Optimal robot base placements for coverage tasks

Rawan Kalawoun[1,2,3], Sébastien Lengagne[1,3] and Youcef Mezouar[1,2]

***Abstract*— Robotic coverage problem is addressed in several fields: painting, stripping, grist-blasting, etc. In multi-robot systems, the collaboration between robots reduces the cycle time and increases the coverage task accuracy. However, the robot base placement must be deeply studied to attend those goals. In this article, we propose a new approach in order to assign tasks within a multiple robots system. In addition, we develop an optimization strategy to find the optimal number of robots with their optimal poses required to cover the entire surface. To assess our method, our algorithm was tested on regular surfaces such as cylinder and hemisphere, and on a complex surface represented in a car shell.**

I. Introduction

The surface coverage using robots is a common problem divided into two cases according to the robot state during the task: the robot could be either static or mobile. For the static case, the robot is positioned at a fixed point to cover a surface, e.g. to achieve painting, stripping, or sand-blasting tasks [1], [2], [3]. In that case, the surface is supposed totally reachable from a fixed point and the coverage problem solution returns the optimal trajectories of the end-effector needed to sweep the entire surface. In the mobile robot case, the robot can move to achieve its task like in demining, inspection and agricultural fields coverage. The optimal paths of the mobile robots needed to cover an environment are computed [4], [5], [6], [7].

A third type of coverage problems can be defined when the robot is fixed and the surface is larger than its workspace. In that case, the surface can not be covered from one given position and the coverage problem consists to reposition the robot(s) to cover a surface under the assumption that the coverage task can not be done continuously and needs to be stopped when repositioning the robot. For instance, airplane stripping and building facades refurbishing are some coverage tasks that require repositioning the robots. Another example of such tasks is the car stripping using KUKA Light Weight Robots (LWRs) shown in Figure 1.

This article treats the third problem by using an optimal robot base placements strategy. Optimal robot positioning can decrease the task cycle time and increase its efficiency as well as its accuracy. To reach those goals, the robot base placement should be studied in combination with the coverage problem.

The placement of single robot has been tackled in several domains. Genetic algorithms are used to optimize robot base placements in manufacturing and underwater environments [8], [9]. In milling domains, the robot pose maximizing its manipulability is chosen using genetic algorithm [10]. In [11] the inverse reachability representation is used to compute robot base poses. The above base placements optimization approaches were extended to deal with multiple industrial robots for the coverage tasks. Hassan, Liu et $al.$ proposed a new strategy to distribute the work between robots for coverage tasks assuming that a reasonable number of robots is intuitively chosen based on the size of the object [12], [13]. The strategy consists in finding an optimal base placement and the visiting sequence of the base placements by each robot. A combination of Simulated Annealing and Genetic algorithm optimization is used to find the optimal robot base placement in [13]. Despite the relevance of this strategy, it suffers from several disadvantages:

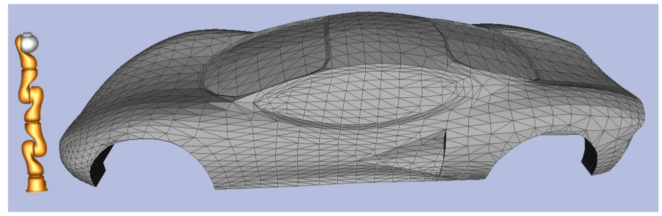


Fig. 1. Car stripping using a KUKA robot

1) The number of robots is harder to guess when the surface gets more complex.
2) The end-effector trajectories used as inputs for base placement optimization are not optimal. They are generated without considering the robot poses.
3) After the base placement optimization, area partitioning and allocation of the surface between different robot poses is required to improve the end-effector trajectories [14].
4) The total surface coverage is not guaranteed.

To deal with those issues, we propose a new approach for a complete coverage using a multiple robots system. An important feature of our approach is that it does not only find the optimal pose of robots bases but it also provides the optimal number of robots $\tilde{N}$ needed to cover the whole surface. Contrary to the state of the art, the knowledge of the total number of robots is not required in our approach.

The optimization of the number of robots with their poses is an extension of the Art Gallery Problem that aims to find the smallest number of guards required to guard the art gallery. Chvátal was the first to tackle this problem in 1973 [15], [16]. Over the years, Art Gallery Problem has been studied in robotics, optimization, vision computational

[1]CNRS, UMR 6602, Pascal Institute, 63171 Aubière, France
[2]SIGMA Clermont, Pascal Institute, BP10448, 63000 Clermont-Ferrand firstname.lastname@sigma-clermont.fr
[3]Clermont-Auvergne University, Pascal Institute, BP10448, 63000 Clermont-Ferrand, France firstname.lastname@uca.fr

graphics, etc [17]. For instance, it was used to optimally position TV cameras in a closed room, to distribute the lighting sources in a small room, or to find the positions of different radar stations in a mountain [18]. It is also used for military goals especially during infiltrating an area and clearing it of threats. Recently, optimal cameras and sensors placement have been deeply studied. A survey on different optimization algorithms used for cameras placement can be found in [19]. The survey differentiates between the FIX and MIN cameras placement problems. FIX is the optimization of a defined number of cameras placement to maximize the coverage of a surface. MIN is the computation of the optimal number of cameras in order to totally cover the surface. In this review, Greedy algorithm was suggested for solving the MIN problem. Moreover, FIX algorithm was applied on a set of different cameras for finding the minimum number required. Those algorithms can be adapted to solve the coverage problem with a set of robots. However, their relevance on complex surfaces is limited by the complexity of the shape needed to be covered. In this paper, we propose an extended optimization algorithm to find the optimal number of robots and their base placements.

The rest of this paper is organized as follows: the problem formulation is presented in Section II, Section III describes the proposed approach for finding the optimal number of robots and their base placements required to cover the whole surface. Additionally, the proposed optimization algorithm is presented. Finally, the simulations results with a cylinder, a sphere, and an action car are presented in Section IV.

II. Problem Statement

Without loss of generality and to illustrate the proposed coverage strategy, robotized stripping processes are considered. Multiple robotic systems composed of mobile platforms with the arms mounted on them were used for surface stripping. The stripping tool is mounted on the end-effector of the manipulator. This technique of paint removal can be applied on large surfaces like air-planes, buildings, cars, etc. Since large surfaces can not be totally covered by a single robot pose, repositioning of the robot is necessary to ensure a complete stripping of the surface. Mobile platforms can be moved manually or automatically between various poses and the stripping process needs to be stopped during their motion.

Our approach aims to find the optimal number of robots N and their base placements required to totally cover the surface. The optimization relies on two parameters: the shape of the surface to be stripped, and the robot workspace. Defining $\mathbb{T}$ as the set of poses $\mathbb{T} = \{\mathbf{T}_i \in \mathrm{SE}(3), 1 \leq i \leq N\}$, the coverage problem can be formulated as follows:

$$\begin{aligned} \min_{N,\mathbb{T}} \quad & N \\ \text{Subject to} \quad & g(N,\mathbb{T},S) = 0 \\ & h(N,\mathbb{T}) \leq 0 \end{aligned} \tag{1}$$

where:
N: the optimal number of robots,
S: the surface to be stripped,
$g(N,\mathbb{T},S) = \bigcup_{j=1}^{N} C(S,\mathbf{T}_j) - S$: the function to test if the surface is totally covered,
$h(N,\mathbb{T})$: the set of additional constraints.

It can be noticed that the task can be achieved using one robot to N robots. Obviously, when the number of robots increases the coverage task can achieved faster. If one robot is only used, it has to be displaced N times while if N robots are used then the task can be achieved at once without moving the robots.

In addition, additional constraints can be taken into account during the optimization using h. For instance, the distance between any two robots can be controlled using a constraint implemented in h. In this paper, the robot constraints are taken into account by projecting them into the surface to be covered using the constraint projection function C. Thus, C returns the reachable part of the surface from a given pose $\mathbf{T}_j$ taking into account the different robot constraints. The function C returns the intersection between the robot workspace and the surface to be covered.

III. General Framework

The general framework that deals with multiple robotic systems for coverage problems is summarized in the flowchart presented in Figure 2.

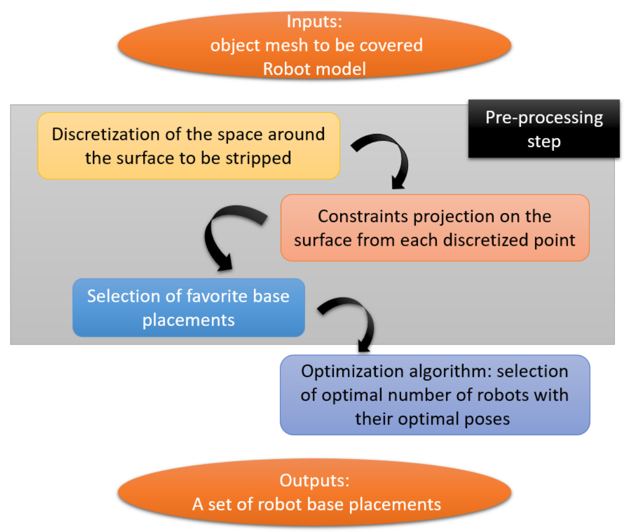


Fig. 2. General framework to optimize the number of robots and their poses.

The inputs are the 3D mesh model of the object surface and the robot workspace. Those inputs are used for the pre-processing step and the optimization algorithm that generates the set of robot poses. The surfaces are modelled using a triangular grid model in common STL (STereo Lithography): the Delaunay triangulation. In this representation, the surface is defined by a point cloud where the points are connected using edges forming small uniform triangles facets.

In this article, a trade-off between accuracy and computation time is made by discretizing the search space. At this aim, a pre-processing step is proposed to turn the continuous problem (1) into a discrete one as presented in Section III-A. Then, we propose an optimization algorithm to solve this discrete problem in Section III-C.

A. Pre-processing step

The pre-processing step turns the continuous 6-D search space of the robot poses into a finite set $\mathbb{F}$ of favorite robot poses. It is composed of three main steps:

1) Search space discretization: this step decomposes the continuous available space around the object into a set $\mathbb{P}$ of possible robot poses. Two adjacent poses are separated by a discretization step s along x, y or z axis. In this article, s is chosen as 0.1% of the length of the manipulator arm.

2) Constraint projection: from each pose $\mathbf{P}$ of $\mathbb{P}$, we assign a score $C(S, \mathbf{P})$ that describes the percentage of the reachable surface S_i. The constraint projection function is used to compute S_i. From a given pose $\mathbf{P}$, $C(S, \mathbf{P}) = 0$ if the robot does not reach the surface or if it provides some unavoidable collisions between its body and the 3D surface. Otherwise, $C(S, \mathbf{P}) = 100\dfrac{S_i}{S}$ that is the percentage of the covered surface S_i regarding the total surface S from a given pose $\mathbf{P}$.

If the robot workspace is sensitive to the orientation along z axis, a simple discretization process is performed in order to define the optimal orientation that maximizes the score $C(S, \mathbf{P})$. Hence, the set of robot poses $\mathbb{P}$ contains all possible poses with different $\{x, y, z\}$ positions and the nearly optimal orientation.

3) Favorite selection : the final step of the pre-processing step is to determine the favorite poses set $\mathbb{F} \subset \mathbb{P}$. The poses in $\mathbb{F}$ are all poses of $\mathbb{P}$ having a score greater than a given threshold $C(S, \mathbf{P}) > t$. The correspondent S_i to every favorite position $\mathbf{F} \in \mathbb{F}$ is part of the set of reachable surface part $\mathbb{C}$.

B. Discrete optimization problem

The preparation phase turns the continuous problem (1) into the following discrete optimization problem:

$$\begin{array}{ll} \min\limits_{\tilde{N}, \tilde{\mathbb{T}} \subset \mathbb{F}} & \tilde{N} \\ \text{Subject to} & g(\tilde{N}, \tilde{\mathbb{T}}, \tilde{\mathbb{C}}) = 0 \\ \forall l, m \in [1, \tilde{N}] & h_{lm}(\tilde{\mathbb{T}}) \leq 0 \end{array} \quad (2)$$

With:

- $\tilde{N}$: the minimal number of robots,
- $\tilde{\mathbb{T}}$: the set of optimal poses of the robots such that $\tilde{\mathbb{T}} = \{\tilde{\mathbf{T}}_i \in \text{SE}(3), 1 \leq i \leq \tilde{N}\}$,
- $\tilde{\mathbb{C}}$: the set of reachable part of the surface from each pose $\tilde{\mathbf{T}}_i \in \tilde{\mathbb{T}}$ such that $\tilde{\mathbb{C}} = \{\tilde{\mathbf{C}}_i, 1 \leq i \leq \tilde{N}\}$,
- $h(\tilde{N}, \tilde{\mathbb{T}}, \tilde{\mathbb{C}})$: the function of the total coverage check.

A surface is considered totally covered if $h(\tilde{N}, \tilde{\mathbb{T}}, \tilde{\mathbb{C}}) = \bigcup_{l=1}^{\tilde{N}} \tilde{\mathbb{C}}_l - S = 0$. When a multi-robot system is considered, one can consider collision avoidance through the constraint $h_{lm}(\tilde{\mathbb{T}})$. In this case, the distance between any two robots is greater than a predefined threshold δ defined with respect the dimension of the robot workspace, e.g. $h_{lm}(\tilde{\mathbb{T}}) = ||\tilde{\mathbb{T}}_l - \tilde{\mathbb{T}}_m|| - \delta$.

Finding the optimal number of robot base placements can be solved using a Binary Integer Programming (BIP). However, our problem is a NP-hard problem: an exact solution of the minimum number of robots is hard to compute using a Binary Integer Programming. Though, we propose a novel combination between three algorithms: Greedy, Genetic, and Simulated Annealing algorithms in III-C.

C. Hybrid optimization algorithm

In this Section, we present our hybrid optimization algorithm that merges the advantages of Greedy, Simulated Annealing and Genetic algorithms. Let's start by briefly presenting those algorithms :

1) The Greedy Algorithm: it is perceptively used to find the optimal number of variables required to respect an optimization function. It makes a local optimal choice at each iteration, hoping to find a global optimum at the end of the algorithm. This algorithm reduces complexity during an exhaustive search: it has $O(n)$ complexity instead of $O(n^K)$ where K is the optimal number of robots.

The algorithm intends to add a robot, at each iteration, until the surface is totally covered. The added robot is chosen randomly in such a way that it should maximize the coverage of the surface, while the previous robot poses are not updated. This lead to locally optimal solutions.

2) Simulated Annealing: Simulated Annealing avoids local optimum by using the temperature procedure [20]: the chance of accepting worse solutions reduces as long as the temperature decreases. This decision is made using an $Acceptance$ function that depends on the temperature and the percentage of the covered surface using the different robot poses. It also shows robustness and flexibility for global search methods, it can deal with highly non-linear problems and non-differentiable functions as well as functions with multiple local optima.

3) Genetic Algorithm: it is a method of search often applied to optimization problems or machine learning. Genetic Algorithms are part of evolutionary computing, they use an evolutionary analogy, "survival of the fittest" [21]. Instead of a single point generation at each iteration, Genetic Algorithm generates a population of points. After that, the best points are chosen as the optimal solution. It is more efficient than the traditional methods and provides a list of good solutions instead of a single solution. Thus, Genetic Algorithm increases the likelihood of finding the global optima.

4) Our hybrid algorithm: the structure of our hybrid algorithm is inspired from greedy algorithms in order to find the optimal number of robots. The hybrid Simulated Annealing and Genetic algorithms are used in order to benefit from their advantages, especially the speed of Simulated

Annealing with the variety of possible solutions of Genetic Algorithm in order to find the optimal pose solution.

Algorithm 1: Hybrid optimization algorithm

Input : The set of favorite robot poses $\mathbb{F}$, the surface to be covered S, the coverage threshold p, the maximum number of iterations N_{iter}, the initial temperature t_s, the ending temperature t_e, the maximum number of generations N_{gen}, the maximum number of populations N_{pop}

Output: Optimal number of robots $\tilde{N}$ and their poses $\tilde{\mathbb{T}}$

1: Set $U = \mathbb{F},\ V = \varnothing,\ Rpn = \varnothing,\ W = S,\ \tilde{\mathbb{T}} = \varnothing,\ \tilde{N} = 0,\ Pop = \varnothing$;
2: **while** $g(\tilde{N}, \tilde{\mathbb{T}}, \tilde{\mathbb{C}}) > 0$ **do**
3: $U = \mathbb{F}$;
4: Pop= a population initialized using U;
5: $ats = t_s$;
6: **for** $k \in \{1, \dots, N_{gen}\}$ **do**
7: $m_i \leftarrow$ Elements of Pop with the highest coverage, such as $m_i = \{m_1, m_2\}$ and $m_1 \neq m_2$;
8: $\rho_i \leftarrow$ Generatechildren(m_1, m_2), such as $\rho_i = \{\rho_1, \rho_2\}$;
9: $\sigma_i \leftarrow$ Elements of Pop with the lowest coverage, such as $\sigma_i = \{\sigma_1, \sigma_2\}$ and $\sigma_1 \neq \sigma_2$;
10: Let $f1$= Coverage(σ_i) and $f2$= Coverage(ρ_i);
11: **if** ($f2 \geq f1$) || ($f2 < f1$ & random(0,1)<= exp(($f1$-$f2$)/ats)) **then**
12: Replace σ_i by ρ_i in the initial population;
13: **end if**
14: $ats = ats - coolrate$;
15: **if** $ats < t_e$ **then**
16: break;
17: **end if**
18: **end for**
19: $Rpn \leftarrow$ The element of Pop having the maximum coverage;
20: $V \leftarrow$ The correspondent coverage set of Rpn;
21: $\tilde{N} = \tilde{N} + 1$;
22: **end while**
23: $\tilde{\mathbb{T}}$ = Rpn;
24: **return** $\tilde{N}$ and $\tilde{\mathbb{T}}$;

Algorithm 1 describes our optimization algorithm. The algorithm returns the number of robots $\tilde{N}$ with their optimal poses $\tilde{\mathbb{T}}$. The number of robots increases until the surface is totally covered (line 2). For each iteration of Greedy Algorithm, the optimisation of robot poses is accomplished using a combination between Simulated Annealing and Genetic algorithms. Genetic Algorithm is used to generate a population of robot poses. The distance between any two elements l and m, e.g. poses, of the population is respected during the population generation $h_{lm}(\tilde{\mathbb{T}}) \leq 0$. Then, for each generation, the two elements having the maximum coverage will be the parents of two children. If those two children cover less than the two population elements having the minimum coverage, then the *Acceptance* function of Simulated Annealing is used to decide if the latter population elements will be replaced by the children.

Hence, as a first step, a population Pop is initialized (line 4). Then, for each generation of Pop, two children are created from the two elements of Pop having the maximum coverage of the surface (line 8). The decision of replacing of the worst two elements of the population by the two generated children is made in line 11 using the *Acceptance* function of Simulated Annealing algorithm: exp(($f1$-$f2$)/ats)). Those three steps are repeated until the maximum number of generation N_{gen} is reached (line 6) or the ending temperature t_e is attended (line 15). The element of Pop covering the larger part of the surface is chosen: Rpn (line 19). If all the robots in Rpn covers 100% of the surface, the loop is broken. Otherwise, the number of robots increases until 100% of the surface is covered.

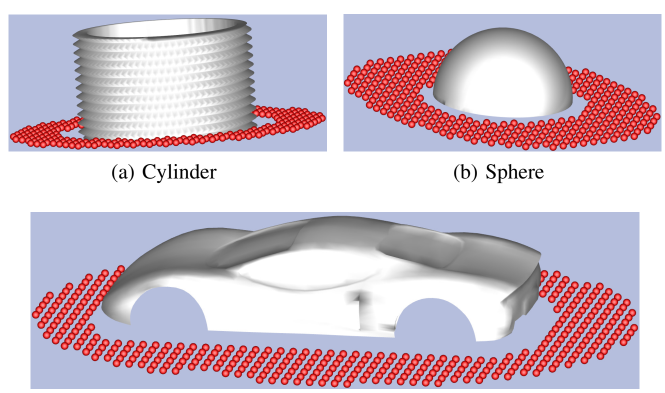
IV. Simulations and results

Our hybrid optimization algorithm is assessed on standard surfaces, e.g. a cylinder and a hemisphere, as well as on a complex surface, e.g. a car using KUKA Light Weight Robots (LWRs) each having 7 degrees of freedom. Furthermore, the hybrid optimization algorithm is compared to Greedy algorithm. The inputs of both optimization algorithms are the different surface models S and the favorite robot poses $\mathbb{F}$.

A. Pre-processing step

The discretization step is set to 0.1 during the pre-processing step. Each discretized point avoiding collision with the surface and allowing to reach more than $t = 15\%$ of the surface is considered as a favorite robot pose. We found 398 favorite robot poses for the cylinder, 375 for the sphere and 397 for the action car represented as the red spheres in Figure 3.

In this application, we consider a spherical workspace with a radius of 1 around the base of the robot. Hence, the reachable part of the surface from a given position is considered as the intersection between the spherical workspace and the surface to be covered.



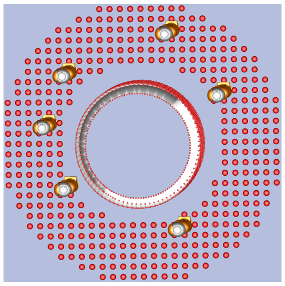
(a) Cylinder

(b) Sphere

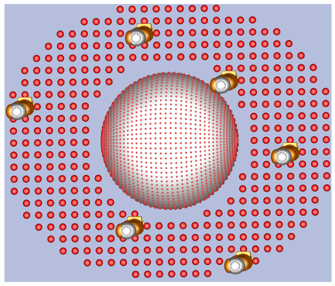
(c) Action car

Fig. 3. Surfaces to be covered by robots with the favorite base position (red spheres)

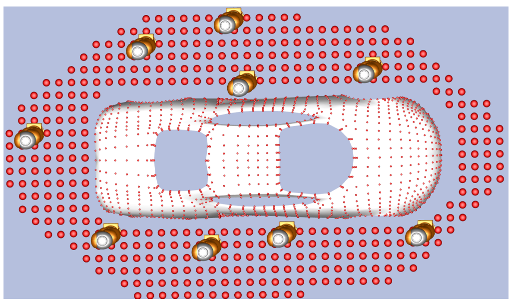
For sake of simplicity, we used the point cloud to compute the percentage of the covered surface. In global, we consider a convex feasible set, hence if the three points of a triangle are reachable, this triangle is supposed reachable. Those triangles are inside the reachable set of the surface. Each reachable point of the point cloud is represented by a red point on the surface as it is clear in Figures 4 and 5.



(a) Cylinder



(b) Hemisphere



(c) Action car

Fig. 4. Optimal base placements of different surfaces (greedy algorithm)

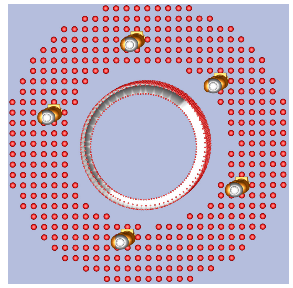
Greedy algorithm asserts that 6 robots are needed to totally cover the cylinder (Figure 4a), 6 poses to totally cover the hemisphere (Figure 4b), and 9 robots to cover the whole action car (Figure 4c).

However, by applying our hybrid optimization algorithm, we find that 5 robots are sufficient to totally cover the cylinder (Figure 5a), 2 robots to cover the whole hemisphere (Figure 5b), and 6 robots are enough to totally cover the action car as shown in Figure 5c.

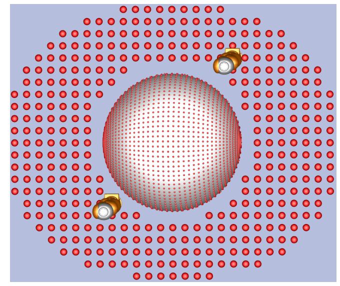
B. Comparison

As shown in Figures 4 and 5, the hybrid optimization algorithm gives less number of robots needed to cover the whole surface. It is clear that the distribution of robot poses around the surface is more homogeneous when we use our hybrid optimization algorithm. This point could be an advantage if the number of robots composing the multiple robotic system is lower than the optimal number of robots required to cover the surface. Furthermore, the more the distribution of robot poses is homogeneous around the surface the more the cycle time is reduced. That is because the distance between any two adjacent robot poses is almost equal: the time required to move the robots of the multi-robot system from one pose to the adjacent pose is optimized.

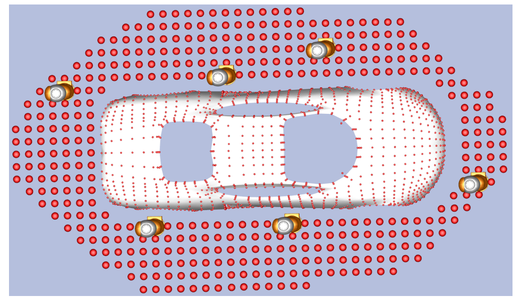
To assess the behavior of the proposed algorithm, we ran the code several times for each surface type. Figures 6, 7, and



(a) Cylinder



(b) Hemisphere



(c) Action car

Fig. 5. Optimal base placements of different surfaces (hybrid optimization)

8 compare both optimization algorithms by showing the average of the optimal number of robots (blue columns) and the margin of this number (red line) required to totally cover the cylinder, the sphere and the action car respectively. We can notice that for all surface types, the average of the optimal number of robots obtained using the proposed optimization algorithm is lower than the average of the optimal number of robots found using the greedy algorithm. Additionally, the margin of the optimal number of robots obtained using the hybrid optimization algorithm is tighter than the margin computed using the greedy algorithm. Furthermore, we can deduce that the proposed optimization algorithm provides more accurate solutions for all surface types.

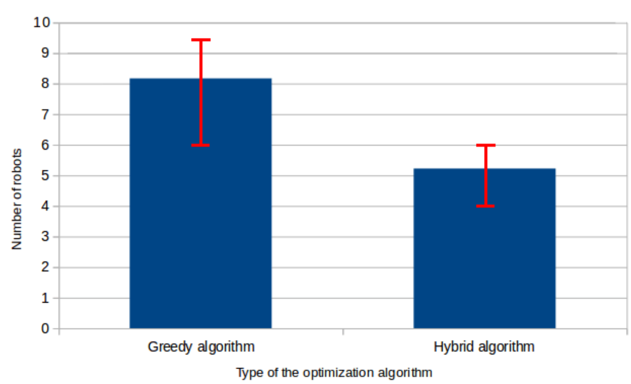


Fig. 6. The number of robots required to cover the whole cylinder using greedy algorithm and the proposed hybrid optimization algorithm.

V. Conclusion and Future Works

This article presents a new approach to distribute tasks between robots of multi-robot systems. It aims to find the

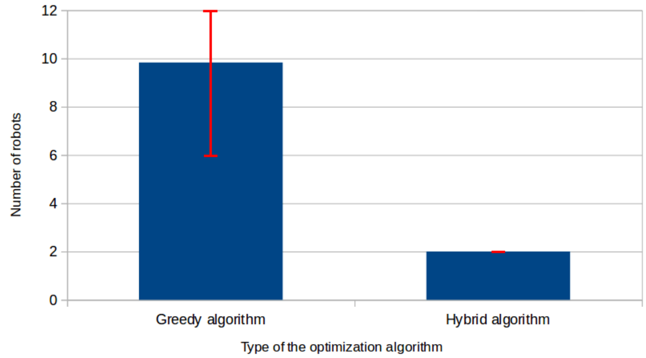


Fig. 7. The number of robots required to cover the whole sphere using greedy algorithm and the proposed hybrid optimization algorithm.

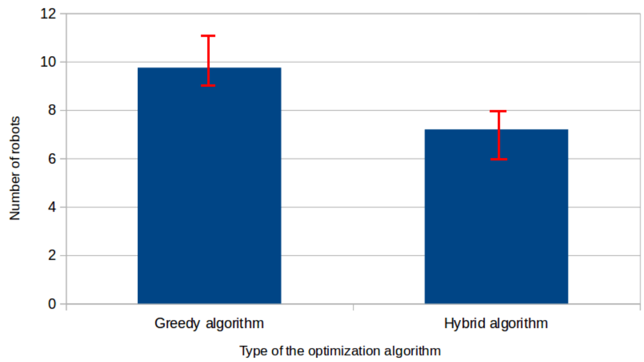


Fig. 8. The number of robots required to cover the whole action car using greedy algorithm and the proposed hybrid optimization algorithm.

optimal number of robots and their optimal poses required to cover large and complex surfaces. An hybrid optimization algorithm is proposed, it combines the three optimization algorithms: greedy, genetic, and simulated annealing. We proved that our hybrid optimization algorithm is more efficient than the Greedy one.

In future, the algorithm will be extended to tackle larger surfaces considering that the robotic manipulator can be placed on a crane. In addition, more realistic workspace will be considered taking into account different robot constraints such as kinematic constraints, singularity avoidance or maximal torque constraints. Furthermore, in case the number of robotic systems is limited, the robots will be redistributed based on new optimal path planner that considers the different systems constraints.

ACKNOWLEDGMENT

This work is funded by the French Government through the FUI Program (20^{th} call with the project AEROSTRIP).

REFERENCES

[1] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset, and A. A. Rizzi, “Uniform coverage of automotive surface patches,” The International Journal of Robotics Research, vol. 24, no. 11, pp. 883–898, 2005.

[2] T. R. Ren, N. M. Kwok, D. K. Liu, and S. D. Huang, “Path planning for a robotic arm sand-blasting system,” in 2008 International Conference on Information and Automation, June 2008, pp. 1067–1072.

[3] W. K. To, G. Paul, N. M. Kwok, and D. Liu, “An efficient trajectory planning approach for autonomous robots in complex bridge environments,” International Journal of Computer Aided Engineering and Technology, vol. 1, no. 2, pp. 185–208, 2009.

[4] T. Oksanen and A. Visala, “Coverage path planning algorithms for agricultural field machines,” Journal of Field Robotics, vol. 26, no. 8, pp. 651–668, 2009.

[5] E. U. Acar, H. Choset, Y. Zhang, and M. Schervish, “Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods,” The International journal of robotics research, vol. 22, no. 7-8, pp. 441–466, 2003.

[6] B. Englot and F. S. Hover, “Sampling-based coverage path planning for inspection of complex structures.” in Icaps, 2012.

[7] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, “Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics,” in Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015, pp. 6423–6430.

[8] M. Aly, A. Abbas, and S. M. Megahed, “Robot workspace estimation and base placement optimisation techniques for the conversion of conventional work cells into autonomous flexible manufacturing systems,” International Journal of Computer Integrated Manufacturing, vol. 23, no. 12, pp. 1133–1148, 2010.

[9] P. Sotiropoulos, N. Aspragathos, and F. Andritsos, “Optimum docking of an unmanned underwater vehicle for high dexterity manipulation,” IAENG International Journal of Computer Science, vol. 38, no. 1, pp. 48–56, 2011.

[10] G.-C. Vosniakos and E. Matsas, “Improving feasibility of robotic milling through robot placement optimisation,” Robotics and Computer-Integrated Manufacturing, vol. 26, no. 5, pp. 517 – 525, 2010.

[11] N. Vahrenkamp, T. Asfour, and R. Dillmann, “Robot placement based on reachability inversion,” in Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013, pp. 1970–1975.

[12] M. Hassan, D. Liu, S. Huang, and G. Dissanayake, “Task oriented area partitioning and allocation for optimal operation of multiple industrial robots in unstructured environments,” in Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on. IEEE, 2014, pp. 1184–1189.

[13] M. Hassan, D. Liu, and G. Paul, “Modeling and stochastic optimization of complete coverage under uncertainties in multi-robot base placements,” in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct 2016, pp. 2978–2984.

[14] M. Hassan and D. Liu, “Simultaneous area partitioning and allocation for complete coverage by multiple autonomous industrial robots,” Autonomous Robots, vol. 41, no. 8, pp. 1609–1628, Dec 2017.

[15] V. Chvatal, “A combinatorial theorem in plane geometry,” Journal of Combinatorial Theory, Series B, vol. 18, no. 1, pp. 39–41, 1975.

[16] S. Fisk, “A short proof of chvátal’s watchman theorem,” Journal of Combinatorial Theory, Series B, vol. 24, no. 3, p. 374, 1978.

[17] J. Kahn, M. Klawe, and D. Kleitman, “Traditional galleries require fewer watchmen,” SIAM Journal on Algebraic Discrete Methods, vol. 4, no. 2, pp. 194–206, 1983.

[18] B. Nilsson, “Guarding art galleries; methods for mobile guards,” Ph. D. thesis, Lund University, 1995.

[19] J. Zhao, R. Yoshida, S. ching Samson Cheung, and D. Haws, “Approximate techniques in solving optimal camera placement problems,” International Journal of Distributed Sensor Networks, vol. 9, no. 11, p. 241913, 2013.

[20] F. Busetti, “Simulated annealing overview,” World Wide Web, 2003.

[21] M. Srinivas and L. M. Patnaik, “Genetic algorithms: a survey,” Computer, vol. 27, no. 6, pp. 17–26, June 1994.