

Three Variants of Three Stage Optimal Memetic Exploration for Handling Non-Separable Fitness Landscapes

Fabio Caraffini^{*†}, Giovanni Iacca[‡], Ferrante Neri^{*†}, and Ernesto Mininno^{*}

^{*}Centre for Computational Intelligence, School of Computer Science and Informatics,
De Montfort University, The Gateway, Leicester LE1 9BH, England, United Kingdom

Email: {fcaraffini, fneri}@dmu.ac.uk

[†]Department of Mathematical Information Technology, University of Jyväskylä

P.O. Box 35 (Agora), 40014 Jyväskylä, Finland

Email: {fabio.caraffini, ferrante.neri, ernesto.mininno}@jyu.fi

[‡]INCAS³ - Innovation Centre for Advanced Sensors and Sensor Systems

P.O. Box 797, 9400 AT Assen, The Netherlands

Email: giovanniacca@incas3.eu

Abstract—Three Stage Optimal Memetic Exploration (3SOME) is a recently proposed algorithmic framework which sequentially perturbs a single solution by means of three operators. Although 3SOME proved to be extremely successful at handling high-dimensional multi-modal landscapes, its application to non-separable fitness functions present some flaws. This paper proposes three possible variants of the original 3SOME algorithm aimed at improving its performance on non-separable problems. The first variant replaces one of the 3SOME operators, namely the middle distance exploration, with a rotation-invariant Differential Evolution (DE) mutation scheme, which is applied on three solutions sampled in a progressively shrinking search space. In the second proposed mechanism, a micro-population rotation-invariant DE is integrated within the algorithmic framework. The third approach employs the search logic (1+1)-Covariance Matrix Adaptation Evolution Strategy, aka (1+1)-CMA-ES. In the latter scheme, a Covariance Matrix adapts to the landscape during the optimization in order to determine the most promising search directions. Numerical results show that, at the cost of a higher complexity, the three approaches proposed are able to improve upon 3SOME performance for non-separable problems without an excessive performance deterioration in the other problems.

I. INTRODUCTION

A function of n independent variables is said to be separable if it can be expressed as a sum of n functions, each of them depending on only one variable. From an optimization viewpoint, these functions are relatively easy to handle as the optimization problem in n variables can be tackled efficiently by perturbing separately each variable. However, real-world applications are often (if not always) characterized by non-separable fitness functions, i.e. functions in which there is some degree of non-linear inter-variable interaction. According to the number of interacting variables, a function can be fuzzily considered fully separable, moderately separable, moderately non-separable, fully non-separable. Modern testbeds tend to classify test problems according to similar criteria, see [1]. It is important to remark that, even if non-separable, an objective

function can still be handled by perturbing separately each variable. Even though this approach does not lead, in general, to the detection of the optimum, it may still be able to detect promising areas of the decision space. Thus, an algorithm combining such a mechanism with some other components more suited for non-separable problems, i.e. involving simultaneous perturbations of multiple variables (diagonal moves) can turn out being efficient and robust.

The idea of performing diagonal search moves is not new: classic optimization methods, such as Rosenbrock and Powell algorithms, already included diagonal moves back to the '60s. In modern computational intelligence optimization, many move operators, e.g. several kinds of recombination in Evolutionary Algorithms (EAs), naturally perform diagonal moves. On the other hand, it is interesting to notice that the separability is explicitly addressed only in a minor portion of literature. An interesting approach is introduced in [2], where non-separability is handled by means of structured populations. Another famous example is the Covariance Matrix Adaptation (CMA) integrated within Evolution Strategy (ES) frameworks, see [3]. The general algorithmic idea is that new trial solutions are generated by a distribution which progressively adapts to the fitness landscape and thus performs search moves along the most convenient direction. Other relevant algorithms based on the CMA have been proposed in literature, e.g. [4], [5], and [6].

However, despite their efficiency, many other modern nature-inspired algorithms inherently perform a biased search along specific axes. This situation occurs for example in Differential Evolution (DE), where crossover is executed inheriting some variables from parent to offspring solutions. As a consequence, even on fitness landscapes where DE is very efficient, if a rotation operation is applied the algorithm performance may dramatically deteriorate. It must be observed that the rotation operation (with respect to the original axes)

over a separable function jeopardizes the separability of the problem making it non-separable. In order to handle these conditions, several corrections to the DE operators have been designed. For example, in [7], a reference rotation procedure is integrated within DE crossover and in [8] a modified DE crossover is introduced by making use of the centroid point. A classical but still efficient way to obtain a rotationally invariant DE is to combine an arithmetic crossover within the mutation scheme. This mutation scheme, namely DE/current-to-rand/1, has been presented in [9].

Recently, in [10], it has been shown that algorithms with a simple structure can be as efficient as more complex methods. This fact has been explained in the light of the Ockham's Razor, and suggested a simple algorithmic design practice for Memetic Computing (MC), based on building up the algorithm with a bottom-up approach. In MC, an algorithm is seen as a structure composed of multiple operators which interact and cooperate to tackle various optimization problems, see [11] and [12]. Following a bottom-up approach, the algorithm is designed from scratch adding the minimum amount of as simple as possible components, each one with a well-defined algorithmic role. As an example of this approach, in [10] a novel MC approach is introduced, named Three Stage Optimal Memetic Exploration (3SOME). Despite its versatility, the main drawback of 3SOME was that the non-separability was not explicitly addressed and thus the algorithm displayed relatively poor performance in some cases. This paper proposes three new variants of the 3SOME structure, where different components specifically designed to tackle non-separability are included in the original algorithmic framework. The remainder of this paper is organized in the following way. Section II briefly introduces the basic structure of 3SOME. Section III describes the proposed variants. Section IV displays the experimental testbed and numerical results related to comparison among 3SOME and the new variants. Section V gives the conclusion of this work.

II. THREE STAGE OPTIMAL MEMETIC EXPLORATION: THE ORIGINAL IMPLEMENTATION

In order to clarify the notation in this paper, we refer to the minimization problem of an objective function $f(x)$, where the candidate solution x is a vector of n design variables (or genes) in a decision space D . The original 3SOME algorithm consists of the following. At the beginning of the optimization problem one candidate solution is randomly sampled within D . In analogy with compact optimization, see [13], we will refer to this candidate solution as elite and indicate it with the symbol x_e . In addition to x_e , the algorithm makes use of another memory slot for attempting to detect other solutions. The latter solution, namely trial, is indicated with x_t . The algorithmic structure is composed of three operators (i.e. exploratory stages) which perturb a single solution, thus exploring the decision space from complementary perspectives.

During the long distance exploration, similar to a stochastic global search, a new trial solution x_t is sampled within the entire decision space, inheriting part (α_e % of n) of the current

elite solution x_e by means of the exponential crossover typical of DE, see [13]. In other words, this exploration stage performs a global stochastic search, attempting to detect unexplored promising basins of attraction. On the other hand, while this search operator extensively explores the decision space, it also promotes retention of a small section of the elite within the trial solution. This kind of inheritance of some genes appears to be extremely beneficial in terms of performance with respect to a stochastic blind search, which would generate a completely new solution at each step. This mechanism is repeated until it does not detect a solution that outperforms the original elite. When a new promising solution is detected, and thus the elite is updated, the middle distance exploration is activated, so to allow a more focused search around it.

