The Importance of Being Structured: a Comparative Study on Multi Stage Memetic Approaches

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
‡INCAS[3] - Innovation Centre for Advanced Sensors and Sensor Systems
P.O. Box 797, 9400 AT Assen, The Netherlands
Email: giovanniiacca@incas3.eu

***Abstract*—Memetic Computing (MC) is a discipline which studies optimization algorithms and sees them as structures of operators, the memes. Although the choice of memes is crucial for an effective algorithmic design, special attention should be paid also to the coordination amongst the memes. This paper presents a study on a basic sequential structure, namely Three Stage Optimal Memetic Exploration (3SOME). The 3SOME algorithm is composed of three operators (or memes) which progressively perturbs a single solution. The first meme, long distance exploration is characterized by a long search radius and is supposed to detect promising areas of the decision space. The second meme, middle distance exploration, is characterized by a moderate search radius and is supposed to focus the search in the the most promising basins of attraction. The third meme, short distance exploration, is characterized by a short search radius and had the role of performing the local optimal search in the areas detected by the first two memes. To assess the importance of the structure within MC we compare the performance of 3SOME with two modified versions of it over two complete benchmarks. In both cases, while retaining the 3SOME structure, we replace one of the three original components (short distance exploration) with an alternative deterministic local search, respectively Rosenbrock and Powell methods. Numerical results show that, regardless of the choice of the specific memes, as far as the 3SOME structure contains memes which perform long, middle, and short distance explorations a similar performance is achieved. These results remark that besides the intuitive finding that a proper choice of operators is fundamental for the algorithmic success, the structure composing them also plays a crucial role.**

I. Introduction

Memetic Computing (MC) is probably one of the most prominent emerging trends in modern Computational Intelligence Optimization. MC approaches are dynamic computational structures composed of multiple interacting *memes*, defined as *units of information encoded in computational representations for the purpose of problem solving* [1], coordinated according to some logic. A cornerstone example of Memetic Computing is represented by Memetic Algorithms (MAs), which are hybrid meta-heuristics composed of an evolutionary framework, acting as global searcher, and one or more local search components activated within its generation cycle [2]. However, MC extends the rather rigid concept of MAs as it entails the definition of hybrid frameworks arbitrarily structured and not limited to the combination of an evolutionary framework and one or more local searchers. Due to their robustness and versatility, memetic approaches have been successfully applied in many cases, ranging from biology to engineering design. For a broad survey of applications of Memetic Computing, the interested reader is referred to [3] and [4].

A crucial issue in MC is the coordination among memes, that is the way (when and how) each meme is activated. In particular, related to MAs, several coordination schemes have been proposed in the last two decades. For example, a rather simple strategy to control the activation of local search, called "partial Lamarckianism" [5] consists in randomly applying the local search with some probability. Another possible scheme, proposed in [6] and [7], consists in classifying the solutions processed by the evolutionary framework according to their fitness. In each generation a different set of local-search parameters is then associated to each set of solutions, so that the "intensity" of the local search, i.e. the exploitation pressure applied on a given solution to be locally improved, is tuned. More complex approaches make use of a learning process on the activation of memes, see [8] and [9], multiple subpopulations [10], or clustering [11]. Other MAs "interweave" the local search within the population-based engine, see e.g. the hill climbing crossover applied in [12] and [13]. A hybrid coordination system which combines a learning component of the landscape features with a fuzzy decision maker is proposed in [14]. All these schemes are based on the idea that an effective MA has to guarantee a proper balance between exploitation and exploration, see [15], [16], [17] and [18].

However, if the study of coordination schemes in MAs is nowadays a quite mature research area, see for example the excellent tutorial [19], the taxonomy proposed in [20] and the theoretical analysis performed in [21], almost no work has been done on the structures used in MC. Although some of the ideas successfully applied in MAs can be extended also to modern MC, this research line still presents many open issues. The reason for that is twofold. First of all, being MC a much broader (and more recent) area than MAs, it is not trivial to perform a general, conceptual analysis of all possible coordination schemes, and their influence on the algorithmic performance. In other words, the definition of a *grammar* of structures (whose syntactic elements are memes) is yet to come. In addition to that, since the algorithmic design process in MC is characterized by a higher degree of freedom, compared to MAs, most of the design focuses more on the choice of components than on the coordination schemes itself. This tendency turns sometimes into a poor understanding of the algorithmic behaviour, and misleads the interpretation of results. A first attempt to analyse the rationales behind the choice of the algorithmic structure in MC has been done recently in [22], where 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 MC, based on building up the algorithm with a bottom-up approach. 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 [22] a novel MC approach is introduced, named Three Stage Optimal Memetic Exploration (3SOME). Despite its conceptual simplicity, 3SOME has proven to efficiently handle different kinds of optimization problems, also in high dimensionality values. Nevertheless, a possible skepticism about such a structure is that the extremely good performance of it is mainly due to the presence of a strongly exploitative component, namely the short distance search operator. Although from an algorithmic design point of view the choice of this component is clearly justified, since it balances the exploration pressure of the other two memes, it is still interesting to further investigate the 3SOME framework in order to understand if the reason for its performance is indeed the exploitation provided by the short distance operator or rather the algorithmic structure itself. Based on this idea, the purpose of this paper is to examine the influence of the structure over the effect of memes using the 3SOME structure as a case of study. We propose two simple variants in which the short search distance operator is replaced by an alternative local search method. In both variants however, the coordination structure is kept constant. In this way we aim to prove that the robustness of a memetic framework is to be found not (only) in its component memes, but in its structure itself. The remainder of this paper is organized in the following way. Section II briefly introduces the basic structure of 3SOME. Section III describes the two proposed variants. Section IV displays the experimental testbed and numerical results related to comparison among 3SOME and the new variants. Finally, Section V gives the conclusion of this work.

