Memetic Algorithm for Intense Local Search Methods Using Local Search Chains

Daniel Molina[1], Manuel Lozano[2], C. García-Martínez[3], and Francisco Herrera[2]

[1] University of Cádiz, Department of Computer Languages and Systems, Cádiz
daniel.molina@uca.es
[2] University of Granada, Department of Computer Science and Artificial Inteligence, Granada
lozano@decsai.ugr.es, herrera@decsai.ugr.es
[3] University of Córdoba, Department of Computing and Numerical Analysis, Córdoba
cgarcia@uco.es

Abstract. This contribution presents a new memetic algorithm for continuous optimization problems, which is specially designed for applying intense local search methods. These local search methods make use of explicit strategy parameters to guide the search, and adapt these parameters with the purpose of producing more effective solutions. They may achieve accurate results, at the cost of requiring high intensity, making more difficult their application into a memetic algorithm. Our memetic algorithm approach assigns to each individual a local search intensity that depends on its features, by chaining different local search applications. With this technique of search chains, at each stage the local search operator may continue the operation of a previous invocation, starting from the final configuration reached by this one. The proposed memetic algorithm integrates the CMA-ES algorithm as their local search operator. We compare our proposal with other memetic algorithms and evolutionary algorithms for continuous optimization, showing that it presents a clear superiority over the compared algorithms.

1 Introduction

It is now well established that hybridisation of *evolutionary algorithms* (EAs) with other techniques can greatly improve the efficiency of search [1,2]. EAs that have been hybridised with local search techniques are often called *memetic algorithms* (MAs) [3,4,5]. One commonly used formulation of MAs improvement the new member of the population using a local search (LS) method, with the aim of exploiting the best search regions gathered during the global sampling done by the EA. That allow to design MAs for continuous optimisation (MACOs) that obtain very accuracy solutions in these type of problems [6].

For function optimization problems in continuous search spaces, an important difficulty must be addressed: solutions of high precision must be obtained by the solvers. MAs comprising efficient local improvement processes on continuous domains (continuous LS methods) have been presented to deal with this problem [6]. In this paper, they will be named MACOs (MAs for continuous optimization problems).

Most well-known continuous LS algorithms make use of explicit *strategy parameters* (e.g., *step sizes*) to guide the search. Generally, they adapt these parameters with the purpose of increasing the likelihood of producing more effective solutions. Because of their explicit parameter adaptation, these algorithms may require a substantial number of evaluations to achieve adequate styles of traversal of solution space to follow certain paths leading to precise final solutions. We call these local search *intense local search methods*. For this behaviour, the usual hybridization model is not adequate enough to these intense continuous LS algorithms, because the total function evaluations invested by the LS operator may become too high, hindering to obtain profitable synergetic effects between the EA and the LS algorithm.

In this contribution, we present a MACO model specially designed to incorporate intense continuous LS methods as LS operators. Our proposal applies the local search with an adaptive intensity, exploiting with higher intensity the most promising individuals. To adapt the LS intensity, our proposal can apply the LS operator several times over the same individual, using a fixed LS intensity, creating *LS chains*.

With this technique of *LS chains*, an individual resulting from a LS invocation can later become the initial point of a subsequent LS application, using the final strategy parameter values achieved by the former as its initial ones. In this way, the continuous LS method may *adaptively* fit its strategy parameters to the particular features of these zones. In our study, we use CMA-ES [7] as intense continuous LS algorithm, which stands out as an excellent local searcher.

This contribution is set up as follows. In Section 2, we present different aspects of *intense* local search and it is described the CMA-ES algorithm. In Section 3, we present the concept of LS chain and how it can be applied to improve the integration with *intense* continuous local searches. In Section 4, we present an experimental study to compare our proposal with other algorithms proposed in the literature. Finally, in Section 5, we provide the main conclusions of this work.

2 Intense Continuous Local Search Algorithms and CMA-ES

In his pioneer work on MACOs, Hart [6] demonstrated that the choice of continuous LS algorithm affects the performance of MACOs significantly on a variety of benchmark problems with diverse properties.

