EURASIP Journal on Applied Signal Processing 2003:8, 748–756

Evolutionary Computation for Sensor Planning: The Task Distribution Plan

Enrique Dunn

Departamento de Electrónica y Telecomunicaciones, División de Física Aplicada, Centro de Investigación Científica y de Educación Superior de Ensenada, 22860 Ensenada, BC, Mexico
Email: edunn@cicese.mx

Gustavo Olague

Departamento de Ciencias de la Computación, División de Física Aplicada, Centro de Investigación Científica y de Educación Superior de Ensenada, 22860 Ensenada, BC, Mexico
Email: olague@cicese.mx

Received 29 June 2002 and in revised form 29 November 2002

Autonomous sensor planning is a problem of interest to scientists in the fields of computer vision, robotics, and photogrammetry. In automated visual tasks, a sensing planner must make complex and critical decisions involving sensor placement and the sensing task specification. This paper addresses the problem of specifying sensing tasks for a multiple manipulator workcell given an optimal sensor placement configuration. The problem is conceptually divided in two different phases: *activity assignment* and *tour planning*. To solve such problems, an optimization methodology based on evolutionary computation is developed. Operational limitations originated from the workcell configuration are considered using specialized heuristics as well as a floating-point representation based on the random keys approach. Experiments and performance results are presented.

Keywords and phrases: sensor planning, evolutionary computing, combinatorial optimization, random keys.

1. INTRODUCTION

Sensor planning is a growing research area, which studies the development of sensing strategies for computer vision tasks [1]. The goal of such planning is to determine, as autonomously as possible, a group of sensing actions that lead to the fulfillment of the vision task objectives. This is important because there are environments (i.e., dynamic environments with physical and temporal constraints) and tasks (i.e., scene exploration, highly accurate reconstruction) where the specification of an adequate sensing strategy is not a trivial endeavor. Moreover, an effective planner must make considerations that require complex spatial and temporal reasoning based on a set of mathematical models dependent of the vision task goals [2]. Indeed, difficult numerical and combinatorial problems arise, presenting a rich variety of research opportunities. Our approach is to state such problems in optimization terms and apply evolutionary computation (EC) methodologies in their solution [3].

The problem of visual inspection of a complex three-dimensional object requires the acquisition of multiple object images from different viewpoints [4]. Accordingly, to formulate a sensing strategy, an effective planner must consider how the spatial distribution of viewpoints affects a specific task goal, what an adequate configuration for an individual sensor is, how the sensing actions will be executed. These are the kind of general considerations that call for the use of a flexible computing paradigm like EC. This work presents the ongoing development of the EPOCA [5] sensor planning system, giving special attention to the task distribution problem that emerges from a multiple manipulator workcell [6].

The literature provides multiple examples of work dealing with automated sensing planning systems which consider a manipulator using a "camera-in-hand" configuration. The HEAVEN system developed by Sakane et al. [7] is an example in which the camera and light illumination placement are studied. The MVP system developed by Abrams et al. [8] considered the viewpoint planning of one manipulator monitoring the movements of a second robot. The work developed by Triggs and Laugier [9] considers workspace constraints of a robot carrying a camera with the goal of automated inspection. More recently, Whaite and Ferrie [10] developed an uncertainty based approach for autonomous exploration using a manipulator robot. The next best view problem for automated surface acquisition working with a range scanner has been addressed by Pito [11]. Marchand and Chaumette [12] studied optimal camera motion in active vision systems for 3D reconstruction and exploration. Ye and Tsotsos [13] developed a sensor planner system for 3D object search applied in mobile robotics. However, none of these systems have studied the problem of assigning and sequencing the best order of movements that a multiple robot system needs to perform.

This paper is organized as follows. First, the problem statement is given in Section 2. Then, our approach to the task distribution problem using EC is presented in Section 3. In this section, we address the aspects of search space reduction, solution representation, and search heuristics. Experimental results are presented next in order to demonstrate the validity and usefulness of the solution. Finally, conclusions and guidelines for future research are provided to end the paper.

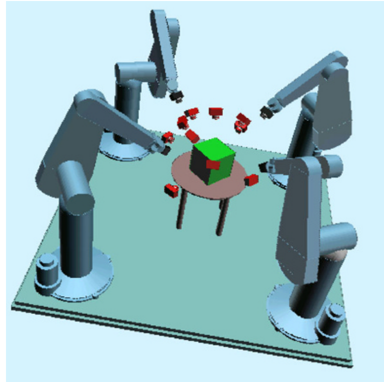

Figure 1: Photogrammetric network simulation of four robots.

2. PROBLEM STATEMENT

The automation of visual inspection tasks can be achieved with the use of manipulator robots, see Figure 1. However, the incorporation of such devices makes additional demands on a sensing planner. In this example, each camera is mounted on the robot hand with the goal of measuring the box on the table. Also, additional floating cameras represent a set of desired viewpoints. The sensing plan must consider not only the constraints and objectives of the particular visual task but also the operational restrictions imposed by the workcell. Additionally, in the case where multiple manipulators are equipped with digital cameras, a problem of robot coordination needs to be resolved. More precisely, sensing actions need to be distributed among the various sensing stations, and an efficient task specification for the entire workcell should be determined. The EPOCA network design module can determine an optimal sensing configuration for multiple cameras converging on a three-dimensional object [14]. We use this configuration as input for our task distribution problem in the proposed multiple robot workcell. It is assumed that the robots move in straight lines between different viewpoints and that each robot must start and finish each tour from a predetermined configuration. In this way, the problem of specifying an efficient task distribution for the manipulator robots consists of the following.