In the middle distance exploration stage, a hyper-cube whose edge has side width equal to δ is constructed around the elite solution x_e . Within this region, $k \times n$ trial points are stochastically generated by random perturbing the elite along a limited number of dimensions, thus making a randomized exploitation of the current elite solution. In other words, this stage attempts to focus the search around promising solutions in order to determine whether the current elite deserves further computational budget or other unexplored areas of the decision space must be explored. If the elite is outperformed, it is replaced. A replacement occurs also if one of the newly generated solutions has the same performance of the elite, in order to prevent the search getting trapped in some plateaus of the decision space. At the end of this stage, if the elite has been updated a new hypercube is constructed around the new elite and this mechanism is repeated. On the contrary, if the middle distance exploration does not lead to an improvement, an alternative search logic is applied, that is the deterministic logic of the short distance exploration.

This final search stage perturbs the variables separately and attempts to quickly and deterministically descend the corresponding basin of attraction. The meaning of the short distance exploration is to perform the descent of promising basins of attraction and possibly finalize the search if the basin of attraction is globally optimal. De facto, this operator is a simple steepest descent deterministic local search algorithm, with an exploratory move similar to that of Hooke-Jeeves algorithm, or the first local search algorithm of the multiple trajectory search, see [14]. The short distance exploration stage requires an additional memory slot, which will be referred to as x_s (s stands for short). Starting from the elite x_e , this local search, explores each coordinate i and samples $x_s[i] = x_e[i] - \rho$, where ρ is the exploratory radius. Subsequently, if x_s outperforms x_e , the trial solution x_t is updated (it takes the value of x_s), otherwise a half step in the opposite direction $x_s[i] = x_e[i] + \frac{\rho}{2}$ is performed. Again, x_s replaces x_t if it outperforms x_e . If there is no update, i.e. the exploration is unsuccessful, the radius ρ is halved. This exploration is repeated for all the design variables and stopped when a prefixed budget (equal to 150 iterations) is exceeded. After that, if there is an improvement in the quality of the solution, the focused search of middle distance exploration is repeated

subsequently. Otherwise, if no improvement in solution quality is found, the long distance search is activated again to attempt to find new basins of attractions.

As a remark the original 3SOME algorithm applies a toroidal management of the bounds. This means that if, along the dimension i , the design variable $x[i]$ exceeds the bounds of a value ζ , it is reinserted from the other end of the interval at a distance ζ from the edge, i.e. given an interval $[a, b]$, if $x[i] = b + \zeta$ it takes the value of $a + \zeta$. The same toroidal mechanism is used also in the algorithms proposed in this paper. For further details about 3SOME, the interested reader is referred to [10].

III. IMPROVING UPON 3SOME: THREE POSSIBLE APPROACHES FOR NON-SEPARABILITY

A careful analysis of the 3SOME structure described above suggests that the long and the short distance operators are somehow algorithmically necessary to properly balance exploration and exploitation, see [15], [16], [17] and [18]. In addition to that, the deterministic short distance search is clearly very efficient at handling separable functions, while its application can be detrimental with non-separable problems. This observation is also supported by an empirical study of the original 3SOME dynamics we performed on a set of non-separable benchmark functions. Thus, in order to improve upon 3SOME, balancing the effect of the deterministic local search, a simple idea is to slightly modify its original structure replacing the middle search operator with an operator specifically tailored for handling non-separability. In this section we propose three different variants inspired by this consideration.

A. Rotation Invariant Shrinking 3SOME

The first approach for handling non-separability, we here refer to as Rotation Invariant Shrinking 3SOME (RIS-3SOME), applies instead of the middle distance exploration operator the following mechanism. Around the solution x_e returned by the long (or short) distance exploration, a hypercube is considered. This hypercube has an initial volume empirically set equal to one fifth of the volume of the entire decision space D and is centered around x_e . Within this hypercube, three points x_r , x_s , and x_v , are sampled (from an implementation viewpoint they are sampled one-by-one and allocated into the trial solution to occupy only one memory slot). These points are then combined with x_e by means of DE/current-to-rand/1 to generate x_t :

$$x_t = x_e + K(x_v - x_e) + F'(x_r - x_s) \quad (1)$$

where K is the combination coefficient, which should be chosen with a uniform random distribution from $[0, 1]$ and $F' = K \cdot F$. Exactly like in the case of DE mutation, the scale factor F is a parameter of the algorithm. If the trial solution x_t displays a higher performance than the current elite x_e , a new repetition of sampling and mutation is repeated within the same hypercube. Otherwise, the volume of the hypercube is halved (shrinking). This process is repeated until the search volume reaches a threshold ε in terms of ratio of the entire

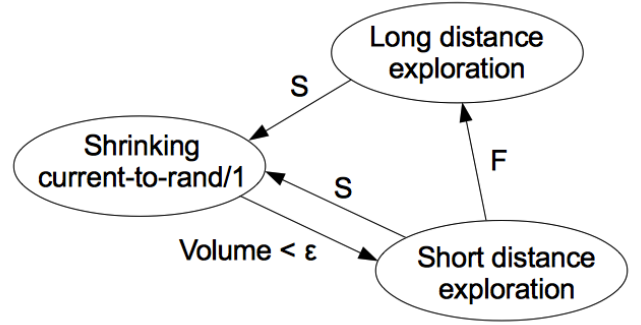


Fig. 1. Coordination scheme of RIS-3SOME

decision space. When this condition holds, the short distance exploration is applied, as in the original 3SOME.

In order to understand the algorithmic contribution of this mechanism, i.e. the generation of points by eq. (1) within a shrinking hypercube, it is important to remark that, similarly to the original middle distance operator, this mechanism encompasses a form of inheritance from x_e to x_t . However, in this case the inheritance is implemented by means of a linear combination and a search along all the directions simultaneously. As it can be geometrically proved, this component is rotation-invariant, thus it is supposed to tackle, in a simple and computationally inexpensive way, non-separable problems. Furthermore, the shrinking progressively narrows down the search space, thus promoting a progressive exploitative pressure.

Figure 1 shows the coordination scheme of the three components of RIS-3SOME. Similar to a Finite State Machine (FSM), the algorithm is described as a composition of states, each one corresponding to a single operator (or meme). Each operator processes an elite x_e and returns, as an output, a (possibly) fitness-wise improved elite solution. The operator can be said to “succeed” if it is able to improve upon the incoming elite, otherwise it can be said to “fail”. With reference to figure 1, the arrows represent the interaction amongst memes. The “S” and “F”, represent success and failure, respectively, of the meme, while the condition on the search volume in the shrinking component is labeled explicitly.