Most well-known continuous LS algorithms make use of explicit *strategy parameters* (e.g., *step sizes*) to guide the search. Generally, they adapt their parameters, in such a way that the moves being made may be of varying sizes, depending on the success of previous steps, with the purpose of increasing the likelihood of producing more effective solutions. Due to their explicit parameter adaptation, these continuous LS algorithms may require high LS intensity values to adapt their strategy parameters to the local topography of the search areas being refined. They are called intense continuous LS algorithms.

The integration of intense continuous LS algorithms into MACOs arises as a research area particularly attractive, because, nowadays, there are advanced intense continuous LS algorithms that stand out as formidable local searchers. The *covariance matrix adaptation evolution strategy* (CMA-ES) [7,8] is one of them. CMA-ES was originally introduced to improve the LS performance of evolution strategies. Even though CMA-ES even reveals competitive global search performances [9], it has exhibited effective abilities for the local tuning of solutions. At the *2005 Congress of Evolutionary Computation*, a *multi-start LS metaheuristics* using these method, called L-CMA-ES [10], was one of the winners of the real-parameter optimization competition [11,12]. Thus, investigating the behaviour of CMA-ES as LS component for MACOs deserves much attention.

In CMA-ES, not only is the step size of the mutation operator adjusted at each generation, but so too is the step direction in the multidimensional problem space, by a covariance matrix whose elements are updated as the search proceeds. In this work, we use the (μ_W, λ) CMA-ES model. For every generation, this algorithm generates a population of λ offspring by sampling a multivariate normal distribution:

$$x_i \sim N\left(m, \sigma^2 C\right) = m + \sigma N_i(0, C) \text{ for } i = 1, \cdots, \lambda,$$

Where the mean vector m represents the favourite solution at present, the so-called step-size σ controls the step length, and the covariance matrix C determines the shape of the distribution ellipsoid. Then, the μ best offspring are used to recalculate the mean vector, σ and m and the covariance matrix C, following equations that may be found in [7] and [9]. The default strategy parameters are given in [9]. Only the initial m and σ parameters have to be set depending on the problem.

It can be interpreted any evolution strategy that uses intermediate recombination as a LS strategy [7]. Thus, since CMA-ES is extremely good at detecting and exploiting local structure, it turns out to be a particularly reliable and highly competitive EA for local optimization [10]. Also, it can be described as an intense continuous LS algorithm, because, as noted by Auger, Schoenauer and Vanhaecke [13]: *"CMA-ES may require a substantial number of time steps for the adaptation of the covariance matrix"*.

3 MACOs Based on Local Search Chains

Due to the potential of the intense continuous LS algorithms, it becomes interesting to build MACO models with them. These MACOs should be specifically designed to accomplish two essential aims:

- The intense continuous LS algorithm should be provided with sufficient LS intensity to make their correct operation possible.
- The MACO should ensure that a profitable synergy between the continuous LS algorithm and the EA is possible.

In this section, we propose a MACO approach conceived to attain these two objectives. It is a steady-state MA model that employs the concept of *LS chain* to adjust the LS intensity assigned to the intense continuous LS method. In particular, our MACO handles LS chains, throughout the evolution, with the objective of allowing the continuous LS algorithm to act more intensely in the most promising areas represented in the EA population. In this way, the continuous LS method may adaptively fit its strategy parameters to the particular features of these zones.

In Section 3.1, we introduce the foundations of steady-state MAs. In Section 3.2, we present the concept of *LS chain*. In Section 3.3, we propose a MACO approach that handles LS chains with the objective of make good use of intense continuous LS methods as LS operators. Finally, in Section 3.4, we present an instance of our MACO model that uses CMA-ES as continuous LS operator.

3.1 Steady-State MAs

In *steady-state* GAs [14] usually only one or two offspring are produced in each generation. Parents are selected to produce offspring and then a decision is made as to which individuals in the population to select for deletion in order to make room for the new offspring. Steady-state GAs are *overlapping* systems because parents and offspring compete for survival.

