

“© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Performance Evaluation of an Evolutionary Multiobjective Optimization Based Area Partitioning and Allocation Approach

Mahdi Hassan and Dikai Liu

Abstract—An Area Partitioning and Allocation (APA) approach was presented in [1]. The approach focused on optimizing the coverage performance of Autonomous Industrial Robots (AIRs) using multiple conflicting objectives and Voronoi partitioning. However, questions related to the optimality, convergence, and consistency of the Pareto solutions were not studied in details. In this paper, Inverted Generational Distance (IGD) metric is used to verify the convergence of the Pareto front towards Pareto optimal front (PF*). The consistency in obtaining similar Pareto fronts for independent optimization runs is studied. The computational complexity of the approach with respect to the size of the coverage area and the number of AIRs is also discussed. Two application scenarios are used in this research.

I. INTRODUCTION

The Area Partitioning and Allocation (APA) approach presented in [1] and [2] is aimed at achieving complete coverage by simultaneously partitioning and allocating the overlapped areas amongst multiple Autonomous Industrial Robots (AIRs). As shown in Fig. 1, each AIR can cover an area of a surface, and there is an overlap of areas that the two AIRs can cover. *Overlapped areas* are the areas of a surface that more than one AIR can reach. On the other hand, *specific areas* of an AIR are the areas that are reachable by that particular AIR only. The APA approach considers complete coverage of surfaces that may be non-planar, complex in shape and separated (unconnected) from each other. To aim for optimal coverage, the APA approach optimizes four objectives. These objectives are in conflict with each other. That is, optimizing one of the objectives can only be done at a cost to another objective (or a subset of the objectives).

Authors are from the Centre for Autonomous Systems (CAS) at the University of Technology Sydney, 15 Broadway, Ultimo NSW 2007, Australia
Mahdi.Hassan@uts.edu.au

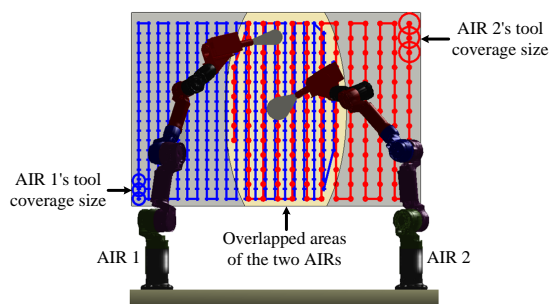


Fig. 1: Overlapped areas of two AIRs.

In real-world applications, many problems are multiobjective optimization problems (MOPs) in essence [3]. Thus, devising and improving multiobjective optimization algorithms are critical. Over the past two decades, the use of Evolutionary Multiobjective Optimization (EMO) is ever more popular since “Evolutionary Algorithms (EAs) proved to be capable of finding a good approximation to the Pareto optimal front (PF*) in Multi-objective Optimization Problems (MOPs) where there exist two or more conflicting objective functions” [4]. In this paper, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [5] which is one of the most popular EMOs is used to optimize the conflicting objectives in the APA approach.

Many approaches to complete coverage problems utilize gradient ascent based optimization to find a solution [6]–[8]. However, many real-world problems are non-convex in nature, and such approaches can only provide a locally optimum solution. In [9], a coverage control for multi-robot teams with heterogeneous sensing capabilities is presented. The work stated that even for the case of homogeneous sensing, knowing whether or not the achieved solution is locally optimum remains an open issue. The work in [8] notes that the camera coverage is a non-convex problem; thus gradient controllers can only achieve locally optimum solutions. In such approaches, often only one objective is considered or multiple objectives are combined together to form one objective. These approaches can be sufficient for some applications, e.g., for online camera coverage. However, for complex applications, e.g., those that account for multiple conflicting objectives or those that require coverage of objects with complex geometric shapes, finding a locally optimum solution may not be sufficient. This is the case for multiple AIRs operating on objects with complex geometric shapes.