II. Three Stage Optimal Memetic Exploration

In the following, 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 [23], 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 [23]. 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

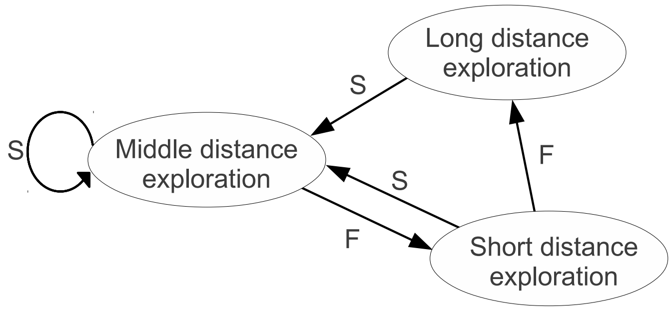

Fig. 1. Coordination scheme of 3SOME

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 [24], or the first local search algorithm of the Multiple Trajectory Search [25]. 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 mechanism is used also in the algorithms proposed in this paper.

Figure 1 shows the coordination scheme of the 3SOME components. Similar to a Finite State Machine, the algorithm is described as a composition of states, each one corresponding to a single operator (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, while the "S" and "F" represent success and failure, respectively, of the meme.

III. Memes and their Structure

In order to determine the contribution of the short distance exploration to the global algorithmic performance, and possibly understand the influence of the structure over the component memes, a simple but effective idea is to replace the short distance component with another local search, keeping the same coordination scheme. In this study, we propose two different variants, one using the Rosenbrock method and one employing the Powell algorithm. A brief description of these two classic optimization methods follows.

A. 3SOME with Rosenbrock Algorithm

In the first 3SOME variant, we refer to as 3SOME-Rosenbrock, the short distance component is replaced by the Rosenbrock optimization method [26]. The Rosenbrock algorithm is a classical deterministic local search which, under specific conditions, has been proved to always converge to a local optimum [27]. Like the Hooke-Jeeves method, at the beginning this method probes each of the n base directions, with an initial step size h. In case of success, the step size is increased of a factor α, otherwise it is decreased of a factor β and the opposite direction is tried. Once a success has been found and exploited in each base direction, the coordinate system is rotated towards the approximated gradient, the step size is reinitialized and the procedure is repeated, using the rotated coordinate system, until a stop criterion is met. In our tests, we used as stop criterion a threshold ε on the fitness function. The main flaw of this algorithm is related to the creation of the new rotated coordinate system: this operation, which is usually performed by means of orthogonalization procedures, is indeed computational expensive, and in some cases may even become numerically unstable.

With respect to figure 1, once the Rosenbrock component (in place of the short distance exploration) is activated, it is executed until the stop criterion is met, and then the coordination scheme activates either the long or middle distance exploration, respectively in case of failure and success.

B. 3SOME with Powell Algorithm

The "direction set" Powell algorithm is a derivative-free local searcher based on the idea of using a set of "non-interfering" directions to search and converge quickly to the local minimum of a function. The procedure, proposed by Powell in [28], makes use of a generically defined search method to minimize the function along a single direction. In each iteration, n separate minimizations are performed along n different directions. The latter are chosen so that the $(i+1)^{th}$ direction does not affect the minimization along the i^{th} one.

At the beginning of the algorithm, an $n \times n$ identity matrix is used as set of conjugated directions; subsequently, the matrix is updated so that its rank is always full (n linear independent vectors) and new search directions are probed. More specifically, at each step a new point P_n is obtained as linear combination of the conjugated directions, and the displacement vector $P_0 - P_n$ is computed, where P_0 is the initial point at the beginning of the iteration. If the displacement vector is linearly independent from the other $n-1$ directions, it replaces the direction which contributed most to the new direction (i.e. the one along which the fitness function showed the largest decrease), and a new minimization is performed along it. A new point is then generated, and the process is repeated until the fitness improvement is smaller than a fixed tolerance f_{tol}.

It is worthwhile to notice that the performance of this algorithm heavily depends on the specific search method used for performing the minimization along each direction. In this study we used the Golden Section Section (GSS). The resulting combination of the 3SOME structure and the Powell algorithm using GSS, called 3SOME-Powell, uses the same coordination shown in figure 1, where the Powell algorithm is used instead of the short distance exploration.

IV. Numerical Results

In order to evaluate in detail the structures presented in the previous section, we performed an extensive comparative study on two different benchmarks, namely the noiseless Black-Box Optimization Benchmark (BBOB) 2010 [29] and the benchmark used for the Special Session on Large-Scale Global Optimization at CEC 2010 [30], consisting respectively of 24 and 20 test functions. Both the two benchmarks comprehend test functions with different properties in terms of modality, separability, and ill-conditioning. This makes the experimental setup extremely heterogeneous and challenging from an optimization point of view. Moreover, to test the scalability of the original implementation of 3SOME and its variants we run the experiments in different dimensionalities, i.e. 10, 40 and 100 for the BBOB 2010 and 1000 for the CEC 2010 benchmark. Thus we considered $24 \times 3 + 20 = 92$ test functions in total. Each algorithm has been run for $5000 \times n$ fitness evaluations for each run in the case of BBOB 2010, and $3e6$ with the CEC 2010 test functions. 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 [31].