Although steady-state GAs are less common than generational GAs, Land recommended their use for the design of *steady-state MAs* (steady-state GAs plus LS) because they may be more stable (as the best solutions do not get replaced until the newly generated solutions become superior) and they allow the results of LS to be maintained in the population. So, steady-state MAs integrate global and local search more tightly than generational MAs [15]. This interleaving of the global and local search phases allows the two to influence each other.

3.2 Local Search Chains

In steady-state MAs, individuals resulting from the LS invocation may be for a long time latter selected to be replaced. At this point, we propose *to chain* an LS algorithm invocation and the next one as follows:

> *The final configuration reached by the former (strategy parameter values, internal variables, etc.) is used as initial configuration for the next application.*

In this way, the LS algorithm may continue under the same conditions achieved when the LS operation was previously halted, providing an *uninterrupted connection between successive LS invocations*, i.e., forming a *LS chain*. Figure 1 shows an example of LS chain formed by a LS algorithm with only one associated strategy parameter, p.

Two important aspects that were taken into account for the management of LS chains are:

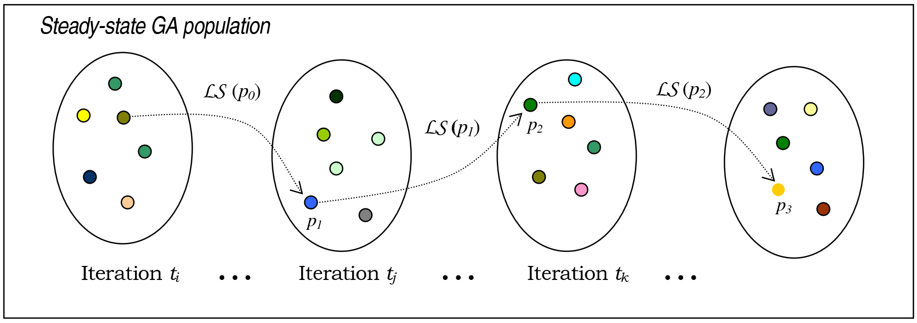

Fig. 1. Example of LS chain. p_x is the value for the strategy value, p_{i+1} is the final parameter value reached when it started with a value of p_i, and p_0 is its default value

– Every time the LS algorithm is applied to refine a particular chromosome, it is applied a fixed LS intensity, that will be called *LS intensity stretch* (I_{str}). In this way, a LS chain formed throughout n_{app} LS applications and started from solution s_0 will return the same solution as the application of the continuous LS algorithm to s_0 employing $n_{app} \cdot I_{str}$ fitness function evaluations.
– After the LS operation, the parameters that define the current state of the LS processing are stored along with the reached final individual (in the steady-state GA population). When this individual is latter selected to be improved, the initial values for the parameters of the LS algorithm will be directly available.

In this work, we argue that a promising approach to *adapt* the LS intensity assigned to intense continuous LS algorithms is using MACOs that allow LS chain to grow throughout the evolution depending on the quality of the search regions being visited, with the aim of acting more intensely in the most promising areas. In this way, the real LS intensity assigned to the continuous LS algorithm may be adaptively determined throughout the run and depends on two crucial choices:

– The way the solutions are selected to apply the LS operator to them.
– The replacement scheme used by the steady-state GA.

The designer of the steady-state GA is responsible for the second election, whereas the first one should be undertaken during the design of the MACO scheme.

3.3 A MACO Model That Handles Local Search Chains

In this section, we propose a MACO model (see Figure 2) with the following main features:

1. It is a steady-state MA model.
2. It ensures that a fixed and predetermined local/global search ratio is always kept. With this policy, we easily stabilise this ratio, which has a strong influence on the final MACO behaviour, avoiding an excessive exploitation.