The works in [1] and [2] presented the APA approach; however, questions related to the optimality, convergence, and consistency of the Pareto solutions were not studied in details. Thus, this paper investigates the following questions: (1) can the EMO-based APA approach provide Pareto fronts (explained in Section II-B) that are close to the PF*?, (2) for independent optimization runs, are the Pareto fronts consistently similar?, and (3) for an optimization run, how well does the Pareto front evolve over the generations (i.e., how good is the convergence)?. The computational complexity of the APA approach with respect to the size of the search space, the number of AIRs, and the size of the overlapped areas are also discussed.

II. BACKGROUND

A. The APA approach

The APA approach utilizes Voronoi partitioning to partition objects' surfaces [1], [2]. As shown in Fig. 2, for n AIRs, Voronoi partitioning creates n Voronoi cells on the overlapped areas. Each Voronoi cell is allocated to an AIR. The size and shape of each cell are dependent on the location of the Voronoi graph's seed points, $\mathbf{p}_1^s, \mathbf{p}_2^s, \dots, \mathbf{p}_n^s$ (as shown in Fig. 2). Note that the overlapped areas to be covered by the AIRs is fixed in size and Voronoi partitioning simply partitions this fixed area into n cells. That is, the total area covered by the Voronoi cells remains fixed in size. Thus, an advantage of utilizing Voronoi partitioning is that an additional objective function is not needed to maximize coverage. This partitioning strategy also prevents missed-coverage or coverage overlaps by the AIRs.

The APA approach has the advantage of being able to use the point cloud information that is generated from sensing the environment. The point cloud information is used to generate circular disks, called targets [1], [2], on the surfaces. Each AIR can have different sized targets to represent the surfaces (Fig. 2). This is because each AIR can have different capacity, e.g., different end-effector speed or tool coverage size. The targets in each Voronoi cell are closest to the seed point of the cell; therefore, $O_i^{al} = \{\mathbf{o} \in O_i \mid \|\mathbf{o} - \mathbf{p}_i^s\| \leq \|\mathbf{o} - \mathbf{p}_i^s\|, \forall i \in \{1, 2, \dots, n\} \setminus i\}$ where O_i^{al} is the set containing the allocated target of the i th AIR, O_i is the set of targets associated with the i th AIR and represent the entire overlapped areas, and \mathbf{p}_i^s is the seed point of the i th cell allocated to the i th AIR.

The design variables Z used in the multi-objective optimization algorithm (NSGA-II) are considered to be the coordinates of the seed points $\mathbf{p}_1^s, \mathbf{p}_2^s, \dots, \mathbf{p}_n^s$ (n seed point for n AIRs). To achieve optimal coverage, the aim of the optimization is to obtain a Voronoi cell for each AIR such that the following four objective functions are optimized [1]:

$$\min_Z F_1(Z) = \frac{1}{n} \sum_{i=1}^n (T_i(Z) - \bar{t})^2, \quad (1)$$

$$\min_Z F_2(Z) = \sum_{i=1}^n \sum_{j=1}^{N_i^o(Z)} \|\mathbf{c}_i^s - \mathbf{o}_{ij}\|, \quad (2)$$

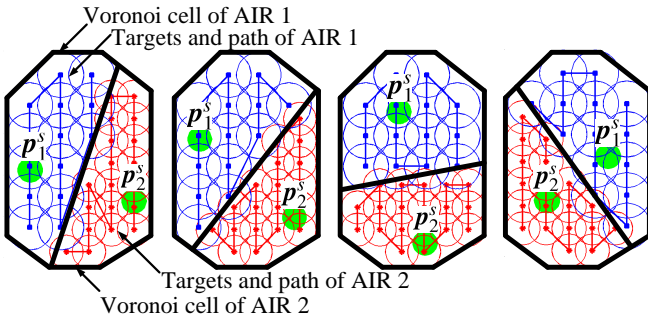


Fig. 2: Examples of Voronoi partitioning on the overlapped areas of two AIRs.