(1) Assigning to each of the robots a set of viewpoints from which to obtain an image, see Figure 2. In other words, determining how many and which viewpoints are to be assigned to each robot.
(2) Deciding on an optimal tour for each of the robots, see Figure 3. This involves specifying the correct order of each viewpoint in a robot's tour.

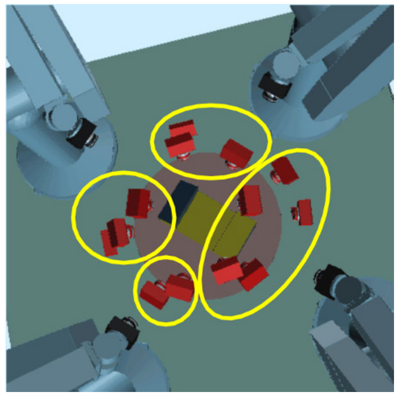

Figure 2: Activity assignment. Each viewpoint is assigned to one of the robots, forming different excluding sets.

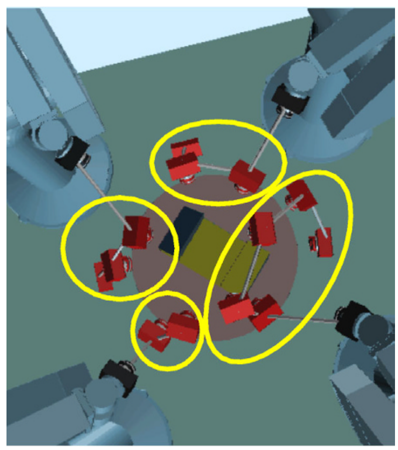

Figure 3: Tour planning. Each of the sets is ordered, specifying the tour to follow each of the robots.

In this way, we have two of the most difficult combinatorial problems in computer science, which are the set partition and traveling salesman problems, see Figures 2 and 3 for the graphical interpretation of these problems. Actually, our task distribution problem consists of a multiple traveling salesman problem instance. The goal is to specify the optimal combination of multiple subtours, with the requirement that every viewpoint specified by the EPOCA network configuration module is visited. In order to describe our task distribution problem, the following definitions are given.

Definition 1 (Photogrammetric network). A photogrammetric network is represented as an ordered set **V** of n three-dimensional viewpoints. Each individual viewpoint is expressed as V_j, where j ranges from $j = 1$ to n.

Definition 2 (Robot workcell). A multirobot active vision system is represented by an ordered set **R** consisting of r robots in the workcell. Each individual robot is expressed by R_i, where i ranges from $i = 1$ to r.

Definition 3 (Operational environment). Each robot has an operational restricted physical space denoted by O_i, where i ranges from $i = 1$ to r.

Accordingly, the problem statement can be expressed as follows.

Definition 4 (Task distribution problem). Find a set of r ordered subsets $X_i \subseteq \mathbf{V}$, where $\mathbf{V} = \{\cup_{i=1}^{r} X_i \mid V_j \in X_i,\ V_j \in O_i\}$ such that the total length traveled by the robots is minimized.

From the above definitions, the activity assignment problem relates each of the n elements of **V** with one of the r possible elements of **R**. Considering that each robot R_i has assigned n_i viewpoints, a problem of sequencing the viewpoints emerges, which we call tour planning. Our goal is to find the best combination of *activity assignment* and *tour planning* in order to optimize the overall operational cost of the task distribution. This total operational cost is produced by adding individual tour costs, Q_i, defined by the Euclidean distance that each robot needs to travel in straight lines among the different viewpoints. Hence, the criterion is represented as $Q_T = \sum_{i=1}^{r} Q_i$. Such a problem statement yields a combinatorial problem which is computationally NP-hard and requires the use of special heuristics in order to avoid an exhaustive search.

3. EC APPROACH TO TASK DISTRIBUTION

Our problem is presented as a combinatorial optimization problem with a large search space. An optimization method based on genetic algorithms is proposed. To obtain a quality solution, three key aspects need to be addressed: *search space reduction, solution representation*, and *search heuristics*. The following sections present our approach to these key aspects in order to develop a global optimization method to solve the task distribution problem.

3.1. Search space reduction

Combinatorial problems generally have to satisfy a given set of competing restrictions. In our task distribution problem, some of these restrictions are straightforward; that is, each viewpoint should be assigned to only one robot, each viewpoint should be visited only once inside a robot tour. On the other hand, implicit restrictions, like the *accessibility* of a robot to a particular viewpoint, need to be determined. Consideration of such restrictions can help reduce the size of the search space. This is relevant because in practice a manipulator has limited workspace, see Figure 4. The method by which such restrictions are computed is presented next.