3. It favours the enlargement of those LS chains that are showing promising fitness improvements in the best current search areas represented in the steady-state GA population. In addition, it encourages the activation of innovative LS chains with the aim of refining unexploited zones, whenever the current best ones may not offer profitability. The criterion to choose the individuals that should undergo LS is specifically designed to manage the LS chains in this way (Steps 3 and 4).

```
1. Generate the initial population.
2. Perform the steady-state GA throughout $n_{frec}$ evaluations.
3. Build the set $S_{LS}$ with those individuals that potentially may be refined
   by LS.
4. Pick the best individual in $S_{LS}$ (Let's $c_{LS}$ to be this individual).
5. if $c_{LS}$ belongs to an existing LS chain then
6.     Initialise the LS operator with the LS state stored together with $c_{LS}$.
7. else
8.     Initialise the LS operator with the default LS state.
9. Apply the LS algorithm to $c_{LS}$ with an LS intensity of $I_{str}$ (Let's $c^r_{LS}$ to be
   the resulting individual).
10. Replace $c_{LS}$ by $c^r_{LS}$ in the steady-state GA population.
11. Store the final LS state along with $c^r_{LS}$.
12. If (not termination-condition) go to step 2.
```

Fig. 2. Pseudocode algorithm for the proposed MACO model

The proposed MACO scheme defines the following relation between the steady-state GA and the intense continuous LS method (Step 2): *every n_{frec} number of evaluations of the steady-state GA, apply the continuous LS algorithm to a selected chromosome, c_{LS}, in the steady-state GA population.* Since we assume a fixed $\frac{L}{G}$ ratio, $r_{L/G}$, n_{frec} may be calculated using the equation $n_{frec} = I_{str}\frac{1-r_{L/G}}{r_{L/G}}$, where n_{str} is the LS intensity stretch (Section 3.2). We recall that $r_{L/G}$ is defined as the percentaje of evaluations spent doing local search from the total assigned to the algorithm's run.

The following mechanism is performed to select c_{LS} (Steps 3 and 4):

1. Build the set of individuals in the steady-state GA population, S_{LS} that fulfils:
   (a) They have never been optimised by the intense continuous LS algorithm, or
   (b) They previously underwent LS, obtaining a fitness function improvement greater than δ^{min}_{LS} (a parameter of our algorithm).

With this mechanism, when the steady-state GA finds a new best so far individual, it will be refined immediately. In addition, the best performing individual in the steady-state GA population will always undergo LS whenever the fitness improvement obtained by a previous LS application to this individual is greater than the δ^{min}_{LS} threshold.

3.4 Memetic Algorithm with LS Chaining and CMA-ES

In this section, we build an instance of the proposed MACO model (Figure 2), which applies CMA-ES (Section 2) as intense continuous LS algorithm. It will be called MA-LSCh-CMA. Next, we list the main features of this algorithm:

Steady-state GA. It is a real-coded steady-state GA [16] specifically designed to promote high population diversity levels by means of the combination of the BLX-α crossover operator with a high value for its associated parameter ($\alpha = 0.5$) and the *negative assortative mating* strategy [17], in combination with the *replacement strategy*. Diversity is favoured as well by means of the BGA mutation operator [18]. This combination of selection parents and replacement strategy let achieve an adequate trade-off between exploration and exploitation, th In the MA literature, keeping population diversity while using LS together with an EA is always an issue to be addressed, either implicitly or explicitly [19,20].

CMA-ES as Continuous LS algorithm. MA-LSCh-CMA follows the MACO approach, presented in Section 3.3, to handle LS chains, with the objective of tuning the intensity of CMA-ES, which is employed as intense continuous LS algorithm (Section 2). The application of CMA-ES for refining an individual, C_i, is carried out following the next guidelines:

- C_i becomes the initial mean of distribution ($\boldsymbol{m}$).
- The initial σ value is half the distance of C_i to its nearest individual in the steady-state GA population (this value allows an effective exploration around C_i).