As for the parameter setting, 3SOME was executed using the parameters suggested in [22], 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 $k = 4$. For a fair comparison, the same values of α_e, δ and k were used also in 3SOME-Rosenbrock and 3SOME-Powell. Four additional parameters, i.e. initial step size $h = 0.1$, the threshold $\varepsilon = 1e-8$, the forward factor $\alpha = 2$ and the backward factor $\beta = 0.5$, were needed in 3SOME-Rosenbrock. Finally, 3SOME-Powell was executed with a fitness tolerance f_{tol} set equal to $1e-5$, while the Golden Section Search was applied with bounds $[-100, 100]$ and a budget of 20 fitness evaluations.

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 [32] was also applied, with a confidence level of 0.95. The symbols "=" and "+" ("-") indicate, respectively, a statistically equivalent performance and a better (worse) performance of original 3SOME compared with the algorithm in the column label.

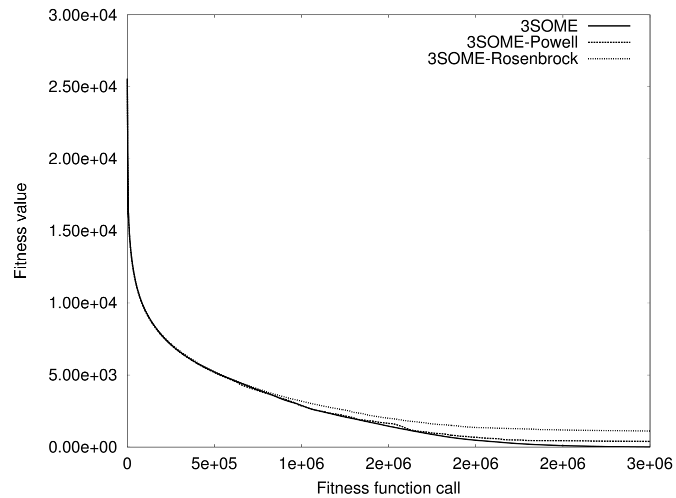

Fig. 2. Fitness trend of function f_2 from CEC 2010

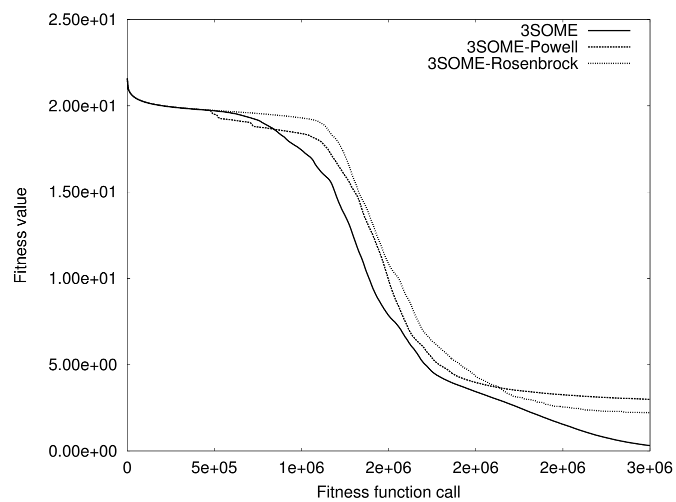

Fig. 3. Fitness trend of function f_3 from CEC 2010

Results show that the three algorithms have similar performances, especially on the BBOB 2010 in lower dimensions (10 and 40). As the problem dimension grows however, both on the BBOB 2010 (100) and on the CEC 2010 benchmark (1000), 3SOME outperforms its Powell-based variant, while 3SOME-Rosenbrock seems to offer a slightly better global behaviour. In general, the use of the Powell method appears beneficial especially on non-separable multimodal functions with weak or adequate global structure and low dimensionality, while it can be detrimental in large scale problems. On these problems, 3SOME-Rosenbrock show better performances: more specifically, it is slightly more effective on separable unimodal functions and non-separable problems in 100 and 1000 dimensions.

A careful analysis of the dynamic behaviour of the three

TABLE I

AVERAGE FITNESS $\pm$ STANDARD DEVIATION AND WILCOXON RANK-SUM TEST ON BBOB 2010 IN 10 DIMENSIONS (REFERENCE = 3SOME)