B. Micro-Population Differential Evolution 3SOME

The second variant we propose uses a micro-population Differential Evolution instead of the original middle distance exploration. The operating principle of this algorithm, named μ DE-3SOME, is the following. Whenever the long (or short) distance exploration returns a new elite, a micro-population of m individuals is sampled within a hypercube centered around x_e , whose volume has been empirically set equal to 40% of the volume of the entire search space. The worst individual of the micro-population is then replaced with the current elite. Subsequently, for a fixed number of iterations, a run of DE/current-to-rand/1 with exponential crossover is

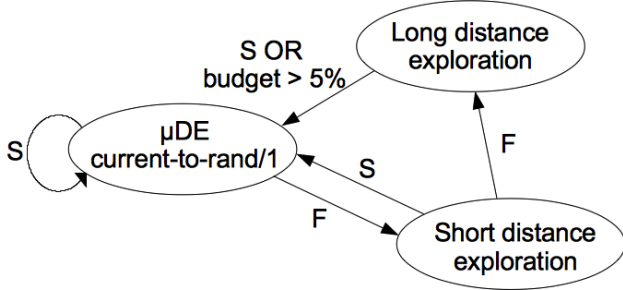


Fig. 2. Coordination scheme of μ DE-3SOME

executed over the micro-population. When the given budget allotted to the μ DE operator is reached, if an improvement is found a new hypercube is constructed around the new elite and μ DE is repeated. Otherwise, the short distance exploration is activated. Compared to 3SOME and RIS-3SOME, the only difference in the inter-operator coordination logics is that, in order to force a more frequent activation of the μ DE operator, thus guaranteeing its convergence, a budget limit equal to 5% of the total budget (in terms of fitness evaluations) is imposed over each activation of the long distance exploration. After this limit is reached, μ DE is activated regardless the long distance exploration has improved upon the current elite or not. This additional control also guarantees a balance in the activation of each of the three operators similar to that one of 3SOME and RIS-3SOME.

It should be noticed that, similar to RIS-3SOME, the μ DE meme naturally embeds a form of “shrinking” over the most promising search region. Moreover, the current-to-rand/1 mutation scheme again guarantees a rotation-invariant behaviour. However, compared to 3SOME and RIS-3SOME this variant is slightly more expensive on a memory viewpoint, because it needs m additional memory slots to store the micro-population. On the other hand, the computational cost is comparable to the two previous algorithms.

Following the same notation used in figure 1, figure 2 shows the coordination scheme of the three components of μ DE-3SOME, where the additional budget control on the long distance exploration is explicitly indicated, and the self-loop on the μ DE meme denotes a repetition in case of success.

C. 3SOME with 1+1 Covariance Matrix Adaptation Evolution Strategy

The last 3SOME variant we propose in this paper replaces the middle distance exploration with the (1+1)-CMA-ES algorithm presented in [6]. The latter algorithm combines a classic (1+1)-ES scheme with an improved Covariance Matrix Adaptation mechanism [3], where an incremental update of the covariance matrix Cholesky factors is performed instead of computing the Cholesky decomposition. In addition to that, (1+1)-CMA-ES does not employ a population of solutions - unlike CMA-ES - but rather it explores the search space using a single solution. Even so, (1+1)-CMA-ES requires to

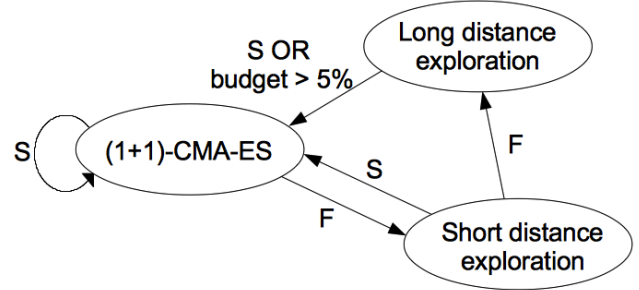


Fig. 3. Coordination scheme of (1+1)-CMA-ES-3SOME

store a covariance matrix, with a memory employment that grows quadratically with the problem dimension. On the other hand, (1+1)-CMA-ES is computationally less demanding and numerically involved than CMA-ES, still being able to obtain similar performance, especially on non-separable problems.

The resulting combination of (1+1)-CMA-ES and 3SOME makes use of the same coordination scheme of 3SOME, in which a run of (1+1)-CMA-ES is executed in place of the middle distance search operator, for a fixed budget. Similarly to μ DE-3SOME, each activation of the long distance stage is given a maximum budget equal to 5% of the total budget of the algorithm. This mechanism ensures a better balance among the three stages, thus preventing budget mis-allocation. We should remark that, due to the covariance matrix, (1+1)-CMA-ES-3SOME requires more memory than the first two variants. For the sake of clarity, figure 3 displays the coordination scheme of (1+1)-CMA-ES-3SOME.

IV. NUMERICAL RESULTS

In order to understand the algorithmic contribution provided by each of the three variants described above, we compared them with the original implementation of 3SOME on the entire noiseless Black-Box Optimization Benchmark 2010 (BBOB) [1], consisting of 24 test functions with different properties in terms of modality, separability, and ill-conditioning. To test the scalability of the proposed approaches, we ran the whole benchmark in 10, 20, 40 and 100 dimensions, thus considering $24 \times 4 = 96$ functions in total.

As for the parameter setting, 3SOME was executed using the parameters suggested in [10], namely inheritance factor for $\alpha_e = 0.05$, δ and ρ respectively equal to 20% and 40% of the total decision space width, and coefficient of generated points at each activation of the middle distance exploration $k = 4$.

RIS-3SOME was executed with the same parameter setting for α_e and ρ , while the DE/current-to-rand/1 mutation was applied with scale factor $F = 0.4$, and the threshold ε was set equal to $1e - 4$.

Also μ DE-3SOME was executed with the same values of α_e and ρ , $m = 5$ individuals, scale factor $F = 0.75$, and number of DE iterations equal to the problem dimension n .

The same values of α_e and ρ were used also in (1+1)-CMA-ES-3SOME. As for (1+1)-CMA-ES, the standard values used in its original Java implementation available on [19] and suggested in the original paper were used, namely $c_p = 1/12$, $p_{succ}^{target} = 2/11$, $p_{thresh} = 0.44$ and $\sigma_0 = 1$. The budget for each activation of (1+1)-CMA-ES was set to $10 \times n$.

Each algorithm has been run for $5000 \times n$ fitness evaluations for each run. For each problem 100 runs have been performed. All the experiments were implemented in Java and executed on a cluster of 160 Pentium 2.4 GHz cores using the optimization platform Kimeme [20].

Numerical results are shown in tables I-IV, expressed as average final value and standard deviation. The best results are highlighted in bold face. In order to strengthen the statistical significance of the results, the Wilcoxon Rank-Sum test has also been applied according to the description given in [21], where the confidence level has been fixed at 0.95. The symbols “=” and “+” (“-”) indicate, respectively, a statistically equivalent performance and a better (worse) performance of original 3SOME compared with the algorithm labeled on the top of the column.

