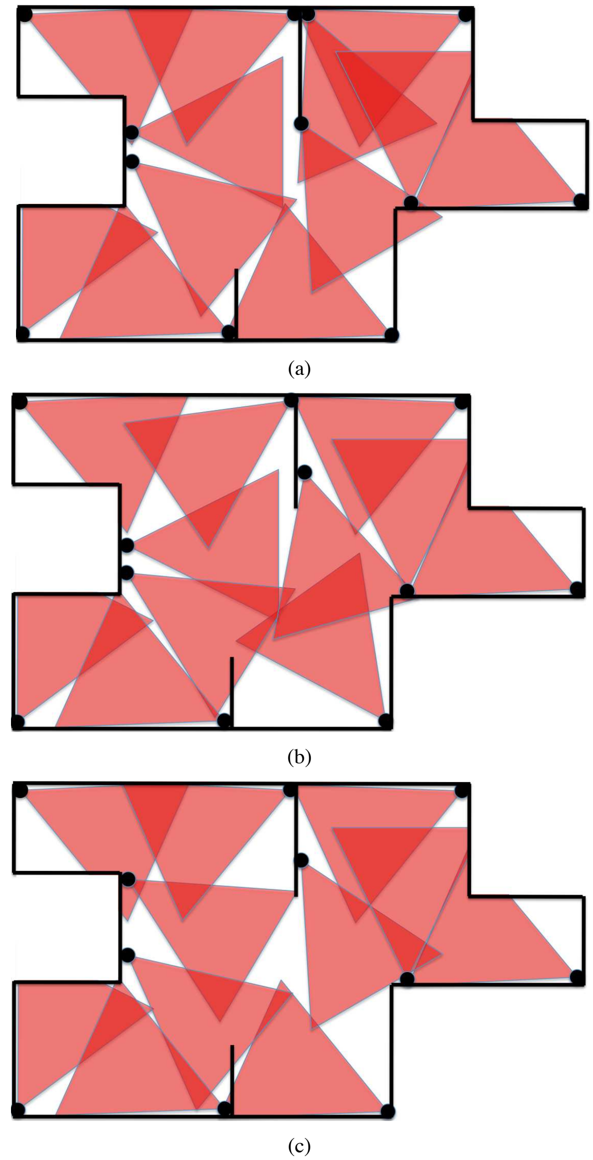

Fig. 2. From the top to bottom, the performance of Bee Colony, Genetic Algorithm and Branch & Bound algorithms for the MIN_{COST} problem is depicted

[12] A. Bottino and A. Laurentini, "A nearly optimal algorithm for covering the interior of an Art Gallery," *Pattern Recognition*, vol. 44, no. 5, pp. 1048–1056, May 2011.

[13] M. C. Couto, P. J. de Rezende, and C. C. de Souza, "An exact algorithm for minimizing vertex guards on art galleries," *International Transactions in Operational Research*, vol. 18, no. 4, pp. 425–448, 2011. [Online]. Available: http://dx.doi.org/10.1111/j.1475-3995.2011.00804.x

[14] J. Gonzalez-Barbosa, T. García-Ramírez, J. Salas, J. Hurtado-Ramos, and J. Rico-Jimenez, "Optimal camera placement for total coverage," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 844–848.

[15] U. Erdem and S. Sclaroff, "Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements," *Computer Vision and Image Understanding*, vol. 103, no. 3, pp. 156–169, 2006.

[16] E. Hörster and R. Lienhart, "On the optimal placement of multiple visual sensors," in *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*. ACM, 2006, pp. 111–120.

[17] Y. Yao, C.-H. Chen, B. Abidi, D. Page, A. Koschan, and M. Abidi, "Can You See Me Now? Sensor Positioning for Automated and Persistent Surveillance," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 1, pp. 101–115, 2010.

[18] S. Y. Chen and Y. F. Li, "Automatic Sensor Placement for Model-Based Robot Vision," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 1, pp. 393–408, Feb. 2004.

[19] K. von Frisch, "Bees: their vision, chemical senses, and language. Rev. ed. Second printing," 1976.

[20] T. Seeley, *The wisdom of the hive: the social physiology of honey bee colonies*. Harvard University Press, 1995.

[21] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, "The bees algorithm–a novel tool for complex optimisation problems," in *Proceedings of IPROMS 2006 Conference*, 2006, pp. 454–461.

[22] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. Oxford University Press, USA, 1999, no. 1.

[23] S. Camazine, *Self-organization in biological systems*. Princeton University Press, 2003.

[24] A. Land and A. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, pp. 497–520, 1960.