	3SOME	3SOME-Powell		3SOME-Rosenbrock	
f_1	$\mathbf{7.95e+01 \pm 1.21e-14}$	$7.95e+01 \pm 8.94e-05$	+	$\mathbf{7.95e+01 \pm 0.00e+00}$	-
f_2	$\mathbf{-2.10e+02 \pm 1.63e-14}$	$-1.86e+02 \pm 4.48e+01$	+	$-2.10e+02 \pm 4.17e-13$	+
f_3	$-4.61e+02 \pm 1.18e+00$	$-4.61e+02 \pm 5.03e-01$	-	$\mathbf{-4.62e+02 \pm 2.54e-01}$	-
f_4	$-4.60e+02 \pm 1.39e+00$	$-4.61e+02 \pm 8.10e-01$	-	$\mathbf{-4.62e+02 \pm 7.35e-01}$	-
f_5	$5.33e+00 \pm 2.91e+01$	$-8.56e+00 \pm 2.49e-01$	-	$\mathbf{-9.21e+00 \pm 6.35e-09}$	-
f_6	$8.25e+01 \pm 2.83e+02$	$4.66e+01 \pm 1.05e+01$	-	$\mathbf{3.63e+01 \pm 6.31e-01}$	-
f_7	$1.05e+02 \pm 1.23e+01$	$\mathbf{1.02e+02 \pm 5.88e+00}$	=	$1.03e+02 \pm 7.60e+00$	=
f_8	$\mathbf{1.49e+02 \pm 1.86e-01}$	$1.52e+02 \pm 2.47e+00$	+	$1.51e+02 \pm 2.09e+00$	+
f_9	$\mathbf{1.25e+02 \pm 1.69e+00}$	$1.47e+02 \pm 4.28e+01$	+	$1.26e+02 \pm 1.10e+01$	+
f_{10}	$3.95e+03 \pm 2.63e+04$	$6.42e+03 \pm 5.41e+03$	+	$\mathbf{2.01e+02 \pm 2.13e+02}$	-
f_{11}	$1.57e+02 \pm 3.36e+01$	$\mathbf{1.01e+02 \pm 9.08e+00}$	-	$1.66e+02 \pm 3.20e+01$	=
f_{12}	$\mathbf{-6.12e+02 \pm 1.33e+01}$	$-2.40e+02 \pm 3.93e+02$	+	$-6.12e+02 \pm 1.50e+01$	=
f_{13}	$\mathbf{4.26e+01 \pm 1.28e+01}$	$4.62e+01 \pm 1.29e+01$	+	$4.28e+01 \pm 1.17e+01$	=
f_{14}	$-5.23e+01 \pm 3.05e-05$	$-5.23e+01 \pm 2.06e-03$	+	$\mathbf{-5.23e+01 \pm 6.52e-05}$	-
f_{15}	$1.10e+03 \pm 6.38e+01$	$\mathbf{1.06e+03 \pm 2.23e+01}$	-	$1.06e+03 \pm 2.51e+01$	-
f_{16}	$7.97e+01 \pm 4.63e+00$	$\mathbf{7.58e+01 \pm 2.03e+00}$	-	$7.67e+01 \pm 3.80e+00$	-
f_{17}	$-1.03e+01 \pm 6.57e+00$	$\mathbf{-1.51e+01 \pm 9.11e-01}$	-	$-2.32e+00 \pm 1.09e+01$	+
f_{18}	$5.80e+00 \pm 2.56e+01$	$\mathbf{-1.17e+01 \pm 3.09e+00}$	-	$3.96e+01 \pm 5.28e+01$	+
f_{19}	$-9.80e+01 \pm 2.98e+00$	$\mathbf{-9.93e+01 \pm 7.62e-01}$	-	$-9.44e+01 \pm 4.24e+00$	+
f_{20}	$-5.46e+02 \pm 2.59e-01$	$\mathbf{-5.46e+02 \pm 2.17e-01}$	-	$-5.46e+02 \pm 2.44e-01$	-
f_{21}	$5.36e+01 \pm 1.34e+01$	$\mathbf{4.46e+01 \pm 3.62e+00}$	-	$4.51e+01 \pm 4.16e+00$	-
f_{22}	$-9.88e+02 \pm 1.55e+01$	$\mathbf{-9.96e+02 \pm 4.73e+00}$	-	$-9.96e+02 \pm 7.80e+00$	-
f_{23}	$7.86e+00 \pm 4.95e-01$	$7.92e+00 \pm 2.46e-01$	=	$\mathbf{7.54e+00 \pm 2.98e-01}$	-
f_{24}	$1.92e+02 \pm 4.46e+01$	$\mathbf{1.56e+02 \pm 1.43e+01}$	-	$1.67e+02 \pm 2.06e+01$	-

TABLE II

AVERAGE FITNESS $\pm$ STANDARD DEVIATION AND WILCOXON RANK-SUM TEST ON BBOB 2010 IN 40 DIMENSIONS (REFERENCE = 3SOME)

	3SOME	3SOME-Powell		3SOME-Rosenbrock	
f_1	$\mathbf{7.95e+01 \pm 2.56e-14}$	$7.95e+01 \pm 2.89e-03$	+	$\mathbf{7.95e+01 \pm 1.99e-14}$	=
f_2	$\mathbf{-2.10e+02 \pm 3.28e-14}$	$-8.97e+01 \pm 1.09e+02$	+	$-2.10e+02 \pm 2.07e-13$	+
f_3	$-4.54e+02 \pm 3.44e+00$	$-4.57e+02 \pm 1.51e+00$	-	$\mathbf{-4.59e+02 \pm 1.86e+00}$	-
f_4	$-4.51e+02 \pm 4.06e+00$	$-4.52e+02 \pm 2.64e+00$	-	$\mathbf{-4.57e+02 \pm 2.56e+00}$	-
f_5	$5.63e+01 \pm 1.78e+02$	$-5.43e+00 \pm 7.20e-01$	-	$\mathbf{-9.21e+00 \pm 6.81e-12}$	-
f_6	$\mathbf{3.59e+01 \pm 9.31e-07}$	$1.13e+02 \pm 1.04e+02$	+	$3.66e+01 \pm 1.07e+00$	+
f_7	$\mathbf{2.10e+02 \pm 6.39e+01}$	$2.30e+02 \pm 5.20e+01$	+	$2.26e+02 \pm 5.19e+01$	+
f_8	$\mathbf{1.53e+02 \pm 1.69e+01}$	$2.31e+02 \pm 3.64e+01$	+	$1.83e+02 \pm 3.56e+01$	+
f_9	$\mathbf{1.25e+02 \pm 1.53e+00}$	$1.77e+02 \pm 3.46e+01$	+	$1.27e+02 \pm 1.54e+00$	+
f_{10}	$1.95e+05 \pm 1.40e+06$	$2.62e+04 \pm 1.25e+04$	-	$\mathbf{7.73e+02 \pm 3.58e+02}$	-
f_{11}	$3.80e+02 \pm 6.30e+01$	$\mathbf{2.26e+02 \pm 3.43e+01}$	-	$3.78e+02 \pm 6.60e+01$	=
f_{12}	$\mathbf{-6.11e+02 \pm 8.98e+00}$	$5.07e+03 \pm 7.43e+03$	+	$-6.10e+02 \pm 8.73e+00$	=
f_{13}	$\mathbf{4.19e+01 \pm 1.28e+01}$	$5.63e+01 \pm 1.56e+01$	+	$4.32e+01 \pm 1.38e+01$	=
f_{14}	$-5.23e+01 \pm 7.18e-05$	$-5.23e+01 \pm 4.10e-03$	+	$\mathbf{-5.23e+01 \pm 2.29e-05}$	-
f_{15}	$2.06e+03 \pm 4.04e+02$	$\mathbf{1.69e+03 \pm 1.82e+02}$	-	$1.80e+03 \pm 1.59e+02$	-
f_{16}	$\mathbf{8.87e+01 \pm 5.44e+00}$	$9.17e+01 \pm 4.68e+00$	+	$9.12e+01 \pm 4.82e+00$	+
f_{17}	$-5.52e+00 \pm 3.25e+00$	$\mathbf{-8.89e+00 \pm 1.52e+00}$	-	$-5.94e+00 \pm 2.91e+00$	=
f_{18}	$2.56e+01 \pm 1.47e+01$	$\mathbf{1.34e+01 \pm 4.92e+00}$	-	$2.55e+01 \pm 1.33e+01$	=
f_{19}	$\mathbf{-9.33e+01 \pm 3.68e+00}$	$-9.32e+01 \pm 1.28e+00$	=	$-8.96e+01 \pm 4.60e+00$	+
f_{20}	$-5.46e+02 \pm 1.28e-01$	$\mathbf{-5.46e+02 \pm 1.08e-01}$	-	$-5.46e+02 \pm 1.19e-01$	-
f_{21}	$5.28e+01 \pm 1.62e+01$	$\mathbf{4.79e+01 \pm 9.75e+00}$	=	$4.94e+01 \pm 1.19e+01$	=
f_{22}	$-9.85e+02 \pm 1.31e+01$	$-9.83e+02 \pm 1.48e+01$	+	$\mathbf{-9.87e+02 \pm 1.07e+01}$	=
f_{23}	$\mathbf{8.10e+00 \pm 5.26e-01}$	$9.38e+00 \pm 4.72e-01$	+	$8.70e+00 \pm 6.40e-01$	+
f_{24}	$9.44e+02 \pm 2.79e+02$	$\mathbf{6.97e+02 \pm 1.18e+02}$	-	$8.06e+02 \pm 1.18e+02$	-

