Memetic Comp. (2010) 2:69–83
DOI 10.1007/s12293-010-0032-9

REGULAR RESEARCH PAPER

Iterated local search with Powell's method: a memetic algorithm for continuous global optimization

Oliver Kramer

Received: 5 April 2009 / Accepted: 28 January 2010 / Published online: 24 February 2010

Abstract In combinatorial solution spaces Iterated Local Search (ILS) turns out to be exceptionally successful. The question arises: is ILS also capable of improving the optimization process in continuous solution spaces? To demonstrate that hybridization leads to powerful techniques in continuous domains, we introduce a hybrid meta-heuristic that integrates Powell's direct search method. It combines direct search with elements from population based evolutionary optimization. The approach is analyzed experimentally on a set of well known test problems and compared to a state-of-the-art technique, i.e., a restart variant of the Covariance Matrix Adaptation Evolution Strategy with increasing population sizes (G-CMA-ES). It turns out that the population-based Powell-ILS is competitive to the CMA-ES, in some cases even significantly faster and behaves more robust than the pure strategy of Powell in multimodal fitness landscapes. Further experiments on the perturbation mechanism, population sizes, and problems with noise complete the analysis of the hybrid methodology and lead to parameter recommendations.

Keywords Memetic algorithms · Iterated local search · Global optimization · Evolution strategies · Powell's direct search method

1 Introduction

In the last years many results have been published in the field of Evolutionary Algorithms (EAs) showing that hybridizations between meta-heuristics and local search techniques are exceptionally successful. Hybrid meta-heuristics are also known as memetic algorithms, Baldwinian or Lamarckian EAs. Successful hybridizations have been proposed in particular for combinatorial and discrete solution spaces [37, 67]. Interestingly, for real-valued solution spaces few results have been introduced yet—an astonishing fact as many direct search methods are fairly fast optimizers. We will give a short survey in Sect. 2.4. In this paper we introduce a hybrid meta-heuristic that is based on Powell's direct search method and population-based Iterated Local Search (ILS), a preliminary investigation of this Powell-ILS has been introduced recently [32]. The ILS is similar to a (μ, λ)-evolution strategy (ES), but each candidate solution is locally optimized using Powell's method. As the latter is exceedingly fast moving into local optima, the outer ES *only* has to search in the *space of local optima* of the fitness landscape by means of Gaussian mutations. An essential part of our approach is the Gaussian based perturbation mechanism: If the search gets stuck, the step sizes are increased. This allows to leave basins of attraction. The strengths of the perturbation mechanism is controlled in order to adapt to the *fitness landscape of local optima*.

The work is structured as follows. First, we will introduce the concept of hybridization in general and the ILS concept in Sect. 2. Section 3 introduces the Powell-ILS starting with Powell's method that is based on conjugate directions and a description of how it is integrated into the ILS technique. Section 4 provides an experimental evaluation of the proposed approach and a comparison to the restart G-CMA-ES by Auger and Hansen [1], and the pure strategy of Powell. The analysis also concentrates on the perturbation mechanism and derives recommendable parameters for the perturbation strength τ. Furthermore, we analyze the influence of population sizes and noise in the fitness function.

O. Kramer (✉)
Department of Computer Science, Technische Universität Dortmund, 44227 Dortmund, Germany
e-mail: oliver.kramer@tu-dortmund.de

2 Hybrid meta-heuristics and iterated local search

ILS belongs to the class of hybrid meta-heuristics. Before we introduce the ILS approach, we give a brief overview of hybrid approaches introducing a taxonomy of hybrid techniques for optimization problems.

2.1 Hybrid meta-heuristics

Search algorithms can be divided into two categories: exact techniques and heuristics. Exact algorithms find local optimal solutions exactly, and with a guaranteed runtime, but the computational efficiency deteriorates significantly with the size of the problem dimension. Heuristics and meta-heuristics usually approximate the solution on the basis of stochastic components, but do not find the optimum in every case. However, their runtime on large problem instances is more acceptable. The hybridization of meta-heuristics and local search methods is motivated by the combination of the advantages of the exact and the heuristic world. The success of hybridization is reflected by an increasing number of publications in this research area and the foundation of international conferences and workshops like the *HM—Hybrid Meta-Heuristics* workshop or the *Workshop on Mathematical Contributions to Meta-Heuristics*. In the case of combinatorial solution spaces exact methods like integer linear programming, dynamic programming approaches [2] or branch-and-bound techniques [33] are frequently combined with EAs.

An important design decision for hybrid techniques is the way of information interchange between its components. In which order shall the components work together, which information is shared, and when? Can general hybridization rules be derived from theory or experiments? For a systematic overview Talbi [71] and Raidl [58] proposed a taxonomy of hybrid meta-heuristics. Our view on hybrid meta-heuristics is based on this taxonomy, see Fig. 1. Hybrids can be classified into collaborative techniques that work successively or intertwined. A *relay* or *sequential* hybrid is a simple successive execution of two or more algorithmic components. The main idea is: A stochastic method *preoptimizes* coarsely while the local search performs fine-tuning and approximation of local optima. The intertwined collaborative hybrid is alternately running various optimizers. The integrative hybrids represent the other branch of the taxonomy. *Coevolutionary* or *concurrent* hybrids are nested approaches. Typically, a local search method is embedded into an evolutionary optimizer: In each iteration the local search optimizes the offspring solutions until a predefined termination condition is fulfilled. Information is passed alternately between the components in the concurrent approach. The local search method might have an own termination condition that can be specified by the embedding optimizer. The alternative to integrate stochastic optimization into a deterministic or local optimizer is rather unusual.

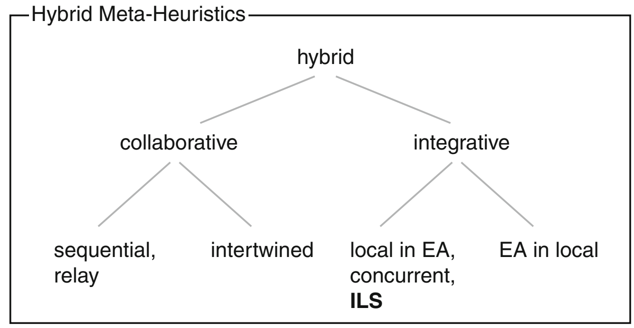

Fig. 1 Survey of hybridization strategies. Hybrids can be divided into collaborative approaches that run successively (relay or intertwined), and integrative approaches use other algorithms as operators in each iteration, e.g., a local search method embedded in an EA or vice versa

```
1  Start
2    s ← generate initial solution;
3    ŝ ← local search (s);
4    Repeat
5      s' ← perturbation (s);
6      ŝ' ← local search (s');
7      ŝ ← apply acceptance criterion (ŝ', ŝ);
8    Until termination condition
9  End
```

Fig. 2 Pseudocode of the ILS method

2.2 Iterated local search

ILS is based on a simple, but successful idea. Instead of repeating local search, starting from initial solutions like *restart*-approaches do, ILS starts with a solution s, and successively applies local search and perturbation of the local optimal solution $\hat{s}$. This procedure is repeated iteratively until a termination condition is fulfilled. Figure 2 shows the pseudocode of the ILS approach. Initial solutions should employ as much information as possible to be a good starting point for local search. Most local search operators are deterministic. Consequently, the perturbation mechanism should introduce non-deterministic components to explore the solution space. The perturbation mechanism performs global random search in the space of local optima that are approximated by the local search method. Blum and Roli [5] point out that the balance of the perturbation mechanism is quite important. The perturbation must be strong enough to allow the escape from basins of attraction, but low enough to exploit knowledge from previous iterations. Otherwise, the ILS will become a simple restart strategy. The acceptance criterion of line 7 may vary from *always accept* to *only in case of improvement*. Approaches like simulated annealing may be adopted.

There are many examples in literature for the successful application of ILS variants on combinatorial optimization problems. A survey of ILS techniques has been presented by Lourenço et al. [38]. The authors also provide a comprehensive introduction [37] to ILS. A famous combinatorial instance, many ILS methods have been developed for, is the traveling salesman problem (TSP). Stützle and Hoos [68] introduced an approach that combines restarts with a specific acceptance criterion to maintain diversity for the TSP, while Katayama and Narihisa [27] proposed a perturbation mechanism that combines 4-opt with a greedy algorithm. Stützle [69] uses an ILS hybrid to solve the quadratic assignment problem. The technique is enhanced by acceptance criteria that allow moves to worse local optima. Furthermore, population-based extensions are introduced and an experimental analysis showing that the approach outperforms other state-of-the-art algorithms. Duarte et al. [13] introduce an ILS heuristic for the problem of assigning referees to scheduled games in sports based on greedy search. Our perturbation mechanism is related to their approach. Preliminary work on the adaptation of the perturbation algorithm has been proposed by Mladenovic et al. [44] for variable neighborhood search and tabu search by Glover et al. [21], but not in real-valued search domains. For a further depiction of ILS methods and a starting point to deeper investigations we refer the reader to the mentioned literature.

2.3 Evolutionary methods in continuous domains

Evolutionary global optimization has a long tradition. EAs are famous for their biological motivation as their genetic operators can be seen as translation of concepts that can be observed in Darwinian evolution: crossover, mutation and selection. In the sixties and seventies Fogel [18], Holland [26], Rechenberg [59], and Schwefel [62] translated this paradigm of evolution into algorithms. EAs have grown to rich and frequently used optimization methods. In the context of global continuous optimization, ES have to be mentioned. They are specialized EAs for continuous optimization. For a comprehensive introduction we refer to Beyer and Schwefel [4]. ES use a $(\mu \overset{+}{,} \lambda)$ population scheme, i.e., in each generation the algorithm produces λ offspring solutions and selects μ parental solutions as parents for generation $t+1$. Often, intermediate recombination is applied. ES became famous for their mutation operator, i.e., σ-self-adaptive Gaussian mutation. Self-adaptation is an important property of ES. A self-adaptively controlled strategy parameter vector $\boldsymbol{\sigma} \in \mathcal{S}$ is subject to variation, specifies defined properties of the EA, and is selected bound to the corresponding objective variables $(\mathbf{x}', \boldsymbol{\sigma}')$ with regard to the solution quality $f(\mathbf{x}')$. For an introduction to self-adaptation we refer to Eiben et al. [14], Meyer-Nieberg and Beyer [42] or Kramer [31]. Advanced ES are the covariance matrix adaptation variants that we will briefly introduce Sect. 4.2.

Related research on evolutionary continuous optimization concerns the work of Deb et al. [11] who developed a generic parent-centric crossover operator and a steady-state, elite-preserving population-alteration model. The authors compare their approach on three test problems, and with six related optimization method reporting competitive results. Herrera et al. [24,25] proposed to apply a two-loop EA with adaptive control of mutation sizes. It adjusts the step size of an inner EA and a restart control of a mutation operator in the outer loop. Differential evolution (DE) is another branch of evolutionary methods for continuous optimization. Price et al. [56] give an introductory survey to DE. Qin et al. [57] proposed an adaptive DE that learns operator selection and associated control parameter values. The learning process is based on previously generated successful solutions. Particle swarm optimization (PSO) is a further line of research that concentrates on continuous global optimization [28,64]. PSO is inspired by the movement of swarms in nature like fish schools or flocks of birds, and simulates the movement of candidate solutions using flocking-like equations with locations and velocities. A learning strategy variant has been proposed by Liang et al. [36]. This variant uses all particles' historical best information to update the particle history. Recently, Das et al. [9] defined PSO-like neighborhood structures based on index-graphs. They proposed schemes to balance exploration and exploitation without much additional costs and report competitive experimental results on artificial benchmark and real-world problems.

Theoretical investigations can enrich experimental analyses. For this sake, Gutin and Karapetyan [23] summarized a selection of theoretical tools to analyze optimization heuristics. They discuss examples of preprocessing procedures and probabilistic instance analysis methods as well as theoretical explanations of successes and failures of heuristics. The purpose of this brief depiction was to show the broad variety of continuous optimizers in evolutionary computation. In the following section we will summarize related *memetic* approaches for continuous solution spaces.

2.4 Hybrid approaches in continuous domains

An early memetic method in continuous domains has been introduced by *Griewank* [22], who proposed to combine a gradient method with a deterministic perturbation term. Toksari and Güner [73] introduced an ILS method for real-valued search spaces based on variable neighborhood search. Their basic idea is to explore various neighborhoods using local search. Two neighborhoods are introduced: random directions and decreasing jumps. In comparison to our Powell-ILS, the lengths of the jumps are gradually decreased and not controlled according to the success of the search.

Mladenović [43] have also introduced a variable neighborhood search meta-heuristic. Random points are generated using different neighborhoods and distributions for the perturbation step. The approach is extended with exterior point penalty functions to be able to handle constraints. Yu et al. [75] analyze various immigrant schemes that maintain the diversity of the population throughout the run. A new immigrant scheme is proposed and experimental results on continuous dynamic problems are presented. Le et al. [34] introduce the concept of local optimum structure for the analysis of Lamarckian memetic algorithms. They generalize the notion of neighborhood to connectivity structure. Solution quality and efficiency of operators are analyzed to get results about structure of local optima of representative benchmark problems. Local optimum and connectivity structure turn out to have a significant influence on the performance of memetic algorithms. Ting et al. [72] propose a hybridization of genetic algorithms and tabu search to balance selection pressure and population diversity. The tabu restriction prevents inbreeding for diversity maintenance, while the aspiration criterion provides moderate selection pressure under the tabu restriction. Experimental analyses on continuous and combinatorial problems show a significant improvement of performance.

Memetic schemes have been proposed that concentrate on the choice and the balance between appropriate evolutionary and local operators. Ong et al. [49] concentrated on the choice of a proper local search technique. They presented a memetic algorithm that chooses among various local search methods at run-time. The approach is called Meta-Lamarckian learning and successfully tested on continuous parametric benchmark problems, as well as a real-world aerodynamic problem. In a follow-up paper they presented a meme adaptation heuristic that is based on classification of memes according to historical knowledge [50]. Empirical studies revealed the behavior on global benchmark problems, complemented by asymptotic convergence analyses. Nguyen et al. [47] proposed a probabilistic memetic framework that is able to adapt the balance between local search and evolutionary operators by estimating the probability of each process to reach the global optimum. Their approach is based on a theoretical upper bound for each individual and each search process. They report a robust behavior and an improvement of performance on a set of benchmark problems. Vrugt et al. [74] recently proposed a multi-method scheme that self-adaptively controls the number of offspring solutions of three methods, i.e., the CMA-ES, a common genetic algorithm, and a PSO approach contribute to the evolutionary search process in each generation. An analysis on problems from the *CEC 2005 Special Session on Real-Parameter Optimization* [70] reveals competitive results on high-dimensional multimodal problems.

Some memetic approaches concentrate on crossover. Lozano et al. [39] presented a continuous memetic algorithm with a crossover hill-climbing method as local search procedure. The approach assigns local search probabilities to each individual and is thus able to balance between local and global search. Sánchez et al. [61] stated that many effective crossover operators for real-coded genetic algorithms exist. Different crossover operators are advantageous in various situations, and at changing stages of the search process. Consequently, they proposed a technique called hybrid crossover operators as alternative in order to increase the variety of crossover operators to choose from. Second, they pointed out the strengths of operators that produce more than two offspring solutions. Noman and Iba et al. [48] proposed a crossover-based adaptive local search operator for DE. As it is not easy to determine the length of the local search for a broad class of problems, they presented a hill climber that adapts this length. An improvement of the DE technique could be observed. Also Li and Wang [53] have recently proposed a DE-based memetic approach. They introduced the differential operator of DE into harmony search, and in turn, embedded harmony search into the DE framework. Furthermore, parameter studies showed the effect on the performance of the new hybridizations. Neri and Tirronen [46] proposed a scale factor local search DE approach consisting of two local search algorithms combined with an adaptive scheme. The local search algorithms detect scale factor values during the optimization process and produce solutions, i.e., they are used as operators generating candidate solutions during the optimization process. The approach has been experimentally tested on a set of test problems from literature.

Other memetic methods stem from the area of evolutionary multi-objective optimization. Emmerich et al. [15] hybridize an ES based on $\mathcal{S}$-metric selection with local search. The main idea of the hybrid is to guide the local search by calculating the gradient of the $\mathcal{S}$-metric. Emmerich et al. report a linear convergence to the optimum. Sindhya et al. [66] have examined a hybrid version of the NSGA-II [12] with an integrated gradient descent method as local optimizer. They use an augmented scalarization function to map the multi-objective solution to a single scalar value. Martínez and Coello [40] proposed to hybridize the NSGA-II with classical direct search techniques. Shukla [65] has also hybridized the NSGA-II with two gradient methods using a perturbation technique as mutation operator. Koch et al. [29] hybridized the SMS-EMOA with Hooke and Jeeves, steepest descent [17] and the Newton method by Fliege et al. [16]. A slight improvement of the optimization process, i.e., the maximization of the $\mathcal{S}$-metric, could be observed.

Research on restart strategies, a further receipt against getting trapped in local optima, is related to the proposed ILS hybridization. In comparison to ILS, restart optimizers start *from scratch* each time, sometimes with different parameter settings. In 1977, Powell [55] introduced a restart procedure for the conjugate gradient method. As the frequency of restarts depends on the objective function, Powell's restart procedure takes it automatically into account. Furthermore, he derived a multiplying factor for the definition of the search direction. Dai et al. [8] introduced a restart algorithm based on Powell's method and the computation of the conjugate gradient direction in a transformed space. Its global convergence is proved. It shows similar experimental results like the Beal-Powell restart algorithm, analyzed by Dai and Yuan [7]. A restart approach for the CMA-ES [51] with increasing population sizes has been introduced by Auger and Hansen [1]. They state that by increasing the population size the search characteristic becomes more global after each restart. We will compare our hybrid approach to a variant of the G-CMA-ES in Sect. 4.3.

3 The Powell-ILS

Our hybrid ILS variant is based on Powell's optimization method. Preliminary experiments revealed the efficiency of Powell's method in comparison to real-valued stochastic search methods. However, as we will observe this in the experimental Sect. 4, Powell's method gets stuck in local optima in multimodal solution spaces. A similar idea to hybridize local search with stochastic optimization methods has been proposed by *Griewank* [22], who combines a gradient method with a deterministic perturbation term. A hybridization with the strategy of Powell and a control of the perturbation strength has not been proposed previously to the best of our knowledge.

3.1 The strategy of Powell

The classical non-evolutionary optimization methods for continuous problems can mainly be classified into *direct*, *gradient* and *Hessian* search methods. The direct methods determine the search direction without using a derivative [63]. Lewis et al. [35] give an overview of direct search methods. Pattern search methods [10] examine the objective function with a pattern of points that lie on a rational lattice. Simplex search [45] is based on the idea that a gradient can be estimated with a set of $N+1$ points, i.e., a simplex. Direct search methods like Rosenbrock's [60] and Powell's [54] collect information about the curvature of the objective function during the course of the search. If the derivatives of a function are available, the gradient and Hessian methods can be applied. Gradient methods take the first derivative of the function into account, while the Hessian methods also use the second derivative. A successful example is the Quasi-Newton method [6]. It searches for the stationary point of a function, where the gradient is 0. Quasi-Newton estimates the Hessian matrix analyzing successive gradient vectors.

```
Start
  Repeat
    For k = 1 To N
      find λ_k that minimizes f(x_{k-1} + λ_k d_k);
      set x_k = x_{k-1} + λ_k d_k;
      For j = 1 To N − 1;
        update vectors d_j = d_{j+1};
      Next
      set d_N = x_N − x_0;
      find λ_N that minimizes f(x_N + λ_N d_N);
      set x_0 = x_0 + λ_N d_N;
    Next
  Until termination condition
End
```

Fig. 3 Pseudocode of the conjugate gradient strategy that is the basis of Powell's method. At the beginning, the algorithm needs a set of linearly independent vectors $\mathbf{d}_i$, $1 \leq i \leq N$ and a starting point $\mathbf{x}_0$

Powell's method belongs to the direct search methods, i.e., no first or second order derivatives are required. It is based on so called conjugate directions. Powell [54,55] states that its main justification is based on the properties, when the objective function $f(\mathbf{x})$ is convex and quadratic:

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c. \tag{1}$$

Two directions $\mathbf{d}_i$ and $\mathbf{d}_j$, $i \neq j$ are mutually conjugate if it holds:

$$\mathbf{d}_i^T\mathbf{A}\mathbf{d}_j = 0. \tag{2}$$

A set of mutual conjugate directions $\mathbf{d}_i, \mathbf{d}_j \in \mathcal{X} \subset \mathbb{R}, i \neq j$ constitutes a basis of $\mathcal{X}$. The conjugate gradient method works as follows. Let $\mathbf{x}_0$ be the initial guess of a minimum of function f. In iteration k we require the gradient $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k)$. If $k = 1$, and $\mathbf{d}_k$ the steepest descent direction is $\mathbf{d}_k = -\mathbf{g}_k$. For $k > 1$ Powell applies the equation:

$$\mathbf{d}_k = -\mathbf{g}_k + \beta_k\mathbf{d}_{k-1}, \tag{3}$$

with the Euclidean vector norms:

$$\beta_k = \frac{\|\mathbf{g}_k\|^2}{\|\mathbf{g}_{k-1}\|^2}. \tag{4}$$

The main idea of the conjugate direction method is to search for the minimal value of $f(\mathbf{x})$ along direction $\mathbf{d}_k$ to obtain the next solution $\mathbf{x}_{k+1}$, i.e., find the λ that minimizes:

$$f(\mathbf{x}_k + \lambda\mathbf{d}_k). \tag{5}$$

For a minimizing λ_k set the vector $\mathbf{x}_{k+1}$ to:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k\mathbf{d}_k. \tag{6}$$

Figure 3 shows the pseudocode of the conjugate gradient method that is the basis of Powell's strategy. In our implementation the search for λ_k is implemented with line search. For a more detailed introduction and thoughts on the strategy of Powell, we refer to the depiction by Powell [54] and Schwefel [63].

3.2 Iterated local search

The Powell-ILS proposed in this paper is based on four key concepts, each focusing on typical problems that occur in real-valued solution spaces:

- Powell's optimization method [54]: Powell's method is a fast direct search optimization method, in particular appropriate for unimodal, i.e., convex fitness landscapes.
- Iterative Local Search: In order to prevent Powell's method from getting stuck in local optima, the ILS approach is based on the successive repetition of Powell's conjugate gradient method as local search technique and a perturbation mechanism.
- Population-based ILS: We propose to evolve a population of candidate solutions similar to ES for exploration.[1]
- Adaptive control of mutation strengths: The strength of the ILS-perturbation is controlled by means of an adaptive control mechanism. In case of stagnation, the mutation strength is increased in order to leave local optima.

In the previous sections we introduced the strategy of Powell and the ILS principle. Figure 4 shows the pseudocode of the Powell-ILS. At the beginning, μ initial solutions $(\mathbf{s}_0)_j \in \mathbb{R}^N$ with $1 \leq j \leq \mu$ are produced and optimized with the strategy of Powell. In an iterative loop λ offspring solutions $(\mathbf{s}_t)'_j \in \mathbb{R}^N$ with $1 \leq j \leq \lambda$ are produced by means of Gaussian mutations with the global mutation strength σ, i.e., each component $x_i \in \mathbb{R}$ of $\mathbf{s} = (x_1, \ldots, x_N)^T$ is mutated independently:

$$x'_i = x_i + \sigma \cdot \mathcal{N}_i(0, 1). \tag{7}$$

Afterwards, $\mathbf{s}'_t$ is locally optimized with the strategy of Powell and we get $\hat{\mathbf{s}}'_t$. After λ solutions have been produced this way, the μ-best are selected according to their fitness, i.e., with comma-selection. Then, we apply global recombination, i.e., the arithmetic mean $\langle \hat{\mathbf{s}}_{t+1} \rangle$ of all selected solutions is computed. The fitness of this arithmetic mean is evaluated and compared to the fitness of the arithmetic mean of the last generation. If the search stagnates, i.e., the condition

$$|f(\langle \hat{\mathbf{s}}_{t+1} \rangle) - f(\langle \hat{\mathbf{s}}_t \rangle)| < \theta \tag{8}$$

becomes true, the mutation strength is increased by multiplication with $\tau > 1$:

$$\sigma = \sigma \cdot \tau. \tag{9}$$

[1] A hybridization with covariance matrix optimizers is no reasonable undertaking as the local search method disturbs the Gaussian based update of the covariance matrix, and our experimental analysis confirmed that no further improvement can be gathered in comparison to the approach at hand.

Start
  $t = 0$;
  $\mathbf{s}_0 \leftarrow$ generate μ initial solutions;
  $\forall \ \hat{\mathbf{s}}_0 \leftarrow$ Powell $(\mathbf{s}_0)$;
  $\langle \hat{\mathbf{s}}_1 \rangle = \frac{1}{\mu} \sum_{i=1}^{\mu} (\hat{\mathbf{s}}_0)_i$;
  Repeat
    $t = t + 1$;
    For $i = 1$ **To** λ
      $(\mathbf{s}'_t)_l \leftarrow$ mutation $(\langle (\hat{\mathbf{s}}_t)_l \rangle, \sigma)$;
      $(\hat{\mathbf{s}}_t)_l \leftarrow$ Powell$((\mathbf{s}'_t)_l)$;
      $\mathcal{P}_t \leftarrow (\hat{\mathbf{s}}_t)_l$;
    Next
    select $\mathcal{P}_{t+1}$ from $\mathcal{P}_t$ with comma-selection;
    $\langle \hat{\mathbf{s}}_{t+1} \rangle = \frac{1}{\mu} \sum_{i=1}^{\mu} (\hat{\mathbf{s}}_t)_i$;
    If $|f(\langle \hat{\mathbf{s}}_{t+1} \rangle) - f(\langle \hat{\mathbf{s}}_t \rangle)| < \theta$ **Then**
      $\sigma = \sigma \cdot \tau$;
    Else
      $\sigma = \sigma / \tau$;
  Until termination condition
End

Fig. 4 Pseudocode of the Powell-ILS

Otherwise, the mutation strength σ is decreased by multiplication with $1/\tau$. The effect of an increasing mutation strength σ is that local optima can be left. Powell's method drives the search into local optima, and the outer ILS performs a search within the space of local optima controlling the perturbation strength σ. A decrease of σ lets the algorithm converge to the local optimum in a range defined by σ. At first, this technique seems to be in contraposition to the 1/5-th success rule by Rechenberg [59]. Rechenberg's rule adapts the mutation strengths in the following way: The whole population makes use of a global mutation strength σ for all individuals. If the ratio of successful candidate solutions is greater than 1/5-th, the step size should be increased, because bigger steps towards the optimum can be done, while small steps would be a waste of time. If the success ratio is lower than 1/5-th the step size should be decreased. This rule is applied every g generations. The aim of Rechenberg's approach is to stay in the so called *evolution window*, guaranteeing nearly optimal progress. The optimal value for the step size factor τ depends on several conditions such as fitness landscape, number of dimensions N or number of generations g. This strategy is reasonable for local approximation: Smaller changes to solutions will increase the probability to be successful during approximation of local optima. However, in our approach the strategy of Powell performs the approximation of the local optimum, not the upper ES. The step control of the Powell-ILS part has another task: leaving local optima when the search stagnates. Of course, the local optimum may be the global one, and left again. The step control might push the ILS away from the global optimum, but if the vicinity of the optimum has been reached, it is probable that the optimum will be reached again. The point is that basins of attractions can be left because of the increasing step size. Hence, the probability is greater than zero that also the

Table 1 Survey of test problems, and corresponding stagnation criteria, perturbation strengths and population ratios for the Powell-ILS

Name	Problem	θ	τ	(μ, λ)
Sphere	$f_{\text{Sp}}(\mathbf{x}) = \sum_{i=1}^{N} x_i^2$	10^{-6}	2.0	(2,8)
Doublesum	$f_{\text{Dou}}(\mathbf{x}) = \sum_{i=1}^{N} \left(\sum_{j=1}^{i}(x_j)\right)^2$	10^{-6}	2.0	(2,8)
Noisy Doublesum	$f_{\text{NDou}}(\mathbf{x}) = \sum_{i=1}^{N} \left(\sum_{j=1}^{i}(x_j)\right)^2 \cdot (1 + \delta \cdot \lvert\mathcal{N}(0,1)\rvert)$	10^{-6}	2.0	(2,8)
Ellipsoidal	$f_{\text{Ell}}(\mathbf{x}) = \sum_{i=1}^{N} (10^6)^{\frac{i-1}{N-1}} \cdot x_i^2$	10^{-6}	2.0	(2,8)
Rosenbrock	$f_{\text{Ros}}(\mathbf{x}) = \sum_{i=1}^{N-1} \left((100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right)$	10^{-6}	2.0	(2,8)
Rastrigin	$f_{\text{Ras}}(\mathbf{x}) = \sum_{i=1}^{N} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	10^{-6}	2.0	(2,8)
Griewank	$f_{\text{Gri}}(\mathbf{x}) = \sum_{i=1}^{N} \frac{x_i^2}{4000} - \prod_{i=1}^{N} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	10^{-1}	5.0	(1,4)
Ackley	$f_{Ack}(\mathbf{x}) = 20 + e - 20 \cdot e^{\left(-0.2\sqrt{\frac{1}{N}\cdot\sum_{i=1}^{N} x_i^2}\right)} - e^{\left(\frac{1}{N}\cdot\sum_{i=1}^{N}\cos(2\pi\cdot x_i)\right)}$	10^{-6}	2.0	(2,8)
Schwefel	$f_{\text{Sch}}(\mathbf{x}) = 418.9820 \cdot N - \sum_{i=1}^{N} \left(x_i \sin\sqrt{\lvert x_i\rvert}\right)$	10^{-1}	2.0	(2,8)
Kursawe	$f_{\text{Kur}}(\mathbf{x}) = \sum_{i=1}^{N} \left(\lvert x_i\rvert^{0.8} + 5 \cdot \sin(x_i)^3 + 3.5828\right)$	10^{-6}	2.0	(2,8)

The first four problems are unimodal, the last six problems are multimodal

global optimum can be found. The problem that the global optimum may also be left, if not recognized, can be compensated by saving the best found solution in the course of the optimization process.

4 Experimental analysis

This section provides an experimental analysis of the Powell-ILS, in particular in comparison to the restart variant of the CMA-ES with increasing population sizes. The experimental analysis concentrates on typical test problems known in literature, oriented to the *CEC 2005 Special Session on Real-Parameter Optimization* [70], see Table 1. We use the following performance measure. The experimental results of this paper show the number of fitness function evaluations *until the optimum is reached with accuracy* f_{stop}, i.e., if the difference between the best achieved fitness $f(\mathbf{x}_b)$ of one candidate solution $\mathbf{x}_b$ of the algorithm and fitness $f(\mathbf{x}^*)$ of the known optimum $\mathbf{x}^*$ is lower than f_{stop}:

$$\|f(\mathbf{x}_b) - f(\mathbf{x}^*)\| \leq f_{\text{stop}} \quad (10)$$

This performance measure is focused on the convergence abilities of the approach. Parameter # counts the number of runs the optimum has been reached. In case of stagnation, the algorithm may also terminate. In this case the number of evaluations is not considered in the performance analysis.

4.1 Powell's method

At first, we analyze Powell's method on the test suite, introduced in Table 1. Solutions are randomly initialized in the interval $[-100, 100]^N$. Each experiment is repeated 30 times. Powell's method terminates, if the improvement from one to the next iteration is lower than $\phi = 10^{-10}$, or if the optimum is found with accuracy $f_{\text{stop}} = 10^{-10}$. As Powell's method is a convex optimization technique, we expect that only the unimodal problem can be solved. Table 2 confirms these expectations. On unimodal functions Powell's method is exceedingly fast. On the *Sphere* problem, e.g., a budget of only 101.7 in mean is sufficient to approximate the optimum. These fast approximation capabilities can also be observed on the other convex problems, i.e., *Doublesum*, the *Noisy Doublesum* (with $\delta = 1/N^2$), the *Ellipsoidal* function (F_3 of the CEC 2005 benchmarks) and on *Rosenbrock*. This success can also be observed on higher dimensions like $N = 30$, and $N = 50$.

The experiments on the *Noisy Doublesum* problem show that noise limits the convergence behavior of Powell's method. Experiments with the settings suggested in the CEC 2005 benchmarks, i.e., noise with the strength $4.0 \cdot \mathcal{N}(0, 1)$, showed that Powell's method is not able to cope with strong noise. However, Powell's method is able to cope with noise of strength $1/N^2 \cdot \mathcal{N}(0, 1)$ and is faster than the G-CMA-ES. We will come back later to optimization with noise in Sect. 4.5. The figures also show that Powell's method is not able to approximate the optima of multimodal functions like *Rastrigin*, *Schwefel* or *Kursawe*. On "easier" multimodal functions like *Griewank* or *Ackley*, the random initializations allow to find the optimum in few of the 30 runs. The fast convergence behavior on convex function parts motivates to perform local search as operator in a global evolutionary optimization framework, and is the basis of the Powell-ILS that we will analyze in Sect. 4.3.

Table 2 Experimental comparison of Powell's method on the test problems with $N = 10, 30$, and 50 dimensions

	Powell's Method					
	Best	Median	Worst	Mean	Std	#
$N = 10$						
f_{Sp}	100	102	102	101.7	0.67	30
f_{Dou}	91	92	92	91.8	0.42	30
f_{NDou}	716	844.5	5,156	1,942.4	1,710.48	30
f_{Ell}	102	102	139	105.7	11.70	30
f_{Ros}	2,947	4,617	12,470	5,941.87	3,353.14	24
f_{Ras}	–	–	–	–	–	0
f_{Gri}	329	329	329	329	0	1
f_{Ack}	3,340	3,480	3,630	3,477	103.49	2
f_{Sch}	–	–	–	–	–	0
f_{Kur}	–	–	–	–	–	0
$N = 30$						
f_{Sp}	299	302	302	301.3	1.05	30
f_{Dou}	291	291.5	292	291.5	0.52	30
f_{NDou}	1,856	2,172	2,447	2,143.7	201.81	30
f_{Ell}	300	302	492	335.1	69.48	30
f_{Ros}	14,888	33,315	59,193	36,455.85	16,789.41	21
f_{Ras}	–	–	–	–	–	0
f_{Gri}	904	997	1,001	967.33	54.88	3
f_{Ack}	–	–	–	–	–	0
f_{Sch}	–	–	–	–	–	0
f_{Kur}	–	–	–	–	–	0
$N = 50$						
f_{Sp}	497	501	780	547	100.10	30
f_{Dou}	488	490.5	492	490.5	1.17	30
f_{NDou}	3,001	3,150.5	3,400	3,153.6	110.85	30
f_{Ell}	500	818	944	731.5	204.04	30
f_{Ros}	70,564	75,815.5	81,067	75,815.5	7,426.74	6
f_{Ras}	–	–	–	–	–	0
f_{Gri}	1,160	1,160	1,160	1,160	0	1
f_{Ack}	–	–	–	–	–	0
f_{Sch}	–	–	–	–	–	0
f_{Kur}	–	–	–	–	–	0

Best, *median*, *worst*, *mean* and *dev* provide statistical information about the number of fitness function evaluations of 30 runs until the difference between the fitness of the best solution and the optimum is smaller than $f_{\text{stop}} = 10^{-10}$. Parameter # states the number of runs that find the optimum

4.2 The G-CMA-ES: a CMA-ES restart variant with increasing population sizes

For comparison with the proposed Powell-ILS we tested a competitive CMA-ES variant, a modified G-CMA-ES from Auger and Hansen [1], on the same set of problems. A first approach, the cumulative path-length control, that is the basis of the CMA-ES has been introduced by Ostermeier et al. [52]. The cumulative path-length control is an approach to derandomize the adaptation of strategy parameters. Two algorithmic variants were the results of their attempt: the cumulative step-size adaptation (CSA) [20], and later the CMA-ES [51]. It is based on a covariance matrix adaptation that determines the shape of the mutation distribution. The covariance matrix implicitly approximates the Hessian and transforms the problem into a simpler one. Beyer and Sendhoff [3] recently introduced the self-adaptive variant CMSA-ES. Many successful examples of CMA-ES applications can be reported, e.g., in flow optimization [30] or in optimization of kernel parameters for classification problems [41].

The G-CMA-ES has been introduced and tested by Auger and Hansen [1] in the context of the CEC 2005 Special

Session on Real-Parameter Optimization [70]. It is a CMA-ES that is restarted in case of a set of conditions, and uses increasing population sizes after each restart. Initial small populations allow fast convergence at the beginning. Increasing population sizes for each independent restart allow global search without getting trapped in local optima. Auger and Hansen proposed five restart criterions, e.g., fitness stagnation for $10 + 30 \cdot N/\lambda$ generations or standard deviation stagnation, see [1]. For our G-CMA-ES variant we restricted the analysis to fitness stagnation, i.e., the same restart criterion as Powell's termination criterion. The G-CMA-ES is restarted if the fitness change in 20 successive generations is below a threshold θ'. In this case the CMA-ES is restarted with double population sizes, increasing the probability of leaