2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Is MO-CMA-ES Superior to NSGA-II for the Evolution of Multi-objective Neuro-controllers?

Amiram Moshaiov and Omer Abramovich[1]

Abstract— **In the last decade evolutionary multi-objective optimizers have been employed in studies concerning evolutionary robotics. In particular, the majority of such studies involve the evolution of neuro-controllers using either a genetic algorithm approach or an evolution strategies approach. Given the fundamental difference between these types of search mechanisms, a valid question is which kind of multi-objective optimizer is better for such applications. This question, which is dealt with here, is raised in view of the permutation problem that exists in evolutionary neural-networks.**

Two well-known Multi-objective Evolutionary Algorithms are used in the current comparison, namely MO-CMA-ES and NSGA-II. A multi-objective navigation problem is used for the testing, which is known to suffer from a local Pareto problem. For the employed simulation case MO-CMA-ES is better at finding a large sub-set of the approximated Pareto-optimal neuro-controllers, whereas NSGA-II is better at finding a complementary sub-set of the optimal controllers. This suggests that, if this phenomenon persists over a large range of case studies, then future studies should consider some modifications to such algorithms for the multi-objective evolution of neuro-controllers.

I. INTRODUCTION

EMPLOYING Multi-Objective Evolutionary Algorithms (MOEAs) to support design under contradicting objectives is becoming widespread [1]. As described in the background section, below, this trend is also evident in the field of Evolutionary Robotics (ER), which involves the use of Evolutionary Computations (EC) to support robot design.

A large part of ER studies deals with the evolution of Neuro-Controllers (NCs) [2, 3]. Such a problem involves complicated search spaces, which may include not just the connection weights but also the network morphology, and the activation parameters that define the NCs. Given the peculiarities of such search spaces, there is a need to investigate the computational aspects of the use of MOEAs for the evolution of NCs in ER applications. This need is apparent when considering the state-of-the-art of the research on Evolutionary Neural Networks (ENN) and on ER, as well as the existing knowledge on MOEAs, as described in the following background section. At present, existing MOEAs are compared using test functions, which are not necessarily indicative of the peculiar search difficulties that may inherently exist in the evolution of NCs.

The current study focuses on a multi-objective navigation problem. The performance of two MOEAs is compared with respect to their ability to find approximated Pareto-optimal NCs for the problem. The MOEAs used here represent two distinct types of EC. The first algorithm is NSGA-II, [4], which uses a Genetic Algorithm (GA) approach, whereas the second is MO-CMA-ES, [5], which employs a self-adaptive Evolution Strategies (ES) approach.

The current study is restricted to Feed-Forward Networks (FFNs) with fixed morphology. It follows the investigations in [6, 7], which suggest that the explored navigation problem suffers from multiple local Pareto fronts. This phenomenon makes the search difficult. Given the lack of analytical solution, and the stochastic nature of the search, the comparison is based on the relative statistical performance as obtained by the employed algorithms.

The rest of this paper is organized as follows. Section II deals with the relevant background for the understanding of the motivation and rational behind this study. Section III describes the methodology including the details of the tested case, and some information on the algorithms and search parameters which are used here. Section IV provides the results and finally section V outlines the conclusions from this study.

II. BACKGROUND

A. Evolution Strategies vs. Genetic Algorithms

Both GA and ES are generic search methods, which are bio-inspired by natural evolution [8]. The main features of such algorithms are the use of a population of candidate solutions as well as mutation, crossover and selection operators, which are devised to drive the evolutionary process towards improved solutions. In GA crossover is the major reproduction mechanism when compared with mutation. ES differs from GA by having mutation as the main reproduction factor, and often the only one. Modern ES variants are based on the capability to self-adapt the internal strategy parameters. Most notable is CMA-ES, [9, 10], which self-adapts the mutation co-variance matrix. Adapting the mutation steps makes it possible to use only (or mainly) mutation for both exploration and exploitation elements of the search. In [11] Kita compares the self-adaptive mechanisms of ES and Real-Coded GA (RCGA) for optimization in continuous search spaces. It has been argued there that crossover in RCGA generates offspring adaptively according to the distribution of parents without any adaptive parameters. According to [11], both adaptation of mutation in ES and adaption by crossover in RCGA work well in function optimization.

[1] The authors are with the School of Mechanical Engineering, The Iby and Aladar Fleischman Faculty of Engineering, Tel-Aviv University, Israel, email: moshaiov@eng.tau.ac.il

Recent comparisons of ES variants with other evolutionary algorithms on black-box optimization benchmarks have shown that the former are better than RCGA (e.g., [12, 13]). However, as argued in [14], while performing better in such benchmarks studies, an algorithm cannot be declared better for a real-world problem without further analysis. With this respect, it should be noted, that test problems, which are commonly used in benchmark studies, usually include no situations with a difference between genotypes and phenotypes. This is in contrast to neuro-evolution where the codes substantially differ from the phenotypes, i.e. from the neural-networks.

In spite of the potential superiority of ES variants, as indicated by studies such as in [12, 13], the use of GAs is much more common when compared with that of ES. Given the above arguments, comparative studies on actual applications, such as the one which is carried out here, are needed.