$$\min_Z F_3(Z) = \sum_{i=1}^n \sum_{j=1}^{N_i^o(Z)} \mathcal{T}^{Rmax}(\mathbf{q}_{ij}^f), \quad (3)$$

$$\max_Z F_4(Z) = \sum_{i=1}^n \sum_{j=1}^{N_i^o(Z)} W(\mathbf{q}_{ij}^f), \quad (4)$$

where $F_1(Z)$ returns the variance of the completion times of the AIRs; $F_2(Z)$ returns the sum of distances between the targets and the appropriate centroid, \mathbf{c}_i^s , of the specific areas; $F_3(Z)$ returns the sum of all the maximum torque ratios of AIRs' joints; and $F_4(Z)$ returns the sum of manipulability measures. In Eq. (1), $T_i(Z)$ returns the completion time of the i th AIR and \bar{t} is the average of the completion times of the n AIRs. In Eq. (2), $N_i^o(Z)$ returns the number of targets in the Voronoi cell allocated to the i th AIR and \mathbf{o}_{ij} is the j th target in the Voronoi cell of the i th AIR. In Eq. (3), $\mathcal{T}^{Rmax}(\mathbf{q}_{ij}^f)$ is the function that calculates the maximum torque ratio experienced by a joint of the i th AIR at an AIR pose \mathbf{q}_{ij}^f generated to reach the target \mathbf{o}_{ij} . In Eq. (4), $W(\mathbf{q}_{ij}^f)$ returns the manipulability measure of the AIR pose \mathbf{q}_{ij}^f . The reader is advised to refer to [1] for more details.

B. Multiobjective Optimization Problem

Multiobjective optimization problems (MOPs) consist of K objective functions, $F(Z) = [F_1(Z), F_2(Z), \dots, F_K(Z)]$, and M constraints. The optimizer aims to minimize the K objective functions while satisfying the M constraints (M can be zero). Objective functions, such as those in Eqs. (1) to (4), are computed over the design variables, Z (also called decision variables) to measure the performance of a solution.

Let Ω be the space in which the design variables in Z coexist. Similarly, let Ψ be the K -dimensional objective space in which objective functions in $F(Z)$ coexist. Therefore, a point in Ω maps to a point in Ψ . Let $\mathbf{x} \in \Omega$ and $\mathbf{y} \in \Omega$ be two solution vectors (values of the design variables in Z). The solution \mathbf{x} *dominates* the solution \mathbf{y} (i.e., $\mathbf{x} \succ \mathbf{y}$) if an only if for all of the objective functions, \mathbf{x} doesn't yield a value worse than \mathbf{y} in Ψ space and it strictly yields a better value for at least one of the objectives [4].

In MOPs, the objective functions are in conflict with each other. That is, optimizing one of the objectives can only be done at a cost to another objective (or a subset of the objectives). Thus, instead of a single optimal solution, the aim is to get the Pareto optimal set, P^* . The solution vectors in P^* are non-dominated by any other solution vectors in Ω . Formally [4]:

$$P^* := \{\mathbf{x} \in \Omega \mid \nexists \mathbf{y} \in \Omega \text{ such that } \mathbf{y} \succ \mathbf{x}\}. \quad (5)$$

The corresponding Pareto optimal front (PF*) in Ψ space is:

$$PF^* := \{F(\mathbf{x}) \in \Psi, \forall \mathbf{x} \in P^*\}. \quad (6)$$

For more details of MOPs refer to [10]. Note that in many real-life applications, the PF* is not known [4]. Thus, a reference set, PF' is used to approximate the PF* [4]. A strategy for obtaining PF' is explained in Section III-B.

C. Performance Metric

Evolutionary Multiobjective Optimization (EMO) algorithms return a set of solution vectors, P , with the corresponding Pareto front PF . To measure how far PF is from PF^* , the Inverted Generational Distance (IGD) is used in this paper. IGD has become increasingly popular in recent years and it is frequently used to evaluate the performance of EMOs [11].

The IGD metric calculates the average distance from each point in PF^* (Pareto optimal front) to the nearest point in PF (Pareto front from EMO algorithm). If the number of points in PF' (reference set approximating PF^*) is large enough to represent PF^* , the IGD metric could measure both convergence and diversity of PF [3]. The IGD value can be calculated as [3]:

$$IGD(PF, PF^*) = \frac{1}{|PF^*|} \sqrt{\sum_{a \in PF^*} \left(\min_{b \in PF} \|a - b\|^2 \right)} \quad (7)$$

where $\|a - b\|$ is the Euclidean distance between $a \in PF^*$ and $b \in PF$. Note that PF^* and PF are normalized using the maximum and minimum values in PF^* . The smaller the value of IGD, the closer PF is to PF^* and the lesser are the regions of PF^* missed by PF . If PF^* is not known then $PF^* = PF'$.

III. PERFORMANCE EVALUATION OF EMO-BASED APA APPROACH

The aim here is to validate the EMO-Based APA approach with respect to optimality, convergence, and consistency of the Pareto front. Metaheuristic evolutionary optimization algorithms are criticized for the randomness in some of their processes and the lack of guarantee in obtaining the global optimal. Using detailed study of two test scenarios and a large number of optimization runs, it is shown in this section that near-optimal solutions could be obtained and that the Pareto front is consistently similar despite the randomness in NSGA-II (chosen EMO algorithm). The convergence of the Pareto front over generations is also shown. Furthermore, the computational complexity is discussed with respect to the size of the search space, the number of AIRs, and the number of targets representing the overlapped areas.

A. Test Scenarios

1) Scenario 1 - Three AIRs with Different Capabilities:

In this scenario, three AIRs grit-blast or spray paint the flat surface shown in Fig. 3a. The size of the targets associated with each AIR is different since the capabilities of the AIRs are different. Table I shows the end-effector speed (v), the target size (r) and the distance between adjacent targets along a path (d) for each AIR. The overlapped areas of the three

TABLE I: Properties of the three AIRs.

	AIR 1	AIR 2	AIR 3
v (m/s)	0.2	0.15	0.1
r (m)	0.03	0.04	0.05
d (m)	0.0402	0.0536	0.067

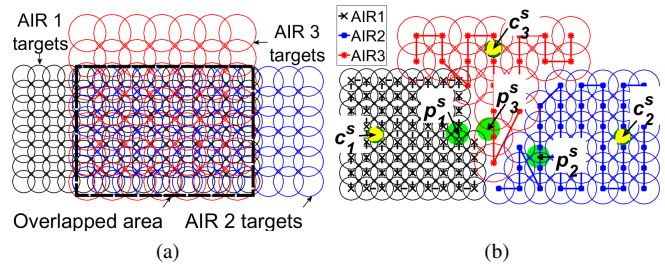


Fig. 3: Overlapped areas (a) and a sample solution (b).

AIRs are also shown in Fig. 3a. For the sake of a better graphical representation of the results (e.g., the Pareto front) only the first two objective functions (Eqs. (1) and (2)) are considered for this scenario. As an example to illustrate the output from the APA approach, a solution [1] selected from the Pareto front of one of the optimization runs is shown in Fig. 3b. For this solution, the AIRs complete the task almost simultaneously with a completion time of 13.5 s (AIR 1), 13.6 s (AIR 2) and 13.4 s (AIR 3). As shown in Fig. 3b, the allocated portion of the overlapped areas for each AIR is close to the corresponding specific areas of the AIR as would be expected from the second objective function (Eq. (2)). In the figure, the green filled circles (p_1^s, p_2^s, p_3^s) are the location of the seed points and the yellow filled circles (c_1^s, c_2^s, c_3^s) are the centroids of the specific areas.

2) Scenario 2 - Two AIRs Operating on Three Separated Objects: As shown in Fig. 4, two identical AIRs operate on objects that are separated from each other, one of which have a complex geometric shape. In this scenario, All four objective functions (Eqs. (1) to (4)) are considered. For both AIRs, the end-effector speeds is $v = 0.04$ m/s; target size is $r = 0.04$ m; and the distance between adjacent targets along a path is $d = 0.0563$ m. For a graphical representation of a sample of solutions refer to [1].