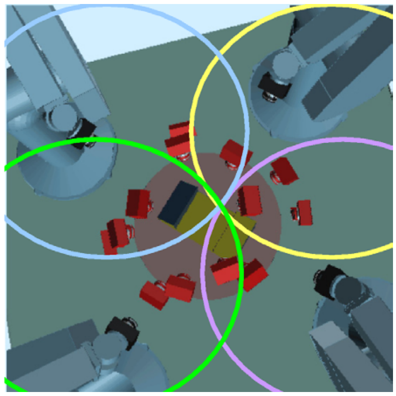

Figure 4: Operational restrictions. The workcell configuration imposes accessibility restrictions. Hence, when a robot reach is limited, it is possible to reduce the search space for the activity assignment phase.

Table 1: Structure ACCESSIBILITY containing the number and the list of robots capable of reaching a particular viewpoint.

Viewpoint ID	Number of robots	List of robots ID's
V_1	r_1	$\text{RobID}_1, \dots, \text{RobID}r_1$
$\vdots$	$\vdots$	$\vdots$
V_n	r_n	$\text{RobID}_1, \dots, \text{RobID}r_n$

Assuming a static and obstacle-free environment, it is reasonable to compute the robots accessibility for a given position and orientation by means of solving the robot inverse kinematic problem. In this work, we consider the PUMA560 manipulator which consists of six degrees of freedom. A three-dimensional computer graphics simulation environment was developed in order to visualize such accessibility restrictions. Multiple manipulators were considered in our computer simulation. The inverse kinematic problem was solved for every robot at each viewpoint. The cases where a robot could access a viewpoint were stored in an auxiliary data structure called ACCESSIBILITY. This structure contains an entry for every viewpoint V_j in order to keep a record of how many and which robots are capable of reaching that particular viewpoint, see Table 1. Such values remain constant throughout the course of task execution, therefore, they only need to be computed once. The above method evaluates the restrictions imposed by the physical arrangement of the workcell, as well as the robot revolute joint limitations. Such operational restrictions are incorporated implicitly as an intrinsic element of our optimization method.

3.2. Solution representation

A representation similar to *random keys* [15] is proposed. In this representation, each viewpoint V_j is assigned a random value S_j in the range $(0, 1)$, allowing for the implementation of very straightforward genetic operators. These

	S_1	S_2	S_3		$\cdots$			S_n
$\mathbf{S} =$	0.41	0.51	0.15	0.79	0.63	0.96	0.84	0.18

Figure 5: Solution encoding. Each of the n viewpoints is assigned a random floating-point value S_i in the range $(0, 1)$. These values are stored in a string $\mathbf{S}$.

values are stored in a representation string denoted by $\mathbf{S}$. Since there are n different viewpoints, $\mathbf{S}$ will consist of n elements, see Figure 5. Random keys use a heuristic we call the *smallest-value-first* heuristic. In our case, the viewpoint with the smallest corresponding value in $\mathbf{S}$ would be the first viewpoint in a given permutation $\mathbf{P}$. The viewpoint with the second smallest value in $\mathbf{S}$ would be the second viewpoint in $\mathbf{P}$, and so forth. In this way, the order of a viewpoint V_j inside a given permutation $\mathbf{P}$ depends on the magnitude of its corresponding value S_j with respect to all the other values in $\mathbf{S}$. To illustrate, given five viewpoints, a possible representation string can be

$$\mathbf{S} = [0.89, 0.76, 0.54, 0.23, 0.62]. \tag{1}$$

The smallest value in $\mathbf{S}$ is found at the fourth position, denoted by S_4. Therefore, V_4 is the first viewpoint in the resulting permutation $\mathbf{P}$. The second smallest value is found in the third position S_3, making V_3 the second viewpoint in $\mathbf{P}$, and so on. The resulting permutation of the five viewpoints is

$$\mathbf{P} = [V_4, V_3, V_5, V_2, V_1]. \tag{2}$$

The random keys approach can be adapted to solve our task distribution problem. The smallest-value-first heuristic avoids the generation of unfeasible solutions common to permutation-based representations. Random keys representation also allows our optimization method to apply genetic operators without the need for additional heuristics.

The convention of encoding a possible solution into a string representation has been specified. The question of how to *describe* the corresponding solution to such a representation is now considered. Recalling the problem statement, initially, there is a set of n viewpoints V_j, and each must be assigned to one of the r possible robots. Using random keys representation, a possible solution is codified into a string $\mathbf{S}$ of n values. As stated in Section 2, we want to optimize the total operational cost Q_T. However, the solution representation $\mathbf{S}$ needs to be decoded into an explicit description of the task distribution problem. Such a description would represent each of the r robot tours. To accomplish this, an auxiliary data structure called TASKS is proposed to represent the global task distribution among robots, see Table 2. This structure has an entry T_i for each robot R_i, which describes that robot tour; that is, T_i lists the sequence of viewpoints assigned to that particular robot. Each of these T_i tours is evaluated to obtain an individual tour cost Q_i, from which the total operational cost Q_T is obtained. The question before us now is how to convert a string representation into a corresponding task distribution description. The following subsection presents the heuristics used by our method to obtain such task distribution description.

Table 2: Structure TASKS containing the list of viewpoints comprising each robot tour T_i.