In 10 dimensions, RIS-3SOME is able to improve upon 3SOME in 10 out of 24 functions, while in 11 cases they are statistically equivalent, and only in 3 cases RIS-3SOME degrades the original performance of 3SOME: two of these three functions, namely f_3 and f_4 , are indeed separable. In general, RIS-3SOME seems to perform better than 3SOME especially on non-separable multi-modal functions showing an adequate or weak of “global structure” [1], particularly the group of functions $f_{15}-f_{19}$. As for μ DE-3SOME, it outperforms 3SOME in 15 cases, it is outperformed only in 3 unimodal cases (f_3 , f_{10} , and f_{12}), and it shows a similar performance in the remaining 6 cases. Thus μ DE-3SOME seems to consistently and regularly improve upon 3SOME, especially on non-separable multi-modal functions. Similarly, (1+1)-CMA-ES-3SOME outperforms 3SOME in 15 cases, it is outperformed in 4 cases (2 separable functions), and it is equivalent in 5 cases. In this case the improvement provided by the CMA-ES scheme seems to be less focused on a specific group of functions, but rather “structural”, since it displays a better performance both on separable (e.g. f_1 and f_2) and non-separable functions, particularly unimodal (f_{11} , f_{12} , f_{14}).

In 20 dimensions, a similar trend emerges. RIS-3SOME improves upon 3SOME in 11 cases (again on non-separable multi-modal functions with global structure), while it is outperformed in 6 cases and it equals 3SOME in the remaining 7 cases. Similarly, μ DE-3SOME outperforms 3SOME in 11 cases, it is outperformed on 5 functions, and it shows the same performance as 3SOME on 8 cases. In general μ DE-3SOME seems to be indeed better suited than 3SOME for non-separable functions (both unimodal and multi-modal, especially with global structure). Also (1+1)-CMA-ES-3SOME clearly outperforms 3SOME: in 17 cases out of 24 it obtains a better result, while it degrades the 3SOME performance only in 4 cases. Again, (1+1)-CMA-ES-3SOME seems to be globally better than 3SOME, although seems to be extremely

good especially on unimodal functions, both separable and non-separable (e.g. f_1 , f_2 and the function group $f_{10}-f_{14}$).

Similar results were obtained also on 40 dimensions. RIS-3SOME displays a better performance than 3SOME in 12 cases (especially non-separable multi-modal functions with global structure), and a worse performance in only 4 cases (among which again f_3 and f_4). μ DE-3SOME outperforms 3SOME in 11 cases, while it is outperformed in 6 cases: once again it seems to obtain better results especially on non-separable functions, both unimodal and multi-modal. (1+1)-CMA-ES-3SOME instead outperforms 3SOME on 15 test functions, with different properties in terms of modality and separability, and it is outperformed in 6 cases (either separable, see f_3 and f_4 , or not, see f_8 and f_9).

These results are confirmed even in 100 dimensions, although in this case the advantages obtained modifying the original structure of 3SOME appear less prominent. In particular, RIS-3SOME outperforms 3SOME in 9 cases, it is outperformed in 7 cases, and it equals 3SOME in the remaining 8 cases: similarly to lower dimensionalities, the improvements are more evident on non-separable multi-modal functions, but in this case only on those having an adequate global structure (function group $f_{15}-f_{19}$). μ DE-3SOME performs better than 3SOME in 11 cases, while it is outperformed in 8 cases. Also in this case the pattern suggests the μ DE is better suited for non-separable multi-modal functions with global structure. Finally, (1+1)-CMA-ES-3SOME displays a better performance than 3SOME in 13 cases (either separable or non-separable), while 3SOME is more promising in 9 other cases. However, in this case there is no clear evidence of a global scheme, except that (1+1)-CMA-ES-3SOME seems to outperform 3SOME especially on non-separable unimodal functions ($f_{10}-f_{14}$).

From the numerical results above summarized a few conclusions can be drawn. First of all, it is quite evident that the three 3SOME variants here proposed are all able to improve, sometimes remarkably, upon 3SOME. This is specially true for non-separable functions with lower dimensionalities (from 10 to 40), while on semi-large scale problems (100 dimensions) the performance improvement is relatively limited. On lower dimensions, the results obtained in this study show also some more specific trends. Referring to the property taxonomy used to structure the BBOB 2010 benchmark, it seems that the two variants based on DE/current-to-rand/1, namely RIS-3SOME and μ DE-3SOME, are able to better exploit the global structure of some landscapes, and in general they show similar performances on the whole benchmark from 10 to 40 dimensions, tending to outperform 3SOME on non-separable functions, especially multi-modal. In a nutshell, these two variants can be considered equivalent in terms of global performance: this can be explained considering that, although their coordination scheme is different, both RIS-3SOME and μ DE-3SOME rely on the same DE/current-to-rand/1 mutation scheme. On the other hand, the combination of 3SOME with the (1+1)-CMA-ES structure seems to produce the best global results on lower dimensions, both on separable and non-separable functions. However, compared to the first two simpler schemes, (1+1)-

CMA-ES-3SOME leads to minor improvements. A possible interpretation of these results is that, despite its robustness and mathematical elegance, (1+1)-CMA-ES is still prone to converge to local optima, especially on highly multimodal problems. Thus its application within the 3SOME structure appears to be beneficial only on those landscapes whose number of optima does not grow with the problem dimension. In other words, especially on high dimensional problems, simpler approaches like 3SOME or RIS-3SOME are already successful without adding more complexity. This finding is in line with the Ockham's Razor [10].

Related to that, it must be remarked that the proposed approaches are characterized not only by different computational complexity, but also by different memory footprint. As in [10] and [13], we measure memory footprint in terms of number of "memory slots" needed for the algorithm execution, i.e. n -dimensional arrays of numeric values (floating/fixed point double/single precision numbers, according to platform and implementation). More specifically, while RIS-3SOME requires only three memory slots (one for the elite, one for the trial solution, and one to store the initial elite which is used for replacements in the short distance operator), μ DE-3SOME employs $3+m$ slots, where m additional slots are needed to store the micro-population (5 in our experiments), and (1+1)-CMA-ES-3SOME uses employs $3+n^2$ slot, where the quadratic term refers to the covariance matrix. Thus the first two methods are more suited for those applications plagued by severe memory constraints (e.g. embedded systems, wireless sensors, wearable devices, micro-robots, etc.), while μ DE-3SOME and (1+1)-CMA-ES-3SOME should be applied, respectively, in cases in which memory is moderately or largely available (from FPGAs to tablets and PC). Thus, if one has to find the best trade-off between performance and memory consumption, RIS-3SOME and μ DE-3SOME should be preferred.

V. CONCLUSION

This paper proposes three novel variants of the 3SOME framework, attempting to improve upon its performance on non-separable problems. The proposed approaches replace the original middle exploration operator with three different algorithmic structures, two of them being based on DE/current-to-rand/1 (RIS-3SOME and μ DE-3SOME), and one on (1+1)-CMA-ES. In addition to that, they slightly modify the original coordination logics of 3SOME, still keeping a simple structure and processing only one solution. Apart from (1+1)-CMA-ES-3SOME which stores a covariance matrix, the other two variants are also characterized by modest memory requirements, thus being suitable for embedded implementations. Numerical results on the BBOB 2010 benchmark executed with different dimensions show that the proposed approaches outperform 3SOME especially on non-separable functions, without degrading its global performance. The conclusion of this study is that, although (1+1)-CMA-ES-3SOME seems to be slightly more promising than the other two algorithms, the best trade-off between complexity/memory footprint and

robustness is provided by simpler approaches such as RIS-3SOME and μ DE-3SOME.

ACKNOWLEDGMENT

This research is supported by the Academy of Finland, Akatemiattutkija 130600 and Tutkijatohtori 140487. INCAS³ is co-funded by the Province of Drenthe, the Municipality of Assen, the European Fund for Regional Development and the Ministry of Economic Affairs, Peaks in the Delta.

REFERENCES

- [1] N. Hansen, A. Auger, S. Finck, R. Ros *et al.*, "Real-parameter black-box optimization benchmarking 2010: Noiseless functions definitions," INRIA, Tech. Rep. RR-6829, 2010.
- [2] D. Whitley, S. Rana, and R. B. Heckendorn, "The island model genetic algorithm: On separability, population size and convergence," *Journal of Computing and Information Technology*, vol. 7, pp. 33–47, 1998.
- [3] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1996, pp. 312–317.
- [4] —, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [5] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.
- [6] C. Igel, T. Suttorp, and N. Hansen, "A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM Press, 2006, pp. 453–460.
- [7] T. Takahama and S. Sakai, "Solving nonlinear optimization problems by differential evolution with a rotation-invariant crossover operation using gram-schmidt process," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing*, 2010, pp. 533–540.
- [8] —, "Efficient nonlinear optimization by differential evolution with a rotation-invariant local sampling operation," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2011, pp. 2215–2222.
- [9] K. Price, "An introduction to differential evolution," in *New Ideas in Optimization*, D. Corne, M. Dorigo, F. Glover, D. Dasgupta, P. Moscato, R. Poli, and K. V. Price, Eds. McGraw-Hill, 1999, pp. 79–108.
- [10] G. Iacca, F. Neri, E. Mininno, Y. S. Ong, and M. H. Lim, "Ockham's razor in memetic computing: Three stage optimal memetic exploration," *Information Sciences*, vol. 188, pp. 17–43, 2012.
- [11] F. Neri, C. Cotta, and P. Moscato, *Handbook of Memetic Algorithms*, ser. Studies in Computational Intelligence. Springer, 2012, vol. 379.
- [12] F. Neri and C. Cotta, "Memetic algorithms and memetic computing optimization: A literature review," *Swarm and Evolutionary Computation*, vol. 2, pp. 1–14, 2012.
- [13] F. Neri, G. Iacca, and E. Mininno, "Disturbed exploitation compact differential evolution for limited memory optimization problems," *Information Sciences*, vol. 181, no. 12, pp. 2469–2487, 2011.
- [14] L. Y. Tseng and C. Chen, "Multiple trajectory search for large scale global optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2008, pp. 3052–3059.
- [15] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flow shop scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 204–223, 2003.
- [16] H. Ishibuchi, Y. Hitotsuyanagi, and Y. Nojima, "An empirical study on the specification of the local search application probability in multiobjective memetic algorithms," in *Proc. of the IEEE Congress on Evolutionary Computation*, September 2007, pp. 2788–2795.
- [17] K. Tan, S. Chiam, A. Mamun, and C. Goh, "Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization," *European Journal of Operational Research*, vol. 197, pp. 701–713, 2009.
- [18] M. Lozano and C. García-Martínez, "Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report," *Computers & Operations Research*, vol. 37, no. 3, pp. 481–497, 2010.

TABLE I
AVERAGE FITNESS ± STANDARD DEVIATION AND WILCOXON RANK-SUM TEST IN 10 DIMENSIONS (REFERENCE = 3SOME)

	3SOME	(1+1)-CMA-ES-3SOME		μ DE-3SOME		RIS-3SOME	
f_1	7.95e + 01 ± 1.21e - 14	7.95e + 01 ± 0.00e + 00	-	7.95e + 01 ± 1.06e - 14	=	7.95e + 01 ± 1.03e - 14	=
f_2	-2.10e + 02 ± 1.63e - 14	-2.10e + 02 ± 0.00e + 00	-	-2.10e + 02 ± 1.52e - 14	=	-2.10e + 02 ± 1.35e - 14	=
f_3	-4.61e + 02 ± 1.18e + 00	-4.56e + 02 ± 2.69e + 00	+	-4.60e + 02 ± 9.81e - 01	+	-4.54e + 02 ± 4.41e + 00	+
f_4	-4.60e + 02 ± 1.39e + 00	-4.55e + 02 ± 3.12e + 00	+	-4.60e + 02 ± 1.54e + 00	-	-4.51e + 02 ± 6.80e + 00	+
f_5	5.33e + 00 ± 2.91e + 01	6.41e + 00 ± 3.04e + 01	=	-4.99e + 00 ± 8.59e + 00	-	-7.56e + 00 ± 9.86e + 00	-
f_6	8.25e + 01 ± 2.83e + 02	1.50e + 02 ± 7.66e + 02	+	3.70e + 01 ± 1.13e + 01	-	3.59e + 01 ± 1.71e - 03	=
f_7	1.05e + 02 ± 1.23e + 01	9.32e + 01 ± 4.00e - 01	-	1.01e + 02 ± 4.58e + 00	-	1.03e + 02 ± 9.01e + 00	=
f_8	1.49e + 02 ± 1.86e - 01	1.49e + 02 ± 0.00e + 00	-	1.49e + 02 ± 1.51e - 01	=	1.49e + 02 ± 1.49e - 01	=
f_9	1.25e + 02 ± 1.69e + 00	1.25e + 02 ± 1.50e + 00	-	1.24e + 02 ± 9.47e - 01	-	1.26e + 02 ± 1.01e + 01	=
f_{10}	3.95e + 03 ± 2.63e + 04	2.26e + 03 ± 2.31e + 04	-	4.03e + 03 ± 1.10e + 04	+	2.60e + 02 ± 1.60e + 02	=
f_{11}	1.57e + 02 ± 3.36e + 01	7.63e + 01 ± 0.00e + 00	-	1.29e + 02 ± 2.62e + 01	-	1.36e + 02 ± 2.72e + 01	-
f_{12}	-6.12e + 02 ± 1.33e + 01	-6.21e + 02 ± 1.02e + 00	-	-6.00e + 02 ± 2.19e + 01	+	-6.08e + 02 ± 1.63e + 01	=
f_{13}	4.26e + 01 ± 1.28e + 01	4.06e + 01 ± 1.08e + 01	=	3.83e + 01 ± 8.89e + 00	=	4.09e + 01 ± 1.11e + 01	=
f_{14}	-5.23e + 01 ± 3.05e - 05	-5.23e + 01 ± 1.94e - 11	-	-5.23e + 01 ± 2.01e - 05	-	-5.23e + 01 ± 2.40e - 05	=
f_{15}	1.10e + 03 ± 6.38e + 01	1.08e + 03 ± 4.71e + 01	=	1.06e + 03 ± 2.48e + 01	-	1.06e + 03 ± 2.92e + 01	-
f_{16}	7.97e + 01 ± 4.63e + 00	7.83e + 01 ± 3.91e + 00	-	7.59e + 01 ± 2.19e + 00	-	7.71e + 01 ± 3.53e + 00	-
f_{17}	-1.03e + 01 ± 6.57e + 00	-1.28e + 01 ± 2.39e + 00	-	-1.32e + 01 ± 3.40e + 00	-	-1.45e + 01 ± 1.50e + 00	-
f_{18}	5.80e + 00 ± 2.56e + 01	-2.47e + 00 ± 9.57e + 00	=	-4.37e - 01 ± 1.65e + 01	=	-9.02e + 00 ± 4.48e + 00	-
f_{19}	-9.80e + 01 ± 2.98e + 00	-9.94e + 01 ± 1.81e + 00	-	-1.00e + 02 ± 1.49e + 00	-	-1.00e + 02 ± 1.38e + 00	-
f_{20}	-5.46e + 02 ± 2.59e - 01	-5.45e + 02 ± 3.69e - 01	+	-5.46e + 02 ± 2.64e - 01	-	-5.45e + 02 ± 3.02e - 01	+
f_{21}	5.36e + 01 ± 1.34e + 01	4.82e + 01 ± 7.15e + 00	-	4.46e + 01 ± 4.07e + 00	-	4.73e + 01 ± 6.24e + 00	-
f_{22}	-9.88e + 02 ± 1.55e + 01	-9.91e + 02 ± 1.29e + 01	-	-9.98e + 02 ± 3.03e + 00	-	-9.94e + 02 ± 8.23e + 00	-
f_{23}	7.86e + 00 ± 4.95e - 01	7.86e + 00 ± 5.54e - 01	=	7.60e + 00 ± 3.09e - 01	-	7.88e + 00 ± 5.66e - 01	=
f_{24}	1.92e + 02 ± 4.46e + 01	1.72e + 02 ± 2.72e + 01	-	1.57e + 02 ± 1.57e + 01	-	1.61e + 02 ± 2.03e + 01	-

TABLE II
AVERAGE FITNESS ± STANDARD DEVIATION AND WILCOXON RANK-SUM TEST IN 20 DIMENSIONS (REFERENCE = 3SOME)

	3SOME	(1+1)-CMA-ES-3SOME		μ DE-3SOME		RIS-3SOME	
f_1	7.95e + 01 ± 1.70e - 14	7.95e + 01 ± 0.00e + 00	-	7.95e + 01 ± 1.77e - 14	=	7.95e + 01 ± 1.68e - 14	=
f_2	-2.10e + 02 ± 1.99e - 14	-2.10e + 02 ± 1.07e - 14	-	-2.10e + 02 ± 2.33e - 14	=	-2.10e + 02 ± 2.07e - 14	=
f_3	-4.59e + 02 ± 1.86e + 00	-4.40e + 02 ± 7.82e + 00	+	-4.56e + 02 ± 2.50e + 00	+	-4.39e + 02 ± 7.33e + 00	+
f_4	-4.57e + 02 ± 2.53e + 00	-4.37e + 02 ± 7.92e + 00	+	-4.53e + 02 ± 3.25e + 00	+	-4.37e + 02 ± 8.66e + 00	+
f_5	2.05e + 01 ± 7.73e + 01	4.88e + 00 ± 5.59e + 01	-	9.27e + 00 ± 3.66e + 01	-	-9.21e + 00 ± 2.07e - 13	-
f_6	3.59e + 01 ± 5.12e - 06	3.59e + 01 ± 0.00e + 00	-	1.94e + 02 ± 1.12e + 03	+	3.59e + 01 ± 6.08e - 07	=
f_7	1.16e + 02 ± 1.60e + 01	9.85e + 01 ± 4.74e + 00	-	1.11e + 02 ± 1.06e + 01	=	1.16e + 02 ± 1.40e + 01	=
f_8	1.49e + 02 ± 5.28e - 01	1.49e + 02 ± 1.02e - 04	-	1.50e + 02 ± 7.77e + 00	=	1.50e + 02 ± 6.82e - 01	+
f_9	1.25e + 02 ± 1.68e + 00	1.25e + 02 ± 1.59e + 00	-	1.26e + 02 ± 1.95e + 00	=	1.27e + 02 ± 7.63e + 00	+
f_{10}	2.90e + 02 ± 2.34e + 02	-4.12e + 01 ± 8.58e + 01	-	2.04e + 04 ± 8.55e + 04	+	3.12e + 02 ± 2.70e + 02	=
f_{11}	2.55e + 02 ± 8.50e + 01	9.35e + 01 ± 1.22e + 02	-	2.01e + 02 ± 4.94e + 01	-	1.97e + 02 ± 3.28e + 01	-
f_{12}	6.89e + 06 ± 4.84e + 07	-6.20e + 02 ± 1.96e + 00	-	-5.95e + 02 ± 2.71e + 01	-	-6.05e + 02 ± 2.32e + 01	-
f_{13}	3.78e + 01 ± 1.01e + 01	3.88e + 01 ± 1.24e + 01	=	5.07e + 01 ± 1.66e + 01	+	4.35e + 01 ± 1.36e + 01	+
f_{14}	-5.23e + 01 ± 8.13e - 05	-5.23e + 01 ± 4.98e - 08	-	-5.23e + 01 ± 7.97e - 05	-	-5.23e + 01 ± 9.11e - 05	=
f_{15}	1.27e + 03 ± 1.58e + 02	1.24e + 03 ± 9.68e + 01	=	1.17e + 03 ± 6.42e + 01	-	1.17e + 03 ± 7.33e + 01	=
f_{16}	8.37e + 01 ± 5.89e + 00	8.28e + 01 ± 4.97e + 00	=	8.12e + 01 ± 3.85e + 00	-	8.02e + 01 ± 4.29e + 00	-
f_{17}	-7.05e + 00 ± 5.64e + 00	-9.84e + 00 ± 2.66e + 00	-	-1.15e + 01 ± 2.19e + 00	-	-1.22e + 01 ± 1.90e + 00	-
f_{18}	2.08e + 01 ± 2.63e + 01	8.62e + 00 ± 1.86e + 01	-	6.08e + 00 ± 1.20e + 01	-	2.35e + 00 ± 7.42e + 00	-
f_{19}	-9.60e + 01 ± 3.34e + 00	-9.82e + 01 ± 2.14e + 00	-	-9.93e + 01 ± 1.48e + 00	-	-9.89e + 01 ± 1.43e + 00	-
f_{20}	-5.46e + 02 ± 1.91e - 01	-5.45e + 02 ± 2.23e - 01	+	-5.46e + 02 ± 2.07e - 01	-	-5.45e + 02 ± 2.27e - 01	+
f_{21}	5.97e + 01 ± 1.80e + 01	5.32e + 01 ± 1.36e + 01	-	5.19e + 01 ± 1.19e + 01	-	5.00e + 01 ± 1.04e + 01	-
f_{22}	-9.84e + 02 ± 1.50e + 01	-9.87e + 02 ± 1.44e + 01	-	-9.89e + 02 ± 1.15e + 01	=	-9.90e + 02 ± 1.11e + 01	-
f_{23}	7.94e + 00 ± 6.06e - 01	8.08e + 00 ± 5.71e - 01	+	7.89e + 00 ± 4.42e - 01	-	7.95e + 00 ± 5.39e - 01	=
f_{24}	3.69e + 02 ± 1.16e + 02	3.16e + 02 ± 6.29e + 01	-	2.72e + 02 ± 4.16e + 01	-	2.69e + 02 ± 5.46e + 01	-