B. Obtaining the Pareto Optimal Front (PF^*)

At first, the PF^* for the test scenarios need to be obtained. In many real-life applications, the PF^* is not know. Thus, a

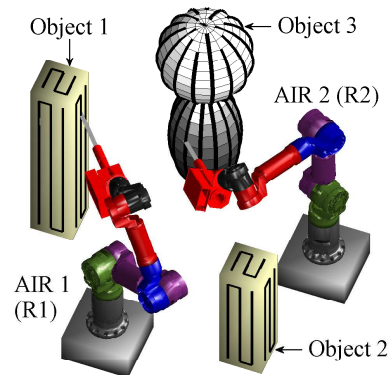


Fig. 4: Two AIRs operating on three separated objects.

reference set, PF' is used to approximate PF^* . A strategy for obtaining PF' is to run the EMO algorithm with a large population and a large generation number [11]. The EMO algorithm can then be run several times and the solutions obtained from each optimization run are merged together to give a single non-dominated solution set which is the reference set [11].

To obtain the reference set, henceforth referred to as the PF^* , both the population size and generation number of NSGA-II is set to 10,000. MATLAB global optimization toolbox, and specifically the function 'gamultiobj' which is based on NSGA-II, is used as the example EMO algorithm. The optimization for each scenario is repeated 30 times. The solutions from independent runs are then merged together and the non-dominated solutions are obtained to represent the PF^* . Note that the optimization terminates before reaching the maximum generation number due to the average change in the spread of the Pareto front being less than the default tolerance ($1e-4$) by 'gamultiobj' function.

Figure 5 shows the PF^* for scenario 1. It is clear that the objective functions are in conflict with each other. For scenario 2, since all four objectives are used (i.e., the objective space is 4-dimensional), then the Pareto front is not plotted. However, to show the trade-off between the objectives, the optimization is run for each pair-wise combination of the objective functions. From Fig. 6, it is again clear that all pairs of objectives are in conflict with each other. The PF^* for the second scenario contains 17,275 points.

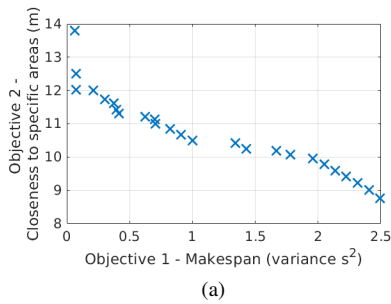


Fig. 5: PF^* for scenario 1.

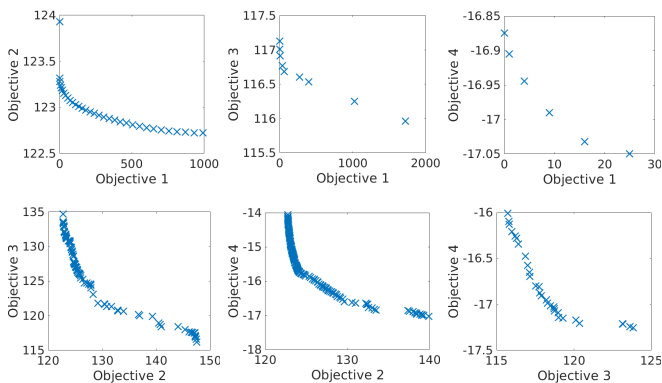


Fig. 6: Trade-off for all pairwise combinations of the objectives for scenario 2.

C. Convergence Towards the PF^*

Optimization is repeated for 50 different population sizes (from 15 to 750 in steps of 15) for both scenarios. Then, the distance of the Pareto front from each optimization run is compared to the PF^* using the IGD value (Figure 7). Note that for each population size, optimization is repeated 30 times and the average of the IGD values is taken.

The results show that the IGD value is very small and tends to approach zero as population size is increased, which means that the Pareto front could potentially converge to PF^* . However, after certain population size, the rate of decrease in IGD value becomes very small, and a much larger population size would be needed to make a significant difference to the IGD value which causes substantial increase in the computation time. For most real-life applications, a near-optimal result is acceptable in favor of reduced computation time. Figure 8 shows the total number of function evaluations (number of individuals evaluated in NSGA-II) for the optimization runs. The number of function evaluations tends to increase linearly. This is because the growth in function evaluations is mainly due to the increase in the population size and not the generation number since the NSGA-II terminates at approximately 115 generations (on average) due to the average change in the spread of the Pareto front being less than the default tolerance.