Robot ID	Number of viewpoints	List of viewpoint ID's
R_1	v_1	$T_1 = [\text{ViewID}_1, \ldots, \text{ViewID}v_1]$
$\vdots$	$\vdots$	$\vdots$
R_r	v_r	$T_r = [\text{ViewID}_1, \ldots, \text{ViewID}v_r]$

3.3. Search heuristics

A solution representation $\mathbf{S}$ needs to be evaluated. Such evaluation is applied to the task distribution description contained in TASKS. Hence, a mapping $M : \mathbf{S} \rightarrow \text{TASKS}$ is necessary. The mapping M assigns and sequences the viewpoints among the different robots and stores the results in the structure TASKS. The mapping M makes use of the solution representation data structures $\mathbf{S}$ and TASKS, as well as the precomputed operational restrictions stored in ACCESSIBILITY. The two distinct phases of activity assignment and tour planning are presented separately.

3.3.1 Activity assignment

The *activity assignment* problem allocates each of the viewpoints V_j to one of the possible robots. The goal is to provide an initial unsequenced set of individual robot tours T_i using the following steps.

Step 1. Obtain the r_j number of robots capable of reaching that particular viewpoint by consulting the ACCESSIBILITY structure, see Table 1.

Step 2. Divide the interval $(0, 1)$ into r_j equally distributed segments in order to determine the size of a comparison segment $\text{Seg} = 1/r_j$.

Step 3. Calculate in which k segment the random value S_j resides, that is, $k = \text{Int}(S_j/\text{Seg}) + 1$.

Step 4. Assign the viewpoint V_j to the kth robot in the corresponding entry in the ACCESSIBILITY structure. In this way, the assigned robot index i is given by RobID_k, which is found on the entry that corresponds to V_j inside the ACCESSIBILITY table.

Step 5. Append V_j to the list of viewpoints, T_i assigned to the ith robot. The tour description T_i is stored in the TASKS structure.

A graphical description of these heuristic steps is shown in Figure 6. The series of actions performed in the activity assignment phase are based on the compliance with operational restrictions, and in doing so, assure that any codified string $\mathbf{S}$ brings a valid solution to the assignment problem. Based on such strategy, each possible codification string $\mathbf{S}$ has only one possible interpretation. After executing this series of steps, each viewpoint is assigned to a robot. The viewpoints assigned to a single robot R_i are grouped into a set T_i. Each

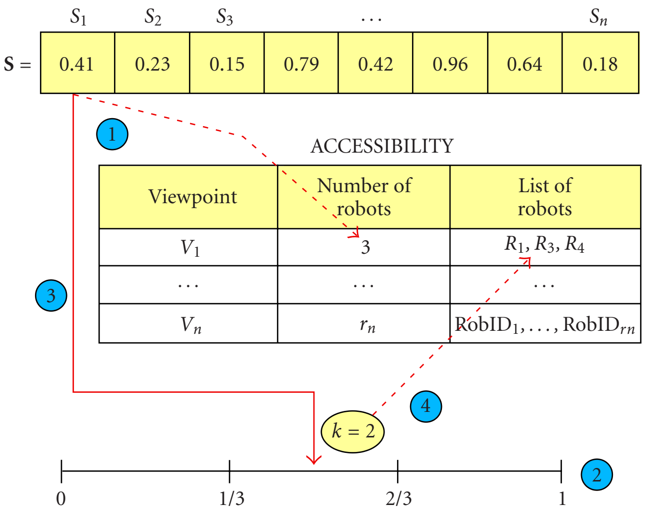

FIGURE 6: Activity assignment heuristics. The diagram shows Steps 1 through 4, corresponding to the assignment phase.

T_i represents a tour of viewpoints assigned to that particular robot and these tours are stored in the structure TASKS. Until this point, the order of each viewpoint inside a given tour has not been specified. This is the problem we approach next.

3.3.2 Tour planning

The *tour planning problem* consists of correctly sequencing each of the r robot tours T_i stored in the structure TASKS. These tours are initially obtained from the activity assignment phase presented above, in which every viewpoint V_j is assigned to one of the r possible robots R_i. The goal of the tour planning phase is to minimize the total operational cost Q_T. This situation is equivalent to solving r different *traveling salesman problems*. The smallest-value-first heuristic can be applied to sequencing problems such as the one presented here. Unfortunately, the rules by which the preceding assignments were made in Steps 1 through 4 produce undesirable tendencies in the representation values S_j that correspond to each tour specification T_i. This is due to the deterministic heuristic applied for robot assignment. As a consequence, the values corresponding to the viewpoints contained in T_i will be, on the average, higher than those corresponding to the viewpoints in T_{i-1} and will create a bias inside each T_i when directly applying the smallest-value-first heuristic. Therefore, the values inside $\mathbf{S}$ need to be adjusted to eliminate such unwanted properties. This is accomplished by the following heuristic steps.

Step 6. Recall in which of the k possible segments of the range $(0, 1)$ lies the S_j value used in the assignment phase.

Step 7. Calculate the value S'_j in the range $(0, 1)$ that reflects the relative position of S_j inside the kth segment. For example, consider the value 0.70 which lies inside the range $(0.60, 0.80)$. This value lies exactly in the middle, hence its corresponding value in the range $(0, 1)$ is 0.5. A graphic description of this heuristic is presented in Figure 7.