B. *Evolutionary Multi-Objective Optimization*

Tradeoffs play a major role in engineering design. To understand tradeoffs, optimization with a vector-valued quality criterion can be used. This is apparent in the literature on Multi-Objective Optimization (MOO) (e.g., [15, 16]). Most past studies on MOO involved non-Pareto approaches, but this has changed dramatically over a decade ago, with the appearance of MOEAs. The goal of such modern algorithms is to provide a good approximation to the set of Pareto-optimal solutions, i.e., those solutions that cannot be improved in one objective without getting worse in another. In such algorithms, selection is based on comparing solutions in objective space by the relation of dominance. A Pareto-optimal set constitutes the non-dominated set of solutions, and the Pareto front is the associated set of performance vectors.

Nowadays, MOEAs are well accepted as a general computational paradigm for MOO. MOEAs have been proven successful in many application areas and have become the method of choice for MOO. MOEAs have successfully been applied to the design of complex systems and in-particular for neural networks (e.g., [17]). The interested reader is referred to a recent review, [1], which describes the current issues concerning the development of MOEAs. In the following some ES-based MOEAs are described to provide a background on possible alternatives to the MO-CMA-ES algorithm, which has been selected for the current study.

Pareto Archived Evolution Strategy (PAES), [18], is one of the earliest well-known classical MOEAs. As an ES algorithm, its main search operator is mutation. The original PAES is a (1+1)-ES, meaning that during each iteration one parent is used to create one mutant, and that their union is used in the selection. Yet the concept is not restricted to such population and mutant sizes. Given the general ES class of evolutionary algorithms, the original PAES has set the stage for a class of ES-based MOEAs, with potentially many alternative algorithms.

The incorporation of the idea of self-adaptive co-variance matrix into multi-objective optimization has been suggested and studied in [5], where MO-CMA-ES has been presented. The adaptive grid for archiving, which was used in PAES has been recently merged with the CMA-ES approach, in [19]. Various ideas, which were originated in non-ES studies, have been adapted into ES-based MOEAs. For example, in [20], the ES-based SMS-EMOA has been suggested, which incorporates selection by hyper-volume domination. Sarker et al., [21], developed an ES-based MOEA where the number of mutants varies from generation to generation based on the number of solutions in the Pareto-archive. A mixture of ideas from ε- MOEA, [22], and PAES, [18], have been attempted recently by Zhao et al., [23], and by Moshaiov and Elias [24]. In the later study the proposed ε-PAES is shown to be superior to ε-MOEA, which is a GA-based MOEA.

The current study deals with the question of what type of MOEA should be used for multi-objective search and optimization of NCs. As evident from reviews such as in [1], there is a wide range of alternative "off-the-shelf" algorithms to potentially be used. It is likely that such algorithms might also be tailored to the evolution of NCs. At present, this study is restricted to the comparison of existing MOEAs. In addition, currently we are not interested in an extensive study that compares a large number of MOEAs for the application. Rather, we are interested in comparing a well-known GA-based MOEA with a well-known ES-based MOEA, to see the effect of the different search mechanisms. For this purpose we use NSGA-II, [4], and MO-CMA-ES, [5].

C. *Evolutionary Neural Networks*

Studies on ENNs deal with the use of EC to evolve Neural Networks (NNs). Reviews on such studies, e.g. [25-27], reveal the special computational considerations, which are required for the evolution of NNs. One well-known peculiarity is the existence of multiple genotypes for the same phenotype, which results in numerical difficulties during the convergence phase of the search. This phenomenon has been referred to as the permutation problem. According to [25], one possible solution to the permutation problem is to use a search method which is primarily based on mutation rather than re-combination. Namely, ES–based algorithms should be advantageous as compared with GA-based algorithms, unless the later are modified to overcome the permutation problem.

A large part of the existing ENN studies concerns single-objective search, and as such does not provide an answer to the research question which is raised here. In fact, to the best of our knowledge, no previous ENN study has dealt with research questions such as what kind of MOEA is better for the multi-objective evolution of NCs. In view of the permutation problem and the suggestion of [25], we aim to investigate the performance differences between typical ES-based and GA-based MOEAs, when employed to evolve NCs.

Nowadays, much of the focus of ENN studies is on TWEAN, namely Topology and Weight Evolutionary Neural Networks. Most notable algorithms are NEAT, [28] and CMA-TWEAN, [29]. The current study, however, concentrates on problems with fixed network structure.

To substantiate our selection of MO-CMA-ES, we note the results of a recent comparative study, in [30], on various ENN algorithms using the pendulum benchmark problem in the context of reinforcement learning. In [30] CMA-NeuroES is proposed and evaluated on five different variants of the common pole balancing problem. The comparison with other ENN algorithms shows that the proposed CMA-ES based algorithm has the overall best performance.

D. Evolutionary Robotics

Born in the 1990s, ER might be perceived as a computational methodology to automatically develop robot controllers as well as other aspects of robots including the mechanics, sensors and actuators. A comprehensive background on ER in the 90's can be found in [2], whereas more current surveys, analyses, and trends can be found in [3, 31]. The evolutionary process in ER may be viewed as reinforcement learning. In such a type of learning an agent learns from interactions with the environment. A major part of ER studies deals with finding useful controllers, and in-particular NCs. ER studies with NCs generally employ ENN methods.