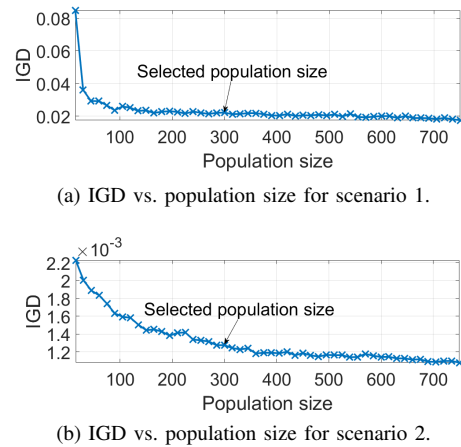


Fig. 7: IGD vs. population size for both scenarios.

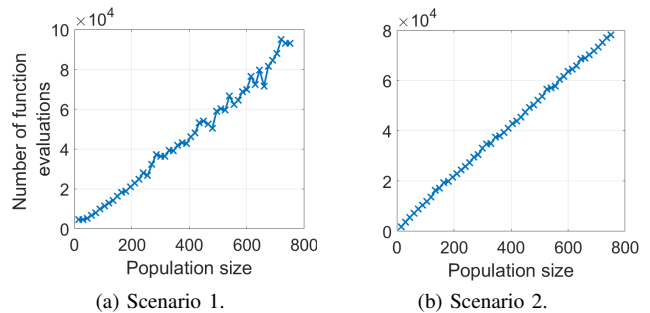


Fig. 8: Number of function evaluations for both scenarios.

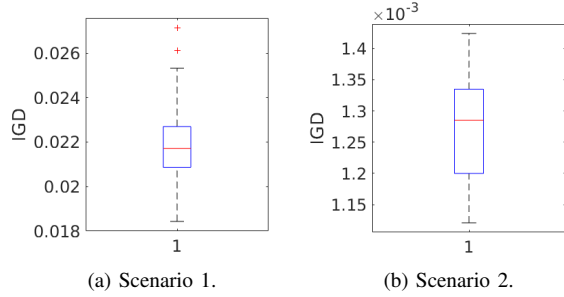


Fig. 9: Boxplot of the IGD values of the 30 optimization runs for both scenarios.

D. Consistency of the Solutions

The aim here is to check that the Pareto fronts are consistently similar for independent optimization runs. First, a population size needs to be determined. From Fig. 7, setting the population size to 300 seems reasonable since the value of IGD doesn't continue to significantly reduce after around this population size. Optimization is repeated 30 times for this population size and the IGD values for each optimization run is stored. Figure 9 shows the boxplot of the IGD values for the 30 optimization runs. For scenario 1, the mean and standard deviation of the IGD values is 0.022 and 0.002, respectively. For scenario 2, the mean and standard deviation of the IGD values is $4.7e-04$ and $1.9e-05$, respectively. From the boxplots and the small values of the standard deviations, it is clear that the solutions obtained are consistent.

The Pareto front of an optimization run that has the same IGD value as the above mean IGD value for scenario 1 is selected. The aim is to graphically show (Fig. 10) the closeness of the Pareto front to the PF* for the population size of 300. It can be seen that many of the solutions in the Pareto front are optimal or near-optimal and the solutions are well spread.

E. Convergence of the Pareto Front for Independent Optimization Runs

The aim here is to show that the Pareto front does converge over the generations. To provide more accurate results, the optimization is repeated 30 times for both scenarios and the IGD values of the Pareto fronts at each generation for all

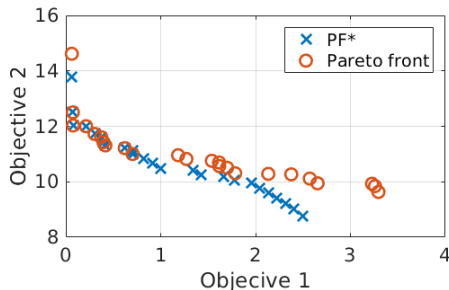


Fig. 10: PF* vs. Pareto front of an optimization run.