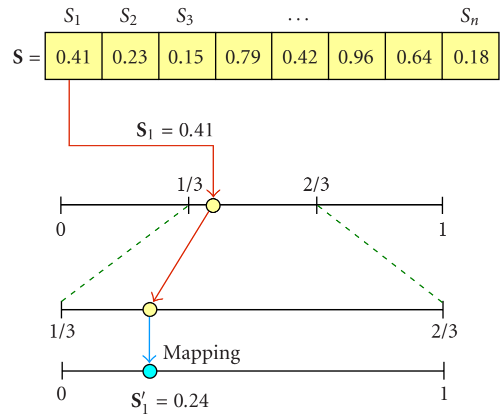

FIGURE 7: Mapping of the representation string values. Each of the values contained in $\mathbf{S}$ is adjusted before applying the smallest-value-first heuristic to the values stored in TASKS.

TASKS

Robot ID	No. of views	List of viewpoints
R_1	3	$T_1 = [V_1, V_3, V_8]$
⋮	⋮	⋮
R_m	r_m	$T_m = [\text{ViewID}_1, \ldots, \text{ViewID}r_m]$

	S_1	S_2	S_3	...				S_8
$\mathbf{S} =$	0.24	0.73	0.04	0.34	0.77	0.69	0.27	0.46

Applying the smallest-value-first heuristic the list T_1 is rearranged in the following manner

Robot ID	No. of views	List of viewpoints
R_1	3	$T_1 = [V_3, V_1, V_8]$

FIGURE 8: Tour planning. The smallest-value-first heuristic is applied to each robot tour considering the previously adjusted values in $\mathbf{S}$.

Step 8. Update S_j to store the new value S'_j.

Step 9. Apply the smallest-value-first heuristic to each of the unordered robot tours T_i using the values stored in $\mathbf{S}'$, see Figure 8.

These series of steps ensure an unbiased tour sequencing, hence, empowering the search algorithm to more effectively seek out a global optima from a very large and complex search space.

4. EXPERIMENTATION AND RESULTS

The solution presented in the previous sections for the task distribution problem was incorporated into an extension of the functionality of the EPOCA system developed by Olague

[5]. EPOCA solves the photogrammetric network design problem for complex objects. The problem of task distribution emerges as a result of the photogrammetric network design performed by EPOCA. The system can be classified as an EC-based system that addresses the complex goal of automating the planning of sensing strategies for accurate three-dimensional reconstruction.

Two different experiments are presented next: the first is a simple scenario intended to illustrate our method's functionality; the second experiment is somewhat more complex and its goal is to show the effectiveness and flexibility of our system.

4.1. *Experiment A*

This experiment consists of eight viewpoints to be distributed among four manipulators. The viewpoints are stacked into four pairs, each pair arranged beneath one of the robots initial position, see Figure 9. The optimal task distribution for this example can be obtained using a greedy heuristic. Hence, such an experiment might seem trivial, but it will exemplify our method's functionality.

Operational restrictions are computed first, with the goal of determining which robots can access a particular viewpoint. As mentioned in Section 3, to compute such restrictions, the inverse kinematic problem is solved for every robot at each viewpoint. The results of such validations are stored in the structure ACCESSIBILITY. The physical arrangement of the robots for Experiment A is such that every camera can be reached by three different robots, see Table 3.

The genetic algorithm works with a population of codified strings, selecting the best individuals for reproduction. Such reproduction process combines the characteristics of two selected *parent* solutions and provides two new *offspring* solutions which, in turn, will be part of the next *generation* of solutions. This process is repeated in an iterative manner until a certain number of generations is executed. At the end of this iterative process, we obtain a set of possible solutions. One of those individuals, which represented the optimal solution, was given by the following random keys representation:

$$\mathbf{S} = [0.72, 0.71, 0.32, 0.14, 0.81, 0.80, 0.27, 0.07]. \tag{3}$$

After the assignment heuristic, we determine in which of the k segments each element S_j resides. For the first viewpoint V_1, there are three possible robots to be assigned, see Table 3; hence, the comparison segment Seg $= 1/3 = 0.33$. In this way, following Steps 1 through 5, the corresponding representation value $S_1 = 0.72$ is determined to be in the third segment, which is delimited by $(0.66, 1.00)$. Therefore, the robot to be assigned is the third robot on V_1's entry on the structure ACCESSIBILITY, in this case RobID $= 3$. The corresponding robot to be assigned to each viewpoint V_j is given by

$$\text{Robot} = [R_3, R_3, R_1, R_1, R_4, R_4, R_2, R_2]. \tag{4}$$

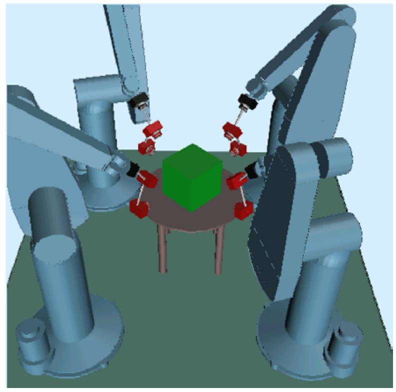

Figure 9: Eight viewpoints are to be distributed among four manipulators. Viewpoints are depicted as individual cameras and solid lines connected such cameras illustrate each robot tour corresponding to an optimal task distribution.

Table 3: ACCESSIBILITY restrictions calculated for Experiment A, depicted in Figure 9.

Viewpoint ID	Number of robots	List of robots ID's
V_1	$r_1 = 3$	R_1 R_2 R_3
V_2	$r_2 = 3$	R_1 R_2 R_3
V_3	$r_3 = 3$	R_1 R_3 R_4
V_4	$r_4 = 3$	R_1 R_3 R_4
V_5	$r_5 = 3$	R_1 R_2 R_4
V_6	$r_6 = 3$	R_1 R_2 R_4
V_7	$r_7 = 3$	R_2 R_3 R_4
V_8	$r_8 = 3$	R_2 R_3 R_4

At this point, we have an appropriately assigned set of viewpoints. The values contained in $\mathbf{S}$ will now be adjusted in accordance with Steps 5 through 9 so that the smallest-value-first heuristic can be applied to the viewpoints assigned to each robot. For the first viewpoint, its corresponding value S_1 is adjusted as follows. Recall that $S_1 = 0.72$ resides on the third segment which is delimited by $(0.66, 1.00)$. The corresponding value of 0.72 on the range $(0, 1)$ with respect to the third segment just mentioned is given by the value 0.18. Applying these steps to every value in $\mathbf{S}$ yields

$$\mathbf{S} = [0.18, 0.15, 0.96, 0.42, 0.45, 0.42, 0.81, 0.21]. \tag{5}$$

Once the values in $\mathbf{S}$ have been adjusted, applying the smallest-value-first heuristic rearranges TASKS as shown in Table 4.

Twenty trials were executed and this global minimum distribution was reached in every single execution in an average of 15.1 generations.

4.2. *Experiment B*

This experiment presents a complex planar object which is measured by four manipulators. The goal is to distribute the

TABLE 4: TASKS for an optimal solution in Experiment A after the tour planning phase.

Robot ID	Number of viewpoints	List of viewpoint ID's
1	2	$T_1 = [V_4 V_3]$
2	2	$T_2 = [V_8 V_7]$
3	2	$T_3 = [V_2 V_1]$
4	2	$T_4 = [V_6 V_5]$

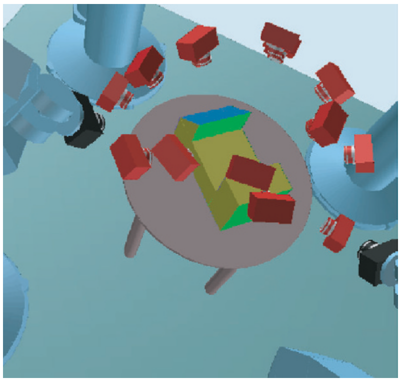

FIGURE 10: Thirteen viewpoints are to be distributed among four manipulators. Viewpoints are depicted as individual cameras.

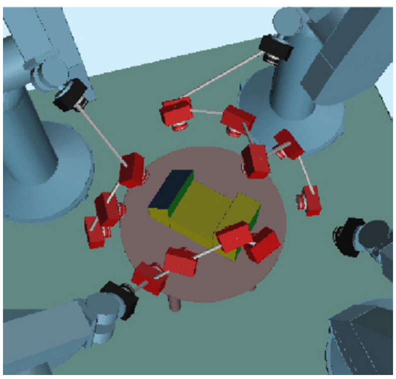

FIGURE 11: Best solution found by the genetic algorithm for the configuration shown in Figure 10.

photogrammetric network consisting of 13 cameras in an optimal manner, see Figure 10. Working with this fixed configuration, we executed several tests. First, to test our method's functionality, we executed the task distribution planner. Several possible solutions are obtained over the course of multiple executions, two of such solutions are depicted in Figures 11 and 12. Notice that the best solution found, represented in Figure 11, does not incorporate all of the available robots. Figure 12 shows a more typical solution which is also found by our system.

In order to test the method's adaptability, two of the four manipulator robots were disabled. This additional restriction is reflected only on changes to the values stored in Table 5.

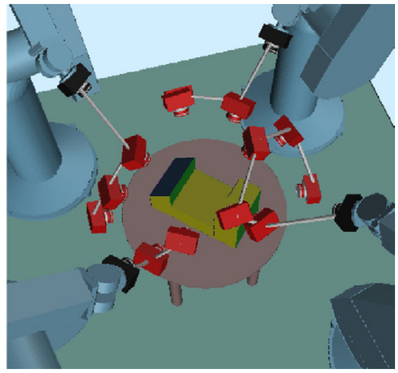

FIGURE 12: Another solution found by the system that corresponds to the configuration shown in Figure 10.

TABLE 5: ACCESSIBILITY restrictions calculated for Experiment B, depicted in Figure 10.

Viewpoint ID	Number of robots	List of robots ID's
V_1	$r_1 = 2$	R_2, R_4
V_2	$r_2 = 2$	R_2, R_3
V_3	$r_3 = 2$	R_1, R_4
V_4	$r_4 = 2$	R_1, R_4
V_5	$r_5 = 2$	R_1, R_4
V_6	$r_6 = 2$	R_2, R_3
V_7	$r_7 = 2$	R_2, R_4
V_8	$r_8 = 2$	R_2, R_3
V_9	$r_9 = 2$	R_1, R_3
V_{10}	$r_{10} = 2$	R_1, R_3
V_{11}	$r_{11} = 3$	R_1, R_2, R_3
V_{12}	$r_{12} = 3$	R_1, R_2, R_4
V_{13}	$r_{13} = 3$	R_1, R_2, R_4

The system is expected to distribute tasks among the two remaining robots. Results from such tests are shown in Figures 13 and 14. In these cases the *activity assignment* problem becomes visually more simple to resolve, but the difficulty of the *tour planning* problem becomes more evident since each tour will consist of more viewpoints.