[19] N. Hansen, "The CMA Evolution Strategy," 2011, <http://www.lri.fr/~hansen/cmaesintro.html>.

[20] Cyber Dyne Srl Home Page, <http://cyberdynesoft.it/>.

[21] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80-83, 1945.

TABLE III
AVERAGE FITNESS \pm STANDARD DEVIATION AND WILCOXON RANK-SUM TEST IN 40 DIMENSIONS (REFERENCE = 3SOME)

	3SOME	(1+1)-CMA-ES-3SOME		μ DE-3SOME		RIS-3SOME	
f_1	$7.95e + 01 \pm 2.56e - 14$	$7.95e + 01 \pm 0.00e + 00$	-	$7.95e + 01 \pm 2.18e - 14$	=	$7.95e + 01 \pm 2.62e - 14$	=
f_2	$-2.10e + 02 \pm 3.28e - 14$	$-2.10e + 02 \pm 1.45e - 14$	-	$-2.10e + 02 \pm 3.13e - 14$	=	$-2.10e + 02 \pm 3.44e - 14$	=
f_3	$-4.54e + 02 \pm 3.44e + 00$	$-4.16e + 02 \pm 1.21e + 01$	+	$-4.39e + 02 \pm 4.92e + 00$	+	$-4.15e + 02 \pm 1.05e + 01$	+
f_4	$-4.51e + 02 \pm 4.06e + 00$	$-4.08e + 02 \pm 1.43e + 01$	+	$-4.31e + 02 \pm 6.85e + 00$	+	$-4.05e + 02 \pm 1.43e + 01$	+
f_5	$5.63e + 01 \pm 1.78e + 02$	$1.23e + 01 \pm 1.05e + 02$	-	$7.67e + 01 \pm 1.42e + 02$	+	$-9.21e + 00 \pm 8.58e - 13$	-
f_6	$3.59e + 01 \pm 9.31e - 07$	$3.59e + 01 \pm 0.00e + 00$	-	$3.59e + 01 \pm 4.52e - 06$	=	$3.59e + 01 \pm 8.18e - 08$	=
f_7	$2.10e + 02 \pm 6.39e + 01$	$1.25e + 02 \pm 1.06e + 01$	-	$1.63e + 02 \pm 3.21e + 01$	-	$1.76e + 02 \pm 3.20e + 01$	-
f_8	$1.53e + 02 \pm 1.69e + 01$	$1.53e + 02 \pm 3.53e + 00$	+	$1.52e + 02 \pm 1.36e + 01$	=	$1.49e + 02 \pm 5.23e - 01$	=
f_9	$1.25e + 02 \pm 1.53e + 00$	$1.31e + 02 \pm 3.32e + 00$	+	$1.25e + 02 \pm 1.07e + 00$	=	$1.25e + 02 \pm 1.48e + 00$	+
f_{10}	$1.95e + 05 \pm 1.40e + 06$	$9.23e + 01 \pm 6.56e + 01$	-	$1.62e + 05 \pm 4.48e + 05$	-	$8.93e + 02 \pm 2.94e + 02$	=
f_{11}	$3.80e + 02 \pm 6.30e + 01$	$8.16e + 01 \pm 5.16e + 01$	-	$3.13e + 02 \pm 5.82e + 01$	-	$3.24e + 02 \pm 4.72e + 01$	-
f_{12}	$-6.11e + 02 \pm 8.98e + 00$	$-6.11e + 02 \pm 8.50e + 00$	=	$-6.16e + 02 \pm 6.47e + 00$	-	$-6.15e + 02 \pm 6.46e + 00$	-
f_{13}	$4.19e + 01 \pm 1.28e + 01$	$4.07e + 01 \pm 1.35e + 01$	=	$4.39e + 01 \pm 1.06e + 01$	+	$4.20e + 01 \pm 1.05e + 01$	=
f_{14}	$-5.23e + 01 \pm 7.18e - 05$	$-5.23e + 01 \pm 6.31e - 07$	-	$-5.22e + 01 \pm 1.88e + 00$	+	$-5.23e + 01 \pm 5.67e - 05$	=
f_{15}	$2.06e + 03 \pm 4.04e + 02$	$1.75e + 03 \pm 2.12e + 02$	-	$1.37e + 03 \pm 1.21e + 02$	-	$1.45e + 03 \pm 1.57e + 02$	-
f_{16}	$8.87e + 01 \pm 5.44e + 00$	$8.95e + 01 \pm 5.53e + 00$	=	$8.72e + 01 \pm 5.46e + 00$	-	$8.45e + 01 \pm 4.71e + 00$	-
f_{17}	$-5.52e + 00 \pm 3.25e + 00$	$-9.42e + 00 \pm 1.41e + 00$	-	$-1.07e + 01 \pm 1.53e + 00$	-	$-1.05e + 01 \pm 1.27e + 00$	-
f_{18}	$2.56e + 01 \pm 1.47e + 01$	$1.14e + 01 \pm 5.42e + 00$	-	$7.91e + 00 \pm 5.93e + 00$	-	$7.21e + 00 \pm 4.66e + 00$	-
f_{19}	$-9.33e + 01 \pm 3.68e + 00$	$-9.54e + 01 \pm 2.38e + 00$	-	$-9.80e + 01 \pm 2.12e + 00$	-	$-9.68e + 01 \pm 1.94e + 00$	-
f_{20}	$-5.46e + 02 \pm 1.28e - 01$	$-5.45e + 02 \pm 1.61e - 01$	+	$-5.46e + 02 \pm 1.97e - 01$	+	$-5.45e + 02 \pm 1.59e - 01$	+
f_{21}	$5.28e + 01 \pm 1.62e + 01$	$4.96e + 01 \pm 1.22e + 01$	-	$4.44e + 01 \pm 6.26e + 00$	-	$4.54e + 01 \pm 8.39e + 00$	-
f_{22}	$-9.85e + 02 \pm 1.31e + 01$	$-9.88e + 02 \pm 8.83e + 00$	-	$-9.87e + 02 \pm 7.35e + 00$	=	$-9.90e + 02 \pm 9.58e + 00$	-
f_{23}	$8.10e + 00 \pm 5.26e - 01$	$8.42e + 00 \pm 6.70e - 01$	+	$8.10e + 00 \pm 5.52e - 01$	=	$8.16e + 00 \pm 5.76e - 01$	=
f_{24}	$9.44e + 02 \pm 2.79e + 02$	$6.49e + 02 \pm 1.43e + 02$	-	$5.17e + 02 \pm 7.50e + 01$	-	$5.80e + 02 \pm 1.18e + 02$	-

TABLE IV
AVERAGE FITNESS \pm STANDARD DEVIATION AND WILCOXON RANK-SUM TEST IN 100 DIMENSIONS (REFERENCE = 3SOME)

	3SOME	(1+1)-CMA-ES-3SOME		μ DE-3SOME		RIS-3SOME	
f_1	$7.95e + 01 \pm 3.29e - 14$	$7.95e + 01 \pm 0.00e + 00$	-	$7.95e + 01 \pm 3.96e - 14$	=	$7.95e + 01 \pm 4.10e - 14$	=
f_2	$-2.10e + 02 \pm 5.69e - 14$	$-2.10e + 02 \pm 2.45e - 14$	-	$-2.10e + 02 \pm 5.49e - 14$	=	$-2.10e + 02 \pm 5.72e - 14$	=
f_3	$-4.39e + 02 \pm 7.28e + 00$	$-3.34e + 02 \pm 2.37e + 01$	+	$-3.64e + 02 \pm 1.64e + 01$	+	$-3.34e + 02 \pm 2.09e + 01$	+
f_4	$-4.27e + 02 \pm 8.70e + 00$	$-3.07e + 02 \pm 2.66e + 01$	+	$-3.36e + 02 \pm 2.14e + 01$	+	$-2.96e + 02 \pm 2.48e + 01$	+
f_5	$7.40e + 00 \pm 1.65e + 02$	$-8.18e + 00 \pm 3.02e - 01$	-	$2.43e + 02 \pm 4.90e + 02$	+	$-9.21e + 00 \pm 4.28e - 12$	-
f_6	$3.59e + 01 \pm 8.86e - 08$	$3.59e + 01 \pm 1.48e - 03$	+	$1.33e + 04 \pm 1.32e + 05$	+	$3.59e + 01 \pm 3.89e - 08$	=
f_7	$5.97e + 02 \pm 2.83e + 02$	$2.90e + 02 \pm 7.06e + 01$	-	$3.90e + 02 \pm 1.06e + 02$	-	$3.96e + 02 \pm 1.03e + 02$	-
f_8	$1.83e + 02 \pm 3.31e + 01$	$2.13e + 02 \pm 1.99e + 01$	+	$1.84e + 02 \pm 4.08e + 01$	=	$1.89e + 02 \pm 4.24e + 01$	=
f_9	$1.76e + 02 \pm 1.36e + 01$	$1.89e + 02 \pm 1.36e + 01$	+	$1.78e + 02 \pm 2.38e + 01$	=	$1.78e + 02 \pm 1.34e + 01$	+
f_{10}	$2.68e + 03 \pm 6.96e + 02$	$1.59e + 03 \pm 4.52e + 02$	-	$6.99e + 04 \pm 5.83e + 05$	+	$2.97e + 03 \pm 6.44e + 02$	+
f_{11}	$3.83e + 02 \pm 8.22e + 01$	$7.63e + 01 \pm 5.83e - 03$	-	$6.82e + 02 \pm 1.21e + 02$	+	$7.26e + 02 \pm 8.46e + 01$	+
f_{12}	$-6.09e + 02 \pm 1.83e + 01$	$-6.12e + 02 \pm 1.65e + 01$	=	$-6.17e + 02 \pm 6.31e + 00$	=	$-6.17e + 02 \pm 9.12e + 00$	-
f_{13}	$3.35e + 01 \pm 4.87e + 00$	$3.30e + 01 \pm 4.63e + 00$	=	$3.64e + 01 \pm 5.08e + 00$	+	$3.61e + 01 \pm 4.80e + 00$	+
f_{14}	$-5.23e + 01 \pm 5.47e - 05$	$-5.23e + 01 \pm 2.08e - 06$	-	$-5.23e + 01 \pm 5.56e - 05$	-	$-5.23e + 01 \pm 5.89e - 05$	=
f_{15}	$4.53e + 03 \pm 5.89e + 02$	$3.65e + 03 \pm 4.54e + 02$	-	$2.26e + 03 \pm 2.81e + 02$	-	$2.49e + 03 \pm 4.94e + 02$	-
f_{16}	$9.51e + 01 \pm 6.11e + 00$	$9.90e + 01 \pm 4.38e + 00$	+	$9.28e + 01 \pm 8.46e + 00$	-	$8.94e + 01 \pm 3.94e + 00$	-
f_{17}	$-2.63e - 02 \pm 3.97e + 00$	$-6.72e + 00 \pm 1.91e + 00$	-	$-8.67e + 00 \pm 1.82e + 00$	-	$-7.28e + 00 \pm 1.79e + 00$	-
f_{18}	$4.55e + 01 \pm 1.54e + 01$	$2.34e + 01 \pm 7.42e + 00$	-	$1.44e + 01 \pm 7.27e + 00$	-	$1.91e + 01 \pm 6.52e + 00$	-
f_{19}	$-9.08e + 01 \pm 3.39e + 00$	$-8.89e + 01 \pm 3.68e + 00$	+	$-9.39e + 01 \pm 2.07e + 00$	-	$-9.27e + 01 \pm 3.76e + 00$	-
f_{20}	$-5.46e + 02 \pm 9.61e - 02$	$-5.45e + 02 \pm 1.04e - 01$	+	$-5.45e + 02 \pm 1.01e - 01$	+	$-5.45e + 02 \pm 9.12e - 02$	+
f_{21}	$5.19e + 01 \pm 1.21e + 01$	$4.95e + 01 \pm 9.21e + 00$	-	$4.81e + 01 \pm 6.53e + 00$	-	$4.94e + 01 \pm 8.84e + 00$	=
f_{22}	$-9.82e + 02 \pm 1.47e + 01$	$-9.84e + 02 \pm 1.38e + 01$	-	$-9.87e + 02 \pm 1.04e + 01$	-	$-9.87e + 02 \pm 1.06e + 01$	=
f_{23}	$8.21e + 00 \pm 4.93e - 01$	$8.75e + 00 \pm 5.79e - 01$	+	$8.30e + 00 \pm 5.67e - 01$	=	$8.24e + 00 \pm 4.56e - 01$	=
f_{24}	$2.79e + 03 \pm 4.75e + 02$	$1.86e + 03 \pm 2.86e + 02$	-	$1.30e + 03 \pm 1.44e + 02$	-	$1.82e + 03 \pm 3.01e + 02$	-