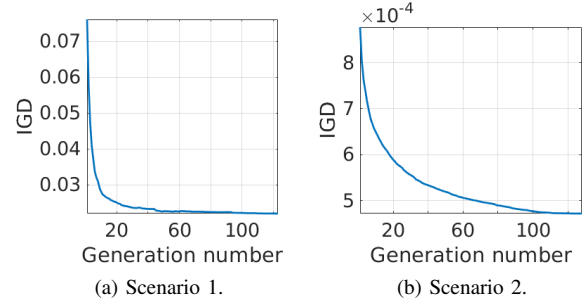


Fig. 11: Convergence of the Pareto fronts for both scenarios.

optimization runs are stored. The average of the IGD values at each generation is then computed and the plots shown in Fig. 11 is constructed to validate the convergence over the generations for both scenarios. Note that at each generation, the Pareto front is made to contain all of the non-dominated solutions up to that particular generation. This is different to how 'gamultiobj' function of MATLAB calculates the Pareto front at each generation since it only keeps a fraction of the non-dominated solutions from the higher fronts based on the option 'ParetoFraction' $\in [0, 1]$ (default 0.35). Hence, the IGD value in Fig. 11 converges to a slightly lower value than that shown in Fig. 7 since there are more points in the Pareto front.

F. Computational Complexity

The computational complexity of the APA approach depends on the size of the search space, the number of AIRs, and the number of targets representing the overlapped areas.

The fitness function constructed for the NSGA-II calculates the values of the objectives functions for inputted locations of the seed points (design variables). Figure 8 showed that the number of function evaluations is linear with respect to the population size. However, the computational complexity of the fitness function needs to be discussed in order to examine the computation complexity of the overall approach.

The fitness function loops through the overlapped targets of the n AIRs and allocates the targets to the AIRs based on the closeness of the targets to the seed points. That is, $O_i^{al} = \{\mathbf{o} \in O_i \mid \|\mathbf{o} - \mathbf{p}_i^s\| \leq \|\mathbf{o} - \mathbf{p}_i^s\|, \forall i \in \{1, 2, \dots, n\} \setminus i\}$ as mentioned in Section II-A. For details of the fitness function, refer to [1].

If the AIRs are identical (i.e., have the same capabilities), then the computation complexity of the fitness function is $\mathcal{O}((n+3)m)$ where n is the number of AIRs and m is the total number of targets representing the overlapped areas. This is because one set of m equally sized targets can represent the overlapped areas of all AIRs, and the distance of each target within this set to the n seed points is checked for allocation to the Voronoi cell with the closest seed point. The number 3 in $\mathcal{O}((n+3)m)$ is related to 3 operations for concatenating each target and its corresponding manipulability measure and torque ratio to 3 separate sets for later use. Note that

the manipulability measure and torque for each target are calculated only once prior to the optimization process, e.g., based on the lookup table explained in [12]. Hence, for n number of identical AIRs, the computation complexity grows linearly with respect to the number of targets m in the overlapped areas. Similarly, for a fixed number of targets m , computation complexity is linear in the number of AIRs n .

If the AIRs are not identical, then the computational complexity becomes $\mathcal{O}(\sum_{i=1}^n (n+3)m_i)$ where m_i is the number of targets associated with the i th AIR and represent the overlapped areas. This is because each AIR is associated with a set of targets that can have different size to other AIRs' targets. Thus, for n AIRs, n sets of targets represent the overlapped areas. Similar to the case of identical AIRs, for n non-identical AIRs, the computation complexity grows linearly with respect to the number of targets ($m = m_1 + m_2 + \dots + m_n$) in the overlapped areas. However, for a fixed number of targets m and considering the worst case where no two AIRs are identical, the computation complexity grows quadratically with respect to the number of AIRs, n .

The above analysis of the computation complexity considers the worst case. The computation complexity can be reduced through strategies such as using appropriate data structures (e.g., k-d tree and Octree). For instance, checking the distance of octants in an Octree to the seed points instead of checking every single target. Note that the computation complexity presented does not consider the special condition where different pairs of AIRs (for $n > 2$) can have different overlapped areas.

The search space is continuous; however, bounded by a bounding rectangle (or a bounding box for 3D coverage) that occupies the overlapped areas. Thus, the seed points of the Voronoi cells are constrained to be within this bounding rectangle/box. Note that the design variables are the position coordinates of the seed points $p_1^s, p_2^s, \dots, p_n^s$. Thus, for n AIRs, there are $3n$ design variables. For a planar surface, one of the coordinates of the seed points will be a constant.

The search space can be discretized to improve computational efficiency. Suppose that the search space is uniformly discretized into G grids. The seed points can then be constrained to be only at the center of the grids. Thus, there are $\frac{G!}{(G-n)!}$ permutations for the search space where n is the number of AIRs. As an example, let the overlapped area of the planar surface presented in the first scenario be discretized based on the smallest target size (i.e., AIR 1) which gives 96 grids (12×8). For the worst case where all AIRs are non-identical, there are 857280 permutations. Thus, three-dimensional coverage space, larger overlapped areas, or greater number of AIRs increase the size of the search space. This may indicate that the optimizer would require a larger population size for a wider exploration of the search space.

IV. CONCLUSION

In this work, an evolutionary multiobjective optimization (EMO) based area partitioning and allocation (APA) approach was validated with respect to optimality, convergence,

and consistency of the Pareto front. The Non-dominated Sorting Genetic Algorithm II (NSGA-II) which is one of the most popular EMOs was used for optimizing the conflicting objectives in the APA. Using detailed study of two application scenarios and a large number of optimization runs, it was shown that the Pareto front converges to the Pareto optimal front as the population size in NSGA-II is increased. It was also illustrated that for a chosen population size, the Pareto front is consistently similar despite the randomness in NSGA-II. The convergence of the Pareto front over generations was also presented. To study these aspects of the APA approach, the Inverted Generational Distance (IGD) metric was used. Furthermore, the computational complexity was provided with respect to the size of the search space, the number of AIRs, and the number of targets representing the overlapped areas.

ACKNOWLEDGMENT

This work is supported by the Centre for Autonomous Systems (CAS) at the University of Technology, Sydney.

REFERENCES

- [1] 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.
- [2] 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 *13th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2014, pp. 1184–1189.
- [3] Y. Yuan, Y.-S. Ong, L. Feng, A. Qin, A. Gupta, B. Da, Q. Zhang, K. C. Tan, Y. Jin, and H. Ishibuchi, "Evolutionary multitasking for multi-objective continuous optimization: Benchmark problems, performance metrics and baseline results," *preprint arXiv:1706.02766*, 2017.
- [4] N. Riquelme, C. V. Lcken, and B. Baran, "Performance metrics in multi-objective optimization," in *2015 Latin American Computing Conference (CLEI)*, Oct 2015, pp. 1–11.
- [5] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Parallel Problem Solving from Nature PPSN VI*, ser. Lecture Notes in Computer Science, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. Schwefel, Eds. Springer Berlin Heidelberg, 2000, vol. 1917, pp. 849–858.
- [6] A. Adaldo, S. S. Mansouri, C. Kanellakis, D. V. Dimarogonas, K. H. Johansson, and G. Nikolakopoulos, "Cooperative coverage for surveillance of 3D structures," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 1838–1845.
- [7] X. Zhang, X. Chen, X. Liang, and Y. Fang, "Distributed coverage optimization for deployment of directional sensor networks," in *IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 246–251.
- [8] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, Sept 2011.
- [9] M. Santos, Y. Diaz-Mercado, and M. Egerstedt, "Coverage control for multirobot teams with heterogeneous sensing capabilities," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 919–925, April 2018.
- [10] K. Deb, *Multi-objective Optimization*. Boston, MA: Springer US, 2014, pp. 403–449.
- [11] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems," in *2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*, Dec 2014, pp. 170–177.
- [12] M. Hassan, D. Liu, G. Paul, and S. Huang, "An approach to base placement for effective collaboration of multiple autonomous industrial robots," in *International Conference on Robotics and Automation*, May 2015, pp. 3286–3291.