Since our approach is based on EC techniques, the determination of the task distribution plan is the product of the evolution process over a population of possible solutions. Therefore, fitness values of each of these individuals, and of the population in general, reflect the effect of such evolution. In this way, the population fitness values evolve over the course of several generations until an optimal solution is found, see Figure 15. The stepwise decrements in the best fitness line point out the combinatorial aspect of our search, while the average fitness confirms the positive effect of the evolution process.

While great detail has been given to the special heuristics used in our approach, the behavior of the curves presented in

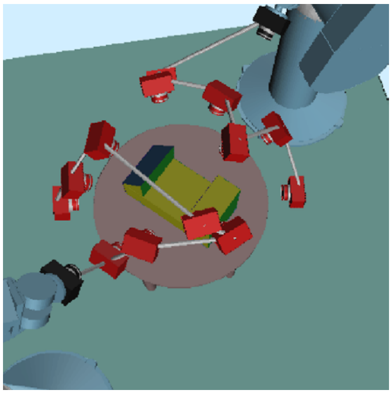

FIGURE 13: Solution found by the system for the case where a pair of robots were disabled from the configuration shown in Figure 10.

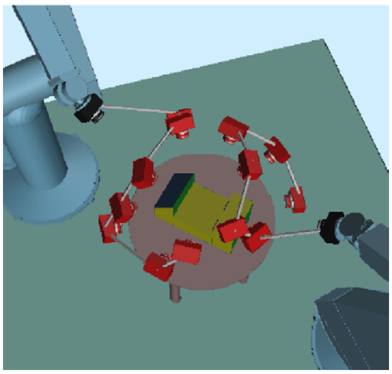

FIGURE 14: An environment similar to Figure 13 showing the system's flexibility to changes in the workcell configuration.

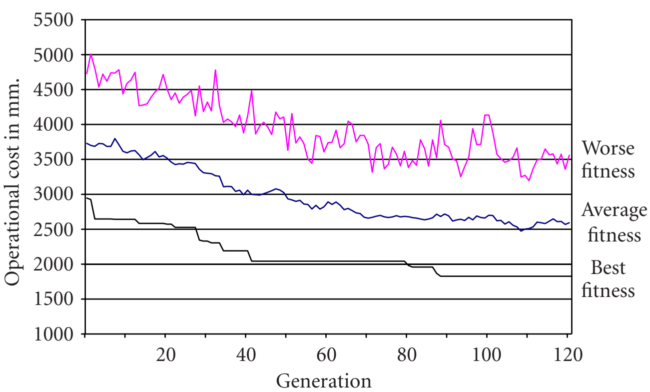

FIGURE 15: Population fitness over the evolution process.

Figure 15 and the overall performance depend on the genetic algorithm operational parameters. A single point crossover operator, subject to a probability $P_c = 0.95$, was utilized. Furthermore, the mutation operator consisting of an additive value obeying a normal distribution $N(0, 0.2)$ for each of the elements in the representation string was also applied according to a probability $P_m = 0.001$.

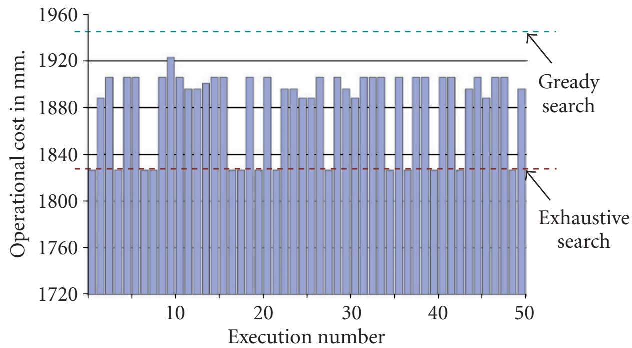

FIGURE 16: Genetic algorithm performance over multiple executions. The obtained solutions are always better than a greedy search, reaching the global optima 14 out of 50 times.

An appreciation of the effectiveness of the proposed methodology is obtained from the comparison of its solutions against those offered by alternative methodologies. The proposed methodology is compared to an exhaustive search and a greedy heuristic. The results for the fixed configuration shown in Figure 10 are presented in Figure 16. As the figure illustrates, our algorithm consistently outperforms a greedy heuristic in terms of the quality of the proposed solutions. The advantage obtained with the genetic algorithm approach refers to the computational cost; considering the EC algorithm requires about 3 seconds against 14 hours for an exhaustive search. On the other hand, our approach reaches a global optima 28% of the time over the course of 50 executions, coming within an average of 2.9% to global optima. As these results reflect, there is an obvious compromise between solution quality and computational efficiency.

5. CONCLUSIONS AND FUTURE WORK

The development of an effective sensor planner for automated vision tasks implies the consideration of operational restrictions as well as the vision tasks objectives. This work presents a solution for the task distribution problem inherent to multiple robot workcells. The problem is conceptualized as two separate combinatorial problems: activity assignment and tour planning. A genetic algorithm-based strategy that concurrently solves these problems was presented along with experimental results. The approach employs auxiliary data structures in order to incorporate accessibility limitations and to specify a task distribution plan. The evolutionary nature of the optimization method allows for multiple approximate solutions of the optimization problem to be found over the course of several executions. Performance considerations support the use of the proposed methodology compared to a greedy heuristic or an exhaustive search.

Future work can consider the robot motion planning problem presented when there are obstacles in the environment or when the manipulator can collide with each other. Also, the representation scheme can be modified to use two values instead of adjusting the original representation string by heuristic means. Furthermore, the genetic operators can be modified in search of improving the evolutionary algorithm performance. Also, a rigorous analysis of the properties of the heuristics used is needed. At present, we are working toward a real implementation of our algorithms for intelligent sensor planning.

ACKNOWLEDGMENTS

This research was founded by Contract 35267-A from CONACyT and under the LAFMI Project. The first author was supported by scholarship 142987 from CONACyT. Figures 1, 2, 3, 4, 9, 10, 11, 12, 13, and 14 were generated with software written at the Geometry Center. The authors thank the anonymous reviewers for their suggestions which greatly helped improve this paper.

REFERENCES

[1] K. A. Tarabanis, P. K. Allen, and R. Y. Tsai, "A survey of sensor planning in computer vision," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 86–104, 1995.

[2] J. Miura and K. Ikeuchi, "Task-oriented generation of visual sensing strategies in assembly tasks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 126–138, 1998.

[3] G. Olague and R. Mohr, "Optimal camera placement for accurate reconstruction," *Pattern Recognition*, vol. 35, no. 4, pp. 927–944, 2002.

[4] T. S. Newman and A. K. Jain, "A survey of automated visual inspection," *Computer Vision and Image Understanding*, vol. 61, no. 2, pp. 231–262, 1995.

[5] G. Olague, *Planification du placement de caméras pour des mesures 3D de précision*, Ph.D. thesis, Institut National Polytechnique de Grenoble, France, October 1998.

[6] G. Olague and E. Dunn, "Multiple robot task distribution: Towards an autonomous photogrammetric system," in *Proc. IEEE Systems, Man and Cybernetics Conference*, vol. 5, pp. 3235–3240, Tucson, Ariz, USA, October 2001.

[7] S. Sakane, R. Niepold, T. Sato, and Y. Shirai, "Illumination setup planning for a hand-eye system based on an environmental model," *Advanced Robotics*, vol. 6, no. 4, pp. 461–482, 1992.

[8] S. Abrams, P. K. Allen, and K. A. Tarabanis, "Dynamic sensor planning," in *Proc. IEEE International Conf. on Robotics and Automation*, Atlanta, Ga, USA, May 1993.

[9] B. Triggs and C. Laugier, "Automatic task planning for robot vision," in *Proc. Int. Symp. Robotics Research*, Munich, October 1995.

[10] P. Whaite and F. P. Ferrie, "Autonomous exploration: Driven by uncertainty," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 193–205, 1997.

[11] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1016–1030, 1999.

[12] E. Marchand and F. Chaumette, "Active vision for complete scene reconstruction and exploration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 1, pp. 65–72, 1999.

[13] Y. Ye and J. K. Tsotsos, "Sensor planning for 3D object search," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 145–168, 1999.

[14] G. Olague, "Automated photogrammetric network design using genetic algorithms," *Photogrammetric Engineering & Remote Sensing*, vol. 68, no. 5, pp. 423–431, 2002, Paper awarded the "2003 First Honorable Mention for the Talbert Abrams Award", by ASPRS.

[15] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA Journal on Computing*, vol. 6, no. 2, pp. 154–160, 1994.

Enrique Dunn received a computer engineering degree from Universidad Autónoma de Baja California, in 1999. He obtained the M.S. degree in computer science from CICESE, Mexico, in 2001. Currently, Dunn is working towards the Ph.D. degree at the Electronics and Telecommunications Department, Applied Physics Division, CICESE, Mexico. His research interests include robotics, combinatorial optimization, evolutionary computation, close range photogrammetry, and 3D simulation. He is a student member of the ASPRS.

Gustavo Olague holds a Bachelor's degree (Honors) in Electronics Engineering and a Master's degree in computer science from the Instituto Tecnológico de Chihuahua, Mexico, in 1992 and 1995, respectively. He received the "Diplôme de Doctorat en Imagerie, Vision et Robotique" (Ph.D.) from Institut National Polytechnique de Grenoble, France, in 1998. From 1999 to 2001, he was an Associate Professor of computer science and in 2002, he was promoted to Professor of the Applied Physics Division at CICESE, Mexico. Dr. Olague is a member of the ASPRS, ISGEC, IEEE, IEEE Computer Society, IEEE Robotics and Automation, IEEE SMC and RSPSoc. Dr. Olague has served on numerous Technical Committees and has been invited to lecture at universities in France, Spain, and Colombia. He has served as Chair and Cochair at numerous international conferences like the ASPRS 2001 and 2003 during the Close-Range Photogrammetry session and the IEEE SMC 2001 Robotics session. He also had visiting appointments at the Technische Universität Clausthal, Germany and the LAAS, France. His research interests include robotics, computer vision, and, in particular, the coupling of evolutionary computation in those two research domains (autonomous systems and visual perception). Dr. Olague is recipient of the 2003 First Honorable Mention for the Talbert Abrams Award.