CMA-ES will work as local searcher consuming I_{str} fitness function evaluations. Then, the resulting solution will be introduced in the steady-state GA population along with the current value of the covariance matrix, the mean of the distribution, the step-size, and the variables used to guide the adaptation of these parameters (B, BD, D, p_c and p_σ). Latter, when CMA-ES is applied to this inserted solution, these values will be recovered to proceed with a new CMA-ES application. When CMA-ES is performed on solutions that do not belong to an existing chain, default values, given in [9], are assumed for the remaining strategy parameters.

Parameter setting. For the experiments, MA-LSCh-CMA applies BLX-α with $\alpha = 0.5$. The population size is 60 individuals and the probability of updating a chromosome by mutation is 0.125. The n_{ass} parameter associated with the negative assortative mating is set to 3. The value of the $\frac{L}{G}$ ratio, $r_{L/G}$, was set to 0.5, which represents an well-balanced choice. Finally, a value of 1e-8 was assigned to the δ_{LS}^{min} threshold.

4 Experiments

We have carried out different experiments to assess the performance of MA-LSCh-CMA. In order to do this, in this section, we detail the test functions and the experimental setup and statistical methods that were used for this experimental study.

This section is structure as following: In Section 4.1 it is presented the test functions applied for the experiments. In Section 4.2 there are presented the of the experiments. In Section 4.3 analyses the influence of the LS intensity stretch in our proposal. In Section 4.4 it is studied the convenience of the LS chaining. In Section 4.4, we compare our proposal with other modern metaheuristics with the L-CMA-ES and in section 4.5 with the DEahcSPX algorithms. Finally, in Section 4.6, they are shown the numerical results (average error) obtained by each one of the algorithms considered in this Section.

4.1 Test Functions

The test suite that we have used for different experiments consists of 20 benchmark functions chosen from the set designed for the *special session on real parameter optimisation organised in the 2005 IEEE congress on evolutionary computation* (CEC2005). We have considered only the multimodal functions (F6-F25) from the CEC2005 test suite; because we are particularly interested in analysing its behaviour with complicated test functions. It is possible to consult in [11] the complete description of the functions. Also, we have considered the dimension 30, because we want to focus our study on the most dificult problems.

4.2 Experimental Setup and Statistical Analysis

The experiments have been done following the instructions indicated in the document associated to the competition. The main characteristics are:

- Each algorithm is run 25 times for each test function, and the error average of the best individual of the population is computed.
- The study has been made with dimensions $D = 30$.
- The maximum number of fitness evaluations that we allowed for each algorithm to minimise the error was $10,000 \cdot N$, where N is the dimension of the problem.
- Each run stops either when the error obtained is less than 10^{-8}, or when the maximal number of evaluations is achieved.

We have carried out the experimental study of MA-LSCh-CMA and the other algorithms following these guidelines in order to make possible its comparison with the results of all the other algorithms involved in the competition (their results are available in the proceedings of the congress).

To analyse the results we have chosen to use *non-parametric tests* because it has been proven that in this benchmark the parametric tests cannot be applied with security [21]. In particular, we have considered two alternative methods based on non parametric tests to analyse the experimental results, previously applied in comparisons of EAs [21]:

- Application of the *Iman and Davenport's* test and the *Holm's* method as post-hoc procedure. The first test is used to see whether there are significant statistical differences among the algorithms in certain group. If differences are detected, then Holm's test is employed to compare the best algorithm (control algorithm) against the remaining ones.
- Utilization of the *Wilcoxon* matched-pairs signed-ranks test. Using this test, the results of two algorithms may be directed compared.

In [21] these statistical tests are explained in detail.

4.3 Influence of the LS Intensity Stretch

In our first empirical study, we investigate the influence of I_{str} on the performance of MA-LSCh-CMA. In particular, we analyse the behaviour of this algorithm when different values for this parameter are considered ($I_{str} = 100$, 500, and 1000).

First, it is applied the *Iman-Davenport* tests at the 5% level, Table 1 shows the results.

Table 1. Results of the Iman-Davenport's test with different I_{str} values

Iman-Davenport value	Critical value	Sig. differences
1,17	2,77	No

From Table 1, we may extract an important conclusion: MA-LSCh-CMA exhibits a low sensitivity degree to the value selected for I_{str}. We have chosen a particular value for I_{str}, in order to allow the incoming study of our proposal and the comparison with other EA models to be easily understandable.

Figure 3 shows the average rankings obtained by the MA-LSCh-CMA instances with different I_{str} values on the test functions for the different dimensions. The height of each column is proportional to the ranking, *the lower a column is, the better its associated algorithm is*. In order to study these results, we may see that $I_{str} = 500$ is the best choice, so it is the selected value.

4.4 Studying the Behaviour of the Proposed MACO Model

In this section, we have performed two different experiments, in order to investigate the behaviour of the proposed MACO model.

Comparison with a Standard MACO. First, we want to check if the LS chain offers an improvement over a standard MACO using a CMA-ES as its

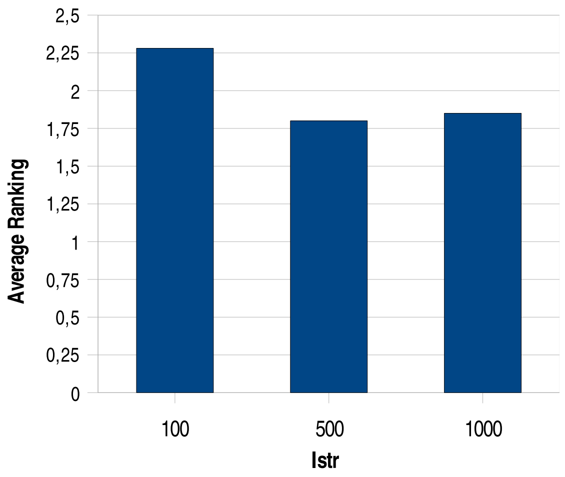

Fig. 3. Rankings obtained by MA-LSCh-CMA instances with different I_{str} values

LS method, which will be denoted as S-MACO. The basic difference between S-MACO and MA-LSCh-CMA is that the former always selects the best performing individual in the steady-state GA population as the one to be improved by CMA-ES, which starts from default values for its strategy parameters and consumes I_{LS} evaluations. It has been tested S-MACO with three different values for I_{LS} were investigated: 100, 500, and 1000, obtained that 1000 is the best value for I_{LS}. We should point out that S-MACO fits the $\frac{L}{G}$ ratio to 0.5, such as MA-LSCh-CMA does.

So, we have compared MA-LSCh-CMA ($I_{str} = 500$) with S-MACO with $I_{LS} = 1000$, using Wilcoxon's test. Table 2 summarizes the results of this procedure, where the values of $R+$ and $R-$ (associated to MA-LSCh-CMA) of the test are specified (the lowest ones, which correspond to the best results, are highlighted in bold face), together with the critical values.

We clearly notice that MA-LSCh-CMA obtains better results than S-MACO (the $R-$ value is lower than the $R+$ one). But in addition, the statistical test indicates that these improvements are statistically significant (because the $R-$ value is lower than the critical value).

Comparison with a Restart Local Search Algorithm In this section, we carry out the comparison of MA-LSCh-CMA with a restart CMA-ES, called

Table 2. S-MACO versus MA-LSCh-CMA using Wilcoxon's test (p-value = 0.05)

$R+$ (S-MACO)	$R-$ (MA-LSCh-CMA)	Critical value	Sig. differences?
168	**42**	52	Yes

Table 3. L-CMA-ES versus MA-LSCh-CMA (Wilcoxon's test with p-value = 0.05)

$R+$ (L-CMA-ES)	$R-$ (MA-LSCh-CMA)	Critical value	Sig. differences?
165	**45**	52	Yes

L-CMA-ES [10] because both algorithms invoke CMA-ES instances that specifically emphasise the local refinement abilities of this algorithm. Table 3 has the results of the comparison of these two algorithms using the Wilcoxon's test.

MA-LSCh-CMA exhibits overall better performance than L-CMA-ES, therefore, the work of the proposed hybridization method outperforms the one of the pure restart local search strategy.

4.5 Comparison with State-of-the-Art MACOs

In a recent publication, it has been presented a MACO model, called DEahcSPX [22], that combines differential evolution with a quick continuous LS method. DEahcSPX was compared with other MACO instances proposed in the literature, and they found that their proposal was superior to the majority of them. Thus, we assume that DEahcSPX is currency the most outstanding representative of the state-of-the-art MACOs.

In this section, we undertake the comparative analysis among DEahcSPX and MA-LSCh-CMA using Wilcoxon's test. Table 4 contains the results of this statistical test.

The results of MA-LSCh-CMA show higher quality than the ones of DEahcSPX. In addition, the superiority is statistically significant. Thus our proposal has turned out to be very competitive with state-of-the-art MACOs.

Then, we may highlight that MA-LSCh-CMA arises as one of the most prominent algorithm for global optimization over continuous spaces, by two important search processes simultaneously:

- The steady-state GA induces a scattered search promoting population diversity.
- The proliferation of long LS chains in the best regions becomes suitable to obtain adequate *accuracy* levels, letting also to search in alternative regions.

4.6 Results of Experiments

We present in this section the results of our proposal and the differents algorithms used into the comparisons. That allow to compare them with other algorithms

Table 4. DEahcSPX versus MA-LSCh-CMA (Wilcoxon's test with p-value = 0.05)

$R+$ (DEahcSPX)	$R-$ (MA-LSCh-CMA)	Critical value	Sig. differences?
169,5	**40,5**	52	Yes

Table 5. Average Errors by each algorithms in the benchmark applied

Test Functions	MA-LSCh CMA	S-MACO	DEahcSPX
F6	1.191003e+1	2.732782e+1	**1.000000e-9**
F7	**8.871392e-4**	2.067364e-3	1.163264e-3
F8	**2.027016e+1**	2.086726e+1	2.094711e+1
F9	7.827714e-9	8.374473e-9	1.000000e-9
F10	**1.838684e+1**	7.243991e+1	9.449920e+1
F11	**4.350834e+0**	9.017085e+0	2.921885e+1
F12	**7.690185e+2**	1.462644e+3	2.956616e+4
F13	2.344814e+0	**2.282783e+0**	2.365826e+0
F14	**1.268192e+1**	1.253313e+1	1.279216e+1
F15	**3.080000e+2**	3.160001e+2	3.506300e+2
F16	1.363134e+2	1.719942e+2	**1.294508e+2**
F17	**1.345630e+2**	1.427101e+2	2.048724e+2
F18	**8.156512e+2**	8.265035e+2	9.060900e+2
F19	**8.163714e+2**	8.237708e+2	9.061617e+2
F20	**8.157765e+2**	8.284801e+2	9.065054e+2
F21	5.120000e+2	5.120000e+2	**5.000000e+2**
F22	5.258481e+2	**5.043677e+2**	9.120960e+2
F23	**5.341643e+2**	5.341645e+2	5.341641e+2
F24	2.000000e+2	2.000000e+2	2.000000e+2
F25	2.108472e+2	**2.092819e+2**	2.105413e+2

using the same benchmark functions. Table 5 shows for each algorithm used its average error with the experimental setup indicated into Section 4.2. It has been remarked in bold type the lower average error for each function.

5 Conclusions

This work presents a new hybridization model specially designed to integrate intense continuous LS methods that need a high intensity. In the proposal model, the continuous LS algorithm is applied with higher intensity in the most promising solutions. It is proposed a MACO algorithm called MA-LSCh-CMA that employs the CMA-ES algorithm using the previous hybridization model.

The proposed MA-LSCh-CMA has been compared, following guidelines recommended for the *CEC 2005 special session on real-parameter optimization*, with other state-of-the-art EAs for continuous optimizations. Our proposal present significant improvements other them.

Other important conclusion is that the new hybridization model opens the design of new MACOs using efficiently a category of local search methods, intense continuous LS methods, that until now could not be easily integrated for requiring a high intensity. The design of new MACOs for other intense LS algorithms using the concept of *LS chains* will be studied as future work.

Acknowledgments

Authors thank Nasimul Noman and Hitoshi Iba for letting us use their DEahc-SPX code, and N. Hansen for kindly providing his implementation of CMA-ES in Matlab.

References

1. Davis, L.: Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York (1991)
2. Goldberg, D.E., Voessner, S.: Optimizing global-local search hybrids. In: Banzhaf, W., et al. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference 1999, pp. 220–228. Morgan Kaufmann, San Mateo (1999)
3. Moscato, P.: On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical report, Technical Report Caltech Concurrent Computation Program Report 826, Caltech, Pasadena, California (1989)
4. Moscato, P.: Memetic algorithms: a short introduction, pp. 219–234. McGraw-Hill, London (1999)
5. Merz, P.: Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies. PhD thesis, University of Siegen, Germany
6. Hart, W.: Adaptive Global Optimization With Local Search. PhD thesis, Univ. California, San Diego, CA (1994)
7. Hansen, N., Ostermeier, A.: Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation. In: Proceeding of the IEEE International Conference on Evolutionary Computation (ICEC 1996), pp. 312–317 (1996)
8. Hansen, N., Müller, S., Koumoutsakos, P.: Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). Evolutionary Computation 1(11), 1–18 (2003)
9. Hansen, N., Kern, S.: Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 282–291. Springer, Heidelberg (2004)
10. Auger, A., Hansen, N.: Performance Evaluation of an Advanced Local Search Evolutionary Algorithm. In: 2005 IEEE Congress on Evolutionary Computation, pp. 1777–1784 (2005)
11. Suganthan, P., Hansen, N., Liang, J., Deb, K., Chen, Y., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC 2005 special session on real parameter optimization. Technical report, Nanyang Technical University (2005)
12. Hansen, N.: Compilation of Results on the CEC Benchmark Function Set. In: 2005 IEEE Congress on Evolutionary Computation (2005)
13. Auger, A., Schoenauer, M., Vanhaecke, N.: LS-CMAES: a second-order algorithm for covariance matrix adaptation. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 182–191. Springer, Heidelberg (2004)
14. Whitley, D.: The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best. In: Proc. of the Third Int. Conf. on Genetic Algorithms, pp. 116–121 (1989)
15. Land, M.S.: Evolutionary Algorithms with Local Search for Combinational Optimization. PhD thesis, Univ. California, San Diego, CA (1998)
16. Herrera, F., Lozano, M., Verdegay, J.L.: Tackling Real-coded Genetic Algorithms: Operators and Tools for the Behavioral Analysis. Artificial Intelligence Reviews 12(4), 265–319 (1998)

17. Fernandes, C., Rosa, A.: A Study of non-Random Matching and Varying Population Size in Genetic Algorithm using a Royal Road Function. In: Proc. of the 2001 Congress on Evolutionary Computation, pp. 60–66 (2001)
18. Mülenbein, H., Schlierkamp-Voosen, D.: Predictive Models for the Breeding Genetic Algorithm in Continuous Parameter Optimization. Evolutionary Computation 1, 25–49 (1993)
19. Lozano, M., Herrera, F., Krasnogor, N., Molina, D.: Real-coded Memetic Algorithms with Crossover Hill-climbing. Evolutionary Computation 12(2), 273–302 (2004)
20. Tang, J., Lim, M., Ong, Y.: Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems. Soft Computing 11(9), 873–888 (2007)
21. García, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the cec 2005 special session on real parameter optimization. Journal of Heuristics (in press, 2008)
22. Noman, N., Iba, H.: Accelerating differential evolution using an adaptive local search. In: IEEE Transactions on evolutionary Computation (in press, 2008)