In ER, a major role of the designers is the choosing of the appropriate fitness function, which is crucial for a successful evolutionary process [31]. While most ER studies deals with problems which are defined either as a single objective one or as an aggregated multi-objective one, this trend is changing. A Pareto-approach using MOEAs is finding its way into the ER research community. There are two main reasons for the use of MOEAs in ER. First, as in most engineering design applications, it can produce a set of optimal solutions with respect to contradicting design objectives. For example, in [32], the robotic navigation problem is defined as a MOO problem and solved using a modification of NSGA-II to obtain a Pareto-optimal set of NCs. Second, MOEAs can be used to overcome numerical problems in single objective ER problems. For example, [33] and [34] used MOEAs to overcome the bootstrap problem, which is common to the evolution of complex behaviors.

In a recent study on the multi-objective evolution of NCs for robot navigation, [6, 7], a local Pareto problem has been reported. In view of this problem, and the permutation problem of ENN, existing MOEAs should be carefully examined to check their search and convergence behaviors in multi-objective evolution of NCs.

ES was used in early ER studies during the mid 90's [35]. According to [35], when compared with the use of GA, the ES approach was shown to be superior by an order of magnitude. In [36] an ES approach is used for on-line adaptation of robot controllers. In such applications the quality of the optimizer is critical. It is stated there that ES was chosen for its very good reputation, but no comparisons with other optimizers are given.

In most ER studies that use MOEAs, NSGA-II or some variant of it are used. This situation appears to be a result of the availability of NSGA-II code and its reputation as a good optimizer for problems with a few objectives. In contrast, ES-based MOEAs, while promising, are relatively less known to the community of ER researchers. Given the potential of ES-based MOEAs for ER applications, and in particular for evolving NCs, it seems important to perform a comparative study as done here.

III. METHODOLOGY

This comparative study is based on a simulation, which consists of a simulated environment, a simulated robot, and simulated Neuro-Controllers (NCs). These are described in the following subsections A, B, and C, respectively. The simulation involves a bi-objective navigation problem, which is described in subsection D. Additional aspects of the evaluation of the NCs are given in subsection E. Details on the use of MO-CMA-ES and NSGA-II are given in subsection F.

A. The Simulated Arena

The simulated arena follows the one used in [6, 7]. It contains rooms and corridors as shown in figure 1. Also shown in the figure are food-targets, marked by red crosses, which are located in most areas of the arena excluding the large room on the left side.

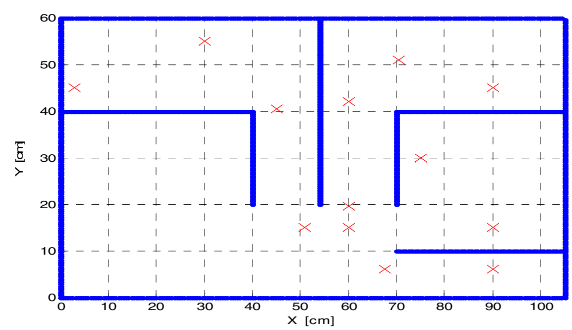

Fig. 1: Simulated Arena

The bi-objective problem, which is described in subsection D, is defined such that the Pareto optimal set of NCs will include both controllers that find the food-targets, as well as controllers that move the robot into the large empty room. Once a robot touches a target the target is removed from the arena. After the removal of the last target, the initial target setup appears once again and so on.

In our research we used one starting point located at x=95 and y=5. The initial direction of the robot is such that it faces to the left.

B. The Simulated Robot

The model of the robot is based on the well-known Khepera robot as described in [6, 7]. The robot has a 5.5 cm diameter circular body with two wheels of 1 cm diameter which are turned by two independent motors. Each motor rotates the associated wheel in a rotational speed within the range [-0.5, 0.5] rad/second. Each simulation step is 5 seconds long and the associated maximal movement distance is 2.5 cm. The Khepera is a modular platform which can generally occupy various sensors. Specifically in our experiments we simulate 16 sensors: 8 Infra-Red (IR) sensors which act as obstacle identifiers and 8 target identifying sensors. The sensor configuration is depicted in figure 2. At each marked location there is one obstacle sensor and one target sensor. Two locations are for rear sensors and six locations cover the front and sides. Sensor characteristics are the same for all sensors of the same type. The maximal sensing range of the IR sensors is 5 cm with beam width span of 6°. Maximum sensing range of the target sensors is 100 cm with beam width span of 30°. The sensors output is a value between the range [0, 1], where 0 is the value for objects at the maximum sensed distance (and beyond) and 1 is for the maximal proximity. All of the sensors are idle with no sensor noise considered.

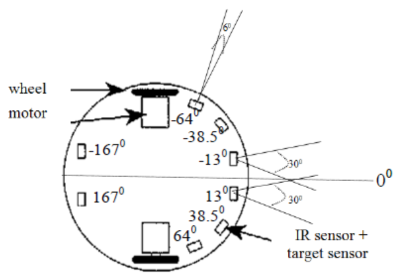

Fig. 2 Khepera robot model with sensors configuration

C. The Neuro-Controller

The control of the simulated robot is achieved by a Feed-Forward Neural-network (FFN). The FFN acts as a non-linear controller which maps input sensor data into motor commands. Although one may use quite a minimal network structure, as in [32], here a larger network is used following [6, 7]. The reason for using a larger network than the minimal one is that we want the evolutionary process to involve a search space of a higher dimension with an apparent permutation problem. With the current number of inputs and outputs, a minimal network with no hidden layer will have only 32 weights, whereas here we use a search space of a much larger dimension as detailed in the following. The selected FFN is based on the studies in [6, 7] where nine classifications of the sensed information have been suggested. The employed network is constructed as follows. The input layer includes 16 components, one per each sensor. The hidden layer includes 9 neurons, and the output layer contains 2 neurons, where each is associated with one of the two wheels. Namely the dimension of the search space in the current study is equal to 16X9 + 9X2 = 162. More details on the used FFN can be found in [7].

D. Objective Functions

Two objective functions are used for evaluating the NC's performance (as in [6, 7]). They are marked by F_1 and F_2. The desire is to maximize both of them. F_1 is based on [37] and is defined as follows:

$$F_1 = \frac{\sum_{i=1}^{final\ step} f_1}{max\ step};\ f_1 = V_i\left(1-\sqrt{\Delta v}\right)_i \left(1-I\right)_i \qquad (1)$$

$$0 \le V \le 1$$
$$0 \le \Delta v \le 1$$
$$0 \le I \le 1$$

Where:

- V is the absolute sum value of the 2 rotational wheel speeds. V is high when the robot is moving straight and fast.
- Δv is the absolute difference value between the two rotational wheel speeds. $1-\sqrt{\Delta v}$ is high when the robot is not making any turns.
- I is the normalized activation value of the sensor with the highest value. $1-I$ is high when the robot does not sense any obstacle.

F_1 is calculated as an average of the accumulated temporary step performances marked as f_1. The sum is taken from the initial step to the final step of the robot. We note that during the evolution some NCs do not manage to complete the maximum allowable number of steps (marked max step). The purpose of F_1 is to achieve fast and straight motions while avoiding obstacles. In this case there is no specific destination.

F_2 is based on [32], and is defined as follows:

$$F_2 = \frac{\sum_{i=1}^{final\ step} f_2}{max\ step};\ f_2 = \begin{cases} H & robot\ reaches\ target \\ 1/\left(1+d\right) & else \end{cases} \qquad (2)$$

Where:

- H is a score the robot gets for reaching a target. Here it is set to 50.
- d is the distance from the robot to the closest of the remaining targets

Similar to F_1, F_2 is calculated as an average over the accumulated temporary step performances marked as f_2. The purpose of F_2 is to achieve an NC that collects as many targets as possible with no concern for obstacle avoidance.

It is shown in [32] that these objectives are contradicting. More specifically, as shown in [6, 7] they are contradicting as related to the current arena and target setup. Hence a Pareto-optimal set of NCs is expected to be found.

E. Evaluation of an NC

Evaluation of an NC is done by a simulation run of the robot roaming the simulated arena. In every run, the robot starts at the same starting location as described in subsection II-A. The maximum number of steps is set to 200 steps. At each step the robot NC receives inputs from the sensors. The output of the NC provides the commands to the wheels, and the robot moves to a new location. Each simulation run with a NC provides the bi-objective performance vector for the NC.

In the evolutionary process any NC that moves the robot into a wall results with a zero performance vector. This prevents any non-physical behavior of passing through walls. Furthermore, a NC that results with the robot going back and forth more than 3 times, without any advancement, stops the simulation and both performance values are set to zero. In such a case, the setting of the performance vector to zero effectively eliminates the NC from being selected as a parent for the next generation.

F. *Evolutionary Algorithms*

This section describes the implementation of the two well-known MOEAs studied here. The first is NSGA-II, based on [4] and the second is MO-CMA-ES based on [5]. In order to have a fair comparison, similar configurations of both algorithms were taken (e.g., number of objective evaluation, number of generation). The particular details of the implementations are outlined in Table I and Table II below. Parameters are denoted as in [4] and [5]. All values that are marked by * are calculated as in [5] with n=162 being the dimension of search space.

TABLE I
NSGA-II CONFIGURATION PARAMETERS

Symbol	Description	Specifications
-	Number of Generation	300
N	Parent population size	44
N	Offspring population size	44
p_c	Crossover probability	0.2
p_m	Mutation probability	0.2
η_c	Crossover distribution index	20
η_m	Mutation distribution index	20
-	Crossover operator	SBX
-	Mutation operator	Polynomial mutation

TABLE II
MO-CMA-ES CONFIGURATION PARAMETERS

Symbol	Description	Specifications
-	Number of Generation	300
λ_{MO}	Parent population size	11
λ	Offspring number	4
p_{succ}^{target}	Target success probability (initialization)	0.1667 *
d	Step size damping (initialization)	21.25 *
c_p	Success rate averaging parameter (initialization)	0.25 *
c_c	Cumulating time horizon parameter (initialization)	0.0122 *
c_{cov}	Covariance matrix learning rate (initialization)	7.61904e-5 *
p_{thresh}	Success rate threshold	0.44 *

IV. RESULTS

In this section we present the result of our comparative experiment. Due to the stochastic nature of both algorithms, we use a large number of repetitive simulations. A total of 61 simulations have been made for both algorithms. Given the lack of an analytical Pareto-optimal solution and the nature of the studied problem, such multiple runs helps to better approximate the solution.

A. *Pareto Fronts Spread*

The Pareto fronts, which were achieved by the multiple runs of the MO-CMA-ES and NSGA-II algorithms, are depicted in Fig. 3 and Fig. 4, respectively. Each front is depicted with a single color for all the performance vectors of that front. Each performance vector represents the performance of a NC. The approximated Pareto front of each algorithm was deduced by finding the non-dominated set of the union of the fronts. For each algorithm, the approximated (combined) front is depicted with a black continuous line connecting the best performance vectors of the combined front. It should be noted that the values of F2 appears to be quantized, which is due to the way F2 is defined.

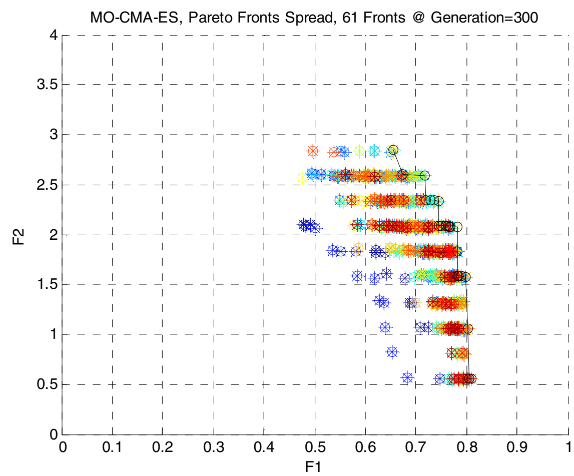

Fig. 3: MO-CMA-ES Pareto Fronts Spread

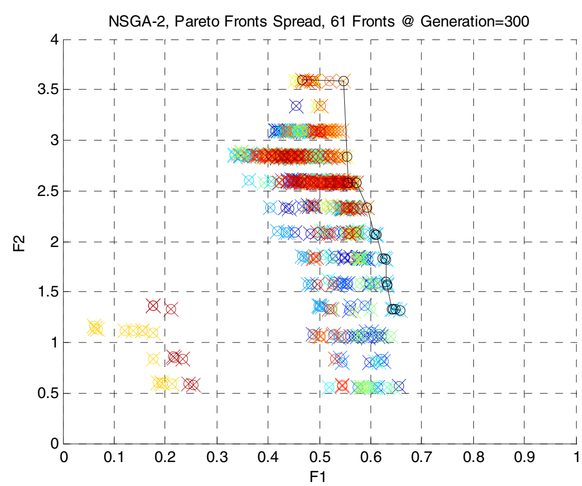

Fig. 4: NSGA-II Pareto Fronts Spread

In each of the figures (3 & 4), we see that the majority of the fronts is concentrated in an area close to the combined front. This supports our approach for analyzing the results by means of statistical inference. It should be noted that different scales are used in the figures, and that in the case of NSGA-II there are quite bad outliers. In contrast the worst front of MO-CMA-ES is relatively close to the combined front.

Regarding the combined fronts, it is discernible that the MO-CMA-ES achieved a much better score in objective F_1 than the one achieved by NSGA-II. On the other hand, it achieved a lower score concerning objective F_2 . The reader is referred to figure 12 in subsection C for some insight about the difference in the best F2 performance.

B. s-Measure Comparison

Fig. 5 and Fig. 6 depict the s-measure versus generation of MO-CMA-ES and NSGA-II, respectively. The figures display, every 10 generations, a boxplot description of the statistics of the s-measure from the multiple runs. The red line inside the boxes marks the median of all runs at each relevant generation. The bottom and top edges of each block are the 25th and 75th percentiles, respectively. The whiskers in each box denotes the most extreme data points still considered as valid measurements and not outliers. Outliers are depicted as red crosses. We note that the number of evaluations at each generation of the ES and GA based evolutions is equal since that the number of mutants in the ES case is equal to the population size in the GA case.

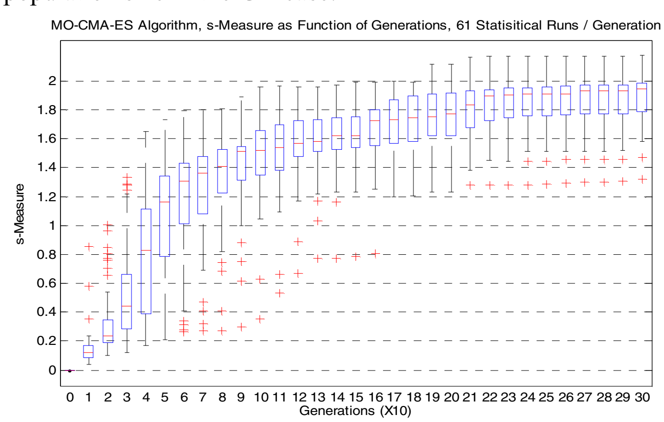

Fig. 5: s-Measure versus Generations (MO-CMA-ES)

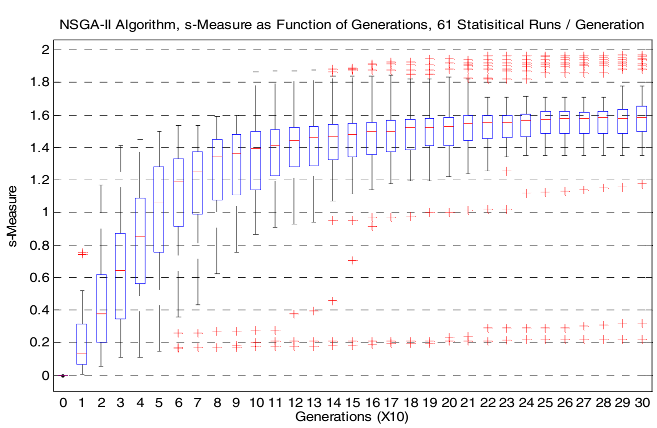

Fig. 6: s-Measure versus Generations (NSGA-II)

It is clear that from an overall viewpoint, as obtained by the s-measure, MO-CMA-ES outperforms the NSGA-II both during the evolution and at the final generation. For example, at 300 generations, when the evolution is stopped the median of the s-measure reaches, in the case of MO-CMA-ES, a value of about 125% of the value obtained by NSGA-II. This large difference is not equal to the difference between the combined fronts, as it is based on the difference between the medians and not on the difference between the combined fronts (see figures 3& 4).

With respect to convergence, it can be seen that in each of the cases, although different, the median value at about generation #220 is quite close to that of generation #300.

C. Best F_1 and F_2 Performances

In the next four figures we present the statistical results for the evolution of the best F_1 and best F_2 as obtained by both algorithms. In all of these figures a clear elitism behavior is depicted. Comparing figures 7 and 8, it can easily be observed that with respect to the best F1, MO-CMA-ES is superior both from the actual value that was reached and from the lower variance viewpoints. In each of the cases, while different, the median value at generation 100 is similar to the value at generation 300.

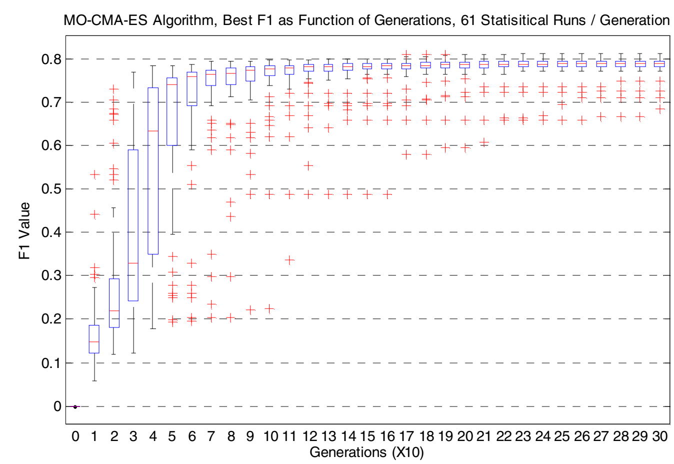

Fig. 7: Best F_1 (MO-CMA-ES)

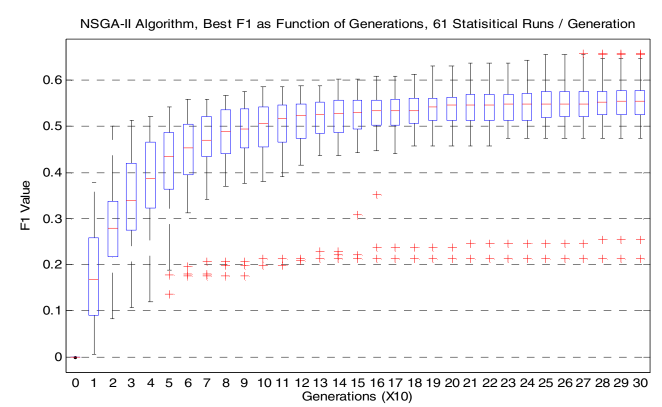

Fig. 8: Best F1 (NSGA-II)

While being superior to NSGA-II for the F1 case, the performance of MO-CMA-ES for the F2 case is not as good as that of NSGA-II. This is easily observed when comparing figures 9 and 10. The median reached at generation 80, by NSGA-II, is higher than the one reached at generation 300 by MO-CMA-ES.

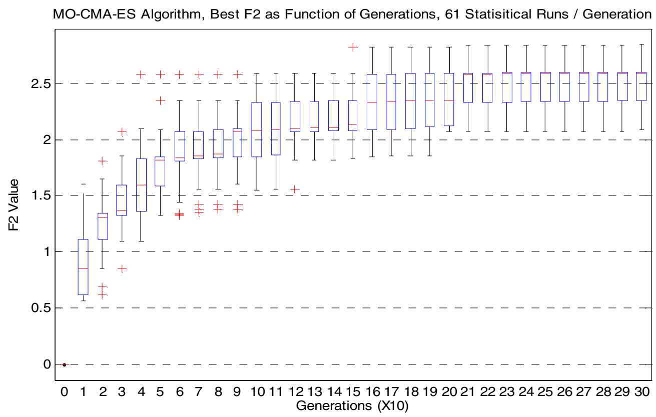

Fig. 9: Best F2 (MO-CMA-ES)

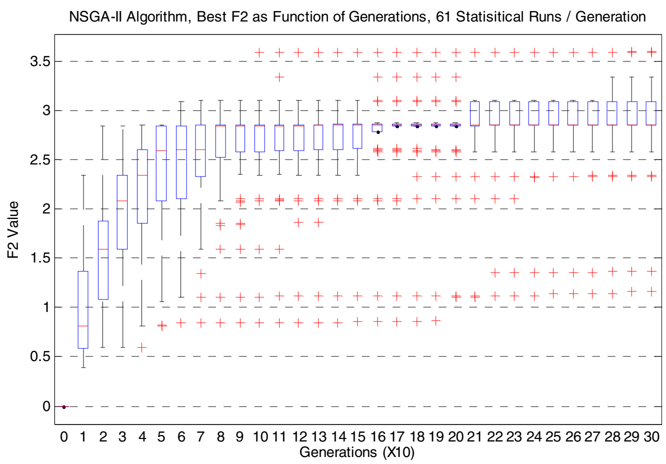

Fig. 10: Best F2 (NSGA-II)

Figures 11 and 12 provide illustrations of typical paths as obtained by a typical best F1 controller and by a typical best F2 controller, respectively. It should be noted that in figure 12 two best F2 controllers are shown, one typically found by NSGA-II and one typically found by MO-CMA-ES. These results are similar in nature to those obtained in [6] and in [7], where the reader can find some detailed explanations concerning the nature of these paths.

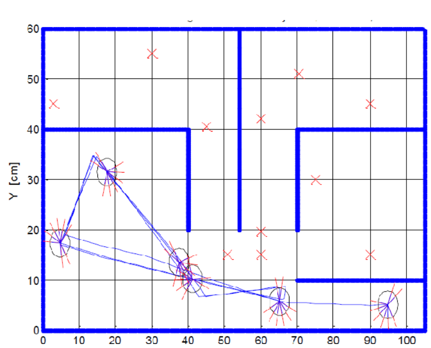

Fig. 11: Typical Path of Best F1

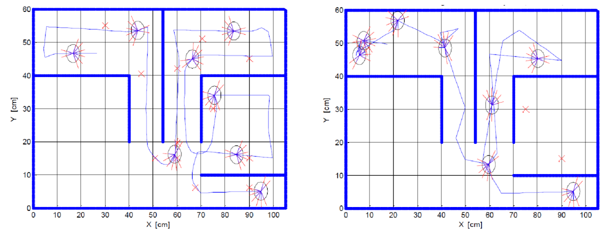

Fig. 12: Typical Path of Best F2 (left: NSGA-II, right: MO-CMA-ES)

V. Discussion

Summarizing the observations from the above section, and considering all aspects of the comparisons, it is concluded that neither algorithm is superior over the other. Yet, careful examination of the comparison of the best F2 and its vicinity indicates that the advantage of NSGA-II is limited to a restricted part of the front. In contrast the MO-CMA-ES is superior over a large part of the front. It should be noted that these conclusions are based on the case examined here. Without an exhaustive study with other NCs applications the generalization of the conclusions is questionable.

As already been suggested in [6] and [7], the multi-objective optimization of the NCs, with the current objective functions and the considered arena, is suspected to involve a multiple local Pareto phenomenon. However, due to the lack of analytical solution and the continuous nature of the decision space, it is impossible to fully verify it. Using sampling near the vicinity of the suspected local Pareto optimal solutions, substantiation of the occurrence of local Pareto fronts has been done (not reported here). This observation suggests that we should examine our comparison of MO-CMA-ES and NSGA-II in view of their comparison using test functions such as ZDT4.

In [5], an s-measure comparison between MO-CMA-ES and NSGA-II has been made for the ZDT4 test function, which has multiple local Pareto fronts. It has been concluded in [5] that NSGA-II is significantly better than MO-CMA-ES for the ZDT4 case. Yet, in contrast to the ZDT4 comparison results of [5], here MO-CMA-ES produces better scores, as compared with NSGA-II.

Originally our hypothesis was that MO-CMA-ES would be superior to NSGA-II due to the permutation problem, which is exhibited in a GA-based optimization of a neural-network (as reported in [25]). However, the results concerning the best F2 contradict this hypothesis.

In the current study we do not try to provide a substantiated explanation to the observed differences in the performances of the algorithms. As hinted above, one possible explanation to the fact that MO-CMA-ES is better, over a large part of the front, is that NSGA-II may suffer from the permutation problem. However, given the elitism operation of NSGA-II, the permutation mechanism should not be able to destroy elitist solutions. It is perceivable that the permutation effect, as realized in NSGA-II, should help diversification even during the convergence stage; hence increasing exploration capabilities without destroying exploitation. This puts any arguments concerning the harmful nature of the permutation phenomenon in question.

VI. CONCLUSIONS

This study compares two well-known MOEAs, which differs primarily by the creation process of offspring. ES versus GA comparison is done for the domain of NCs as applied for robot navigation. In contrast to our expectations MO-CMA-ES is not fully superior when compared with NSGA-II for the studied application. Yet, for a large portion of the sought approximated Pareto optimal solutions it does provide better search capabilities.

It is suggest here that the permutation phenomenon cannot be easily used to explain the observed results. Given the lack of a simple explanation, future studies should try to provide an answer to why there is no clear superior algorithm to all portions of the sought solutions. It is acknowledged that the scope of the employed simulation case is limited and therefore future work should include a wider range of examples. If the observed phenomenon will be persistent in other case studies, then it may lead to a conclusion that an hybrid of search methods should be used to ensure an effective search.

REFERENCES

[1] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganathan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, pp. 32-49, 2011.

[2] S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology, Intelligence and Technology of Self-Organizing Machines*. Bradford Book, 2000.

[3] S. Doncieux, J.-B. Mouret, N. Bredeche, and V. Padois, "Evolutionary robotics: exploring new horizons," in *New Horizons in Evolutionary Robotics*, pp. 3-25, Springer (Ed.), 2011.

[4] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast elitist multi-objective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions*, vol. 6, pp. 182-197, April 2002.

[5] C. Igel, N. Hansen, S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary Computation,* vol 15, pp. 1-28, 2007.

[6] A. Moshaiov and M. Zadok, "Evolution of CPN controllers for multi-objective robot navigation in various environments," Proceedings of the International Workshop on Evolutionary and Reinforcement Learning for Autonomous Robot Systems, (ERLARS 2012), Montpellier, France, 2012.

[7] A. Moshaiov and M. Zadok, "Evolving counter-propagation neuro-controllers for multi-objective robot navigation," Proceedings of the 16th European Conference, EvoApplications 2013, Lecture Notes in Computer Science, LNCS 7835, pp 589-598, 2013.

[8] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolutionary Strategies, Evolutionary Programming, and Genetic Algorithms*. Oxford Press, 1996.

[9] N. Hansen, D. S. Muller and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary Computation*, vol. 11, pp. 1-18, 2003.

[10] A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," The 2005 IEEE Congress on Evolutionary Computation , Vol. 2 , 2-5 Sept. 2005, pp. 1769 – 1776.

[11] H. Kita, "A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms", *Evolutionary Computation* Volume 9 Issue 2, Pages 223 – 241, June 2001.

[12] A. Auger, S. Finck, N. Hansen, and R. Ros, "BBOB 2010: Comparison tables of all algorithms on all noiseless functions, INRIA report N° 0388 ", July 2010.

[13] A. Auger, S. Finck, N. Hansen, R. Ros, "BBOB 2010: Comparison tables of all algorithms on all noisy functions, INRIA report N° 0389", July 2010.

[14] O. Mersmann, M. Preuss, H. Trautmann, "Benchmarking evolutionary algorithms: Towards exploratory landscape analysis," Parallel Problem Solving from Nature, PPSN XI , Part I, LNCS 6238, pp. 73–82, 2010.

[15] K. Deb, *Multi Objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.

[16] C. A. C. Coello and G. B. Lamont, *Applications of Multi-objective Evolutionary Algorithms.* World Scientific 2004.

[17] Y. Jin (Ed.) *Multi-objective Machine Learning*. Springer, 2006.

[18] J. Knowles, D. Crone, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evolutionary Computation*, vol 8, pp. 149-172, 2000.

[19] S. Rostami and A. Shenfield, " CMA-PAES: Pareto archived evolution strategy using covariance matrix adaptation for Multi-Objective Optimisation, " The 12th UK Workshop on Computational Intelligence, Edinburgh, pp. 1 – 8, 5-7 September 2012.

[20] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume*," European J. of Operational Research*, vol 181, pp. 1653-1669, September 2007.

[21] R. Sarker, K-H. Liang, and C. Newton, "A new multiobjective evolutionary algorithm," *European J. of Operational Research*, Vol. 140, 1, pp. 12–23, 2002.

[22] K. Deb, M. Mohan and S. Mishra, "Evaluating the ε-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evolutionary Computation*, vol. 13, pp. 501-525, 2006.

[23] F. Zhao, Z. Zhang, and W. Ma, "The ε -pareto archived evolutionary strategy (ε -PAES) for multi-objective problem," *J. of Computational Information Systems* 8: 20 pp. 8447-8454, 2012.

[24] A. Moshaiov and M. Elias, "Variable-based ε – PAES with Adaptive Fertility Rate," Proceedings of the 13th Annual UK Workshop on Computational Intelligence (UKCI 2013), Guildford, UK, 2013.

[25] X. Yao, "A review of evolutionary artificial neural networks," *International Journal of Intelligent Systems*, vo. 8, pp. 539-567, 1998.

[26] D. Floreano, P. Duerr, and C. Mattiussi, "Neuroevolution: From architectures to learning," *Evolutionary Intelligence*, vol. 1, pp. 47-62, 2008.

[27] A. Azzini and A. Tettamanzi, "Evolutionary ANNs: A State-of-the-art survey*," Intelligenza Artificiale*, vol. 5, 2011.

[28] K. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, 10(2):99–127, 2002.

[29] H. Moriguchi and S. Honiden, " CMA-TWEANN: Efficient Optimization of Neural Networks via Self-Adaptation and Seamless Augmentation, " GECCO 2012.

[30] V. Heidrich-Meisner and C. Igel, "Neuroevolution strategies for episodic reinforcement learning," *J. Algorithms, Cognition, Informatics and Logic*, 64 pp. 152–168, 2009.

[31] A. L. Nelson, G. J. Barlow, and L. Doitsidis, "Fitness Functions in Evolutionary Robotics: A Survey and Analysis," Robotics and Autonomous Systems, vol. 57, pp. 345-370, 2009.

[32] A. Moshaiov and A. Ashram, "Multi-objective Evolution of Robot Neuro-Controllers," Proceedings of the IEEE Congress on Evolutionary Computation, 2009.

[33] J.-B. Mouret and S. Doncieux, "Overcoming the Bootstrap Problem in Evolutionary Robotics Using Behavioral Diversity," Proceedings of the IEEE Congress on Evolutionary Computation, 2009.

[34] S. Israel and A. Moshaiov, "Bootstrapping Aggregate Fitness Selection with Evolutionary Multi-Objective Optimization," Parallel Problem Solving from Nature - PPSN XII, Lecture Notes in Computer Science, LNCS 7492, pp: 52-61, 2012.

[35] R. Salomon, " Increasing Adaptivity through Evolution Strategies," The 4th Int. Conf. on Simulation of Adaptive Behavior (SAB96).

[36] E. Haasdijk, A.E. Eiben, and G. Karafotias, "On-line evolution of robot controllers by an encapsulated evolution strategy," CEC 2010.

[37] D. Floreano and F. Mondada, "Evolution of homing navigation in a real mobile robot," Systems, Man and Cybernetics, Part B, vol.26, no.3, pp.396-407, Jun 1996.