TABLE III
AVERAGE FITNESS ± STANDARD DEVIATION AND WILCOXON RANK-SUM TEST ON BBOB 2010 IN 100 DIMENSIONS (REFERENCE = 3SOME)

	3SOME	3SOME-Powell		3SOME-Rosenbrock	
f_1	**7.95e+01 ± 3.29e−14**	7.95e+01 ± 7.05e−03	+	7.95e+01 ± 2.86e−14	+
f_2	**−2.10e+02 ± 5.69e−14**	4.80e+02 ± 5.03e+02	+	−2.10e+02 ± 1.75e−13	+
f_3	−4.39e+02 ± 7.28e+00	−4.43e+02 ± 4.04e+00	-	**−4.45e+02 ± 5.80e+00**	-
f_4	−4.27e+02 ± 8.70e+00	−4.28e+02 ± 5.07e+00	=	**−4.36e+02 ± 7.52e+00**	-
f_5	7.40e+00 ± 1.65e+02	3.31e+00 ± 1.71e+00	-	**−9.21e+00 ± 6.33e−12**	-
f_6	**3.59e+01 ± 8.86e−08**	4.43e+02 ± 2.11e+02	+	4.31e+01 ± 5.29e+00	+
f_7	**5.97e+02 ± 2.83e+02**	9.36e+02 ± 2.65e+02	+	8.45e+02 ± 2.19e+02	+
f_8	**1.83e+02 ± 3.31e+01**	3.19e+02 ± 6.10e+01	+	2.31e+02 ± 4.62e+01	+
f_9	**1.76e+02 ± 1.36e+01**	2.53e+02 ± 4.53e+01	+	1.77e+02 ± 1.76e+01	=
f_{10}	2.68e+03 ± 6.96e+02	6.31e+04 ± 1.78e+04	+	**2.10e+03 ± 5.99e+02**	-
f_{11}	3.83e+02 ± 8.22e+01	**3.45e+02 ± 7.84e+01**	-	5.51e+02 ± 1.14e+02	+
f_{12}	−6.09e+02 ± 1.83e+01	6.00e+03 ± 4.95e+03	+	**−6.12e+02 ± 1.64e+01**	=
f_{13}	3.35e+01 ± 4.87e+00	5.79e+01 ± 9.02e+00	+	**3.32e+01 ± 4.16e+00**	=
f_{14}	−5.23e+01 ± 5.47e−05	−5.23e+01 ± 4.90e−03	+	**−5.23e+01 ± 1.67e−05**	-
f_{15}	4.53e+03 ± 5.89e+02	**4.08e+03 ± 5.10e+02**	-	4.20e+03 ± 5.26e+02	-
f_{16}	**9.51e+01 ± 6.11e+00**	1.03e+02 ± 4.79e+00	+	1.02e+02 ± 5.54e+00	+
f_{17}	−2.63e−02 ± 3.97e+00	**−2.60e+00 ± 2.97e+00**	-	−1.32e+00 ± 3.44e+00	-
f_{18}	4.55e+01 ± 1.54e+01	**3.57e+01 ± 1.04e+01**	-	4.02e+01 ± 1.27e+01	-
f_{19}	**−9.08e+01 ± 3.39e+00**	−8.80e+01 ± 2.19e+00	+	−8.90e+01 ± 3.92e+00	+
f_{20}	−5.46e+02 ± 9.61e−02	**−5.46e+02 ± 7.49e−02**	-	−5.46e+02 ± 1.17e−01	=
f_{21}	5.19e+01 ± 1.21e+01	5.17e+01 ± 1.29e+01	=	**5.02e+01 ± 1.11e+01**	=
f_{22}	−9.82e+02 ± 1.47e+01	−9.80e+02 ± 1.54e+01	+	**−9.84e+02 ± 1.30e+01**	=
f_{23}	**8.21e+00 ± 4.93e−01**	1.02e+01 ± 4.95e−01	+	9.13e+00 ± 5.39e−01	+
f_{24}	2.79e+03 ± 4.75e+02	**2.65e+03 ± 2.50e+02**	-	2.68e+03 ± 3.15e+02	-

TABLE IV
AVERAGE FITNESS ± STANDARD DEVIATION AND WILCOXON RANK-SUM TEST ON CEC 2010 IN 1000 DIMENSIONS (REFERENCE = 3SOME)

	3SOME	3SOME-Powell		3SOME-Rosenbrock	
f_1	8.81e−03 ± 1.72e−02	7.12e+07 ± 1.59e+07	+	**0.00e+00 ± 0.00e+00**	-
f_2	**1.48e+01 ± 2.05e+01**	3.97e+02 ± 2.73e+01	+	1.16e+03 ± 2.11e+02	+
f_3	**3.36e−01 ± 3.36e−01**	3.01e+00 ± 1.32e−01	+	2.23e+00 ± 2.29e−01	+
f_4	8.65e+12 ± 2.50e+12	9.40e+12 ± 2.99e+12	=	**8.47e+12 ± 2.56e+12**	=
f_5	6.89e+08 ± 1.11e+08	**6.18e+08 ± 1.25e+08**	-	6.54e+08 ± 1.19e+08	=
f_6	1.98e+07 ± 8.83e+04	1.98e+07 ± 1.06e+05	=	**1.98e+07 ± 1.31e+05**	-
f_7	**1.50e+09 ± 3.67e+08**	1.51e+09 ± 3.91e+08	=	1.54e+09 ± 4.38e+08	=
f_8	3.65e+08 ± 1.47e+09	**2.69e+08 ± 1.04e+09**	=	2.94e+08 ± 1.27e+09	-
f_9	3.76e+08 ± 7.12e+07	4.67e+08 ± 4.37e+07	+	**5.42e+07 ± 1.09e+08**	-
f_{10}	**6.79e+03 ± 3.75e+02**	7.15e+03 ± 3.69e+02	+	8.19e+03 ± 4.36e+02	+
f_{11}	**1.99e+02 ± 5.33e−01**	2.08e+02 ± 6.47e−01	+	2.18e+02 ± 3.59e−01	+
f_{12}	1.53e+05 ± 6.80e+04	3.92e+05 ± 2.43e+04	+	**1.33e+04 ± 5.65e+04**	-
f_{13}	1.54e+04 ± 5.67e+03	4.37e+04 ± 9.84e+03	+	**1.23e+03 ± 6.24e+02**	-
f_{14}	1.19e+08 ± 2.71e+07	1.27e+09 ± 8.65e+07	+	**2.95e+07 ± 9.98e+06**	-
f_{15}	**1.38e+04 ± 5.64e+02**	1.42e+04 ± 5.31e+02	+	1.39e+04 ± 5.20e+02	=
f_{16}	**3.71e+02 ± 7.75e+01**	4.18e+02 ± 3.63e+00	+	3.97e+02 ± 3.16e−01	+
f_{17}	2.85e+05 ± 2.41e+05	9.27e+05 ± 3.92e+04	+	**1.51e+05 ± 3.15e+05**	-
f_{18}	2.71e+04 ± 1.43e+04	5.64e+05 ± 1.45e+05	+	**2.22e+03 ± 8.48e+02**	-
f_{19}	1.47e+05 ± 2.19e+04	1.87e+05 ± 2.86e+04	+	**9.20e+04 ± 1.69e+04**	-
f_{20}	1.14e+03 ± 1.40e+02	5.55e+05 ± 1.64e+05	+	**9.22e+02 ± 4.21e+02**	-

algorithms with different problem dimensionalities was also performed. Indicating with "L", "M" and "S", the long, middle and short distance exploration (or Powell/Rosenbrock) operators, respectively, Tables V-VII display the memes activation, in terms of percentage of the total budget consumed by each meme, averaged over 100 runs on a subset of the BBOB 2010 test functions in 10, 40 and 100 dimensions. In order to assess if the algorithmic dynamics depends on the optimization problem, the subset was chosen selecting the first functions of each of the five subgroups of the BBOB 2010 [29], which differ in terms of separability, multi-modality, and ill-conditioning. It can be seen that, for each algorithm, the coordination scheme scales up nicely with the problem dimensionality, since the activation percentages seems to be almost constant (apart from natural stochastic fluctuations) as the number of dimensions grows. This behaviour is likely a consequence of the serial nature of the 3SOME structure (and its variants), which guarantees the same budget allocation regardless the dimensionality. On the other hand, the comparison among the three algorithms shows that their behaviour is very similar (see f_1, f_{10} apart from 3SOME-Powell in 10 dimensions, and f_{15}). Two interesting exceptions are f_6 and f_{20}, where the three algorithms show different trends, especially in larger dimensionalities. In particular, while 3SOME and 3SOME-Rosenbrock seem to have similar dynamics, 3SOME-Powell uses a lager budget in the long exploration stage. A possible explanation of this phenomenon is that on some peculiar non-separable multimodal landscapes the short distance search and the Rosenbrock method tend to consume more budget than the Powell algorithm, thus guaranteeing a better budget balance.

TABLE V
MEMES ACTIVATION ON BBOB 2010 IN 10 DIMENSIONS

	3SOME	3SOME-Powell	3SOME-Rosenbrock
f_1	L = 90.85%	L = 97.36%	L = 90.53%
	M = 0.91%	M = 1.04%	M = 1.34%
	S = 8.24%	S = 1.6%	S = 8.13%
f_6	L = 14.19%	L = 83.97%	L = 0.005%
	M = 2.59%	M = 4.88%	M = 3.593%
	S = 83.22%	S = 11.15%	S = 96.402%
f_{10}	L = 0.01%	L = 23.24%	L = 0.008%
	M = 2.15%	M = 5.33%	M = 1.624%
	S = 97.84%	S = 71.43%	S = 98.368%
f_{15}	L = 74.77%	L = 82.22%	L = 80.01%
	M = 0.95%	M = 1.84%	M = 1.64%
	S = 24.28%	S = 15.94%	S = 18.35%
f_{20}	L = 75.36%	L = 90.63%	L = 54.47%
	M = 1.25%	M = 2.17%	M = 2.38%
	S = 23.39%	S = 7.2%	S = 43.15%

TABLE VI
MEMES ACTIVATION ON BBOB 2010 IN 40 DIMENSIONS

	3SOME	3SOME-Powell	3SOME-Rosenbrock
f_1	L = 85.31%	L = 95.36%	L = 95.96%
	M =1.64%	M = 2.24%	M = 1.52%
	S = 13.05%	S = 2.4%	S = 2.52%
f_6	L = 29.08%	L = 80.67%	L = 0.002%
	M =3.74%	M = 3.93%	M = 2.624%
	S = 67.18%	S = 15.4%	S = 97.374%
f_{10}	L = 0.01%	L = 0.001%	L = 0.001%
	M = 4.1%	M = 4.72%	M = 2.032%
	S = 95.89%	S = 95.279%	S = 97.967%
f_{15}	L = 64.64%	L = 63.85%	L = 78.65%
	M = 1.99%	M = 7,69%	M = 3.01%
	S = 33.37%	S = 28.46%	S = 18.34%
f_{20}	L = 47.81%	L = 82.8%	L = 42.07%
	M = 2.45%	M = 5.2%	M = 3.36%
	S = 49.74%	S = 12%	S = 54.57%

The same analysis was also performed on the CEC 2010 benchmark, where we obtained similar results. For the sake of brevity, we don't report numerical results on memes activation. We report instead the fitness trends on four of the test functions from the benchmark, see Figures 2-5. It is interesting to notice that, except f_{11} and f_{16} where the three algorithms show remarkably different dynamics (with 3SOME outperforming its two variants), in the other cases the fitness trends are specular, see for example f_2 and f_3. Similar trends were obtained in the remaining 16 functions of the CEC 2010 benchmark. This result confirms that, despite the three algorithms use different exploitative components (with different

TABLE VII
MEMES ACTIVATION ON BBOB 2010 IN 100 DIMENSIONS

	3SOME	3SOME-Powell	3SOME-Rosenbrock
f_1	L = 85.31%	L = 95.36%	L = 95.96%
	M =1.64%	M = 2.24%	M = 1.52%
	S = 13.05%	S = 2.4%	S = 2.52%
f_6	L = 29.08%	L = 80.67%	L = 0.002%
	M =3.74%	M = 3.93%	M = 2.624%
	S = 67.18%	S = 15.4%	S = 97.374%
f_{10}	L = 0.01%	L = 0.001%	L = 0.001%
	M = 4.1%	M = 4.72%	M = 2.032%
	S = 95.89%	S = 95.279%	S = 97.967%
f_{15}	L = 64.64%	L = 63.85%	L = 78.65%
	M = 1.99%	M = 7,69%	M = 3.01%
	S = 33.37%	S = 28.46%	S = 18.34%
f_{20}	L = 47.81%	L = 82.8%	L = 42.07%
	M = 2.45%	M = 5.2%	M = 3.36%
	S = 49.74%	S = 12%	S = 54.57%

budget conditions), their global behaviour is almost completely ruled by the structure, that is the coordination scheme.

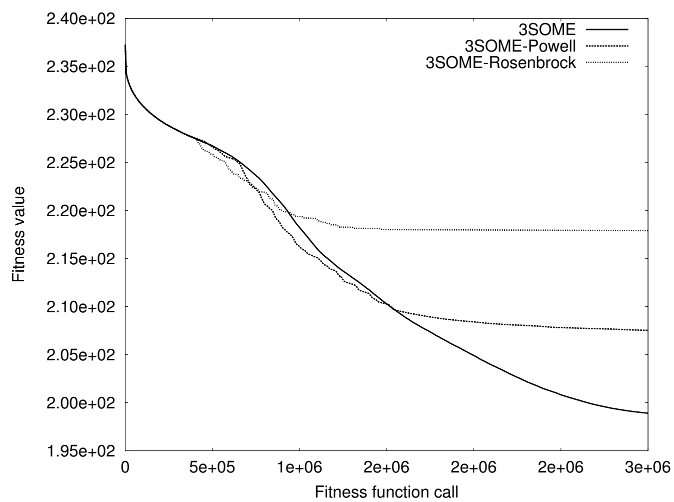

Fig. 4. Fitness trend of function f_{11} from CEC 2010

V. CONCLUSION

In this paper we have presented a comparative study focused on the structure of a Multi Stage Memetic Computing approach recently proposed in literature, namely 3SOME. The original framework was modified replacing its most exploitative component with two different classic local search methods, namely the Rosenbrock and Powell algorithms. Numerical results obtained on two broad sets of fitness functions show that, despite the perturbation, the algorithmic structures show a similar behaviour. This result allows us to conclude that the 3SOME structure is natural and efficient scheme for single solution progressive perturbation. Future studies will aim at extending the results in this work to Memetic Computing structures in general by conjecturing that structure is at least as important as memes. The demonstration of this concept could then be used as cornerstone in future developments and designs

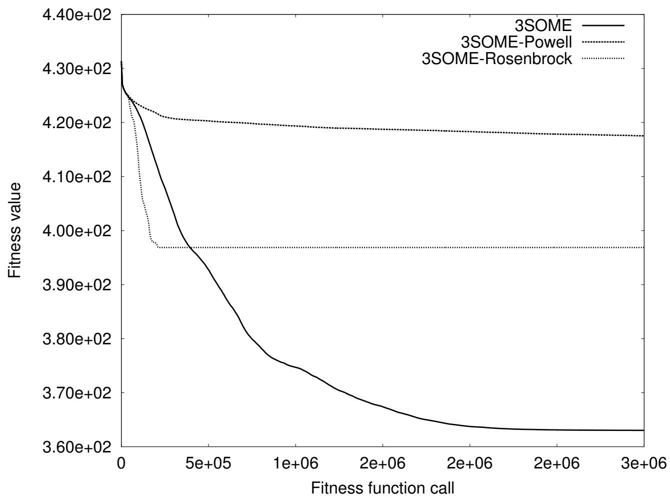

Fig. 5. Fitness trend of function f_{16} from CEC 2010

of Memetic Computing approaches, which in our opinion is the generation of an automatic algorithmic designer. Under such conditions, it is clear that it will be important that the system not only selects the most appropriate operators but also effectively combines them.

ACKNOWLEDGMENT

This research is supported by the Academy of Finland, Akatemiatutkija 130600 and Tutkijatohtori 140487. INCAS[3] 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] Y. S. Ong, M. H. Lim, and X. Chen, "Memetic computation-past, present and future," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 24–31, 2010.

[2] W. E. Hart, N. Krasnogor, and J. E. Smith, "Memetic evolutionary algorithms," in *Recent Advances in Memetic Algorithms*, W. E. Hart, N. Krasnogor, and J. E. Smith, Eds. Berlin, Germany: Springer, 2004, pp. 3–27.

[3] F. Neri, C. Cotta, and P. Moscato, *Handbook of Memetic Algorithms*, ser. Studies in Computational Intelligence. Springer, 2012, vol. 379.

[4] F. Neri and C. Cotta, "Memetic algorithms and memetic computing optimization: A literature review," *Swarm and Evolutionary Computation*, vol. 2, pp. 1–14, 2012.

[5] C. Houck, J. Joines, M. Kay, and J. Wilson, "Empirical investigation of the benefits of partial lamarckianism," *Evolutionary Computation*, vol. 5, no. 1, pp. 31–60, 1997.

[6] D. Molina, "Adaptive local search parameters for real-coded memetic algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2005, pp. 888–895.

[7] D. Molina, M. Lozano, and F. Herrera, "Ma-sw-chains: Memetic algorithm based on local search chains for large scale continuous global optimization," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1–8.

[8] Q. C. Nguyen, Y. S. Ong, and M. H. Lim, "A probabilistic memetic framework," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 604–623, 2009.

[9] Y. S. Ong and A. J. Keane, "Meta-lamarkian learning in memetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 99–110, 2004.

[10] H. Mühlenbein, M. Schomisch, and J. Born, "The parallel genetic algorithm as function optimizer," *Parallel Computing*, vol. 17, no. 6–7, pp. 619–632, 1991.

[11] G. Seront and H. Bersini, "A new ga-local search hybrid for continuous optimization based on multi-level single linkage clustering," in *GECCO*, 2000, pp. 90–95.

[12] T. Jones, "Crossover, macromutation, and population-based search," in *Sixth International Conference on Genetic Algorithms*, L. Eshelman, Ed. San Mateo CA: Morgan Kaufmann, 1995, pp. 73–80.

[13] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina, "Real-coded memetic algorithms with crossover hill-climbing," *Evolutionary Computation, Special Issue on Memetic Algorithms*, vol. 12, no. 3, pp. 273–302, 2004.

[14] G. Acampora, J. M. Cadenas, V. Loia, and E. Munoz Ballester, "Achieving memetic adaptability by means of agent-based machine learning," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 557–569, 2011.

[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.

[19] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 474–488, 2005.

[20] Y. S. Ong, M. H. Lim, N. Zhu, and K. W. Wong, "Classification of adaptive memetic algorithms: A comparative study," *IEEE Transactions On Systems, Man and Cybernetics - Part B*, vol. 36, no. 1, pp. 141–152, 2006.

[21] D. Sudholt, "The impact of parametrization in memetic evolutionary algorithms," *Theor. Comput. Sci.*, vol. 410, no. 26, pp. 2511–2528, 2009.

[22] 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.

[23] 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.

[24] R. Hooke and T. A. Jeeves, "Direct search solution of numerical and statistical problems," *Journal of the ACM*, vol. 8, pp. 212–229, Mar. 1961.

[25] 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.

[26] H. H. Rosenbrock, "An automatic method for finding the greatest or least value of a function," *The Computer Journal*, vol. 3, no. 3, pp. 175–184, 1960.

[27] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory And Algorithms*. Wiley-Interscience, May 2006.

[28] M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *The Computer Journal*, vol. 7, no. 2, pp. 155–162, Jan. 1964.

[29] 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.

[30] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization," University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL): Hefei, Anhui, China, Tech. Rep.

[31] Cyber Dyne Srl Home Page, http://cyberdynesoft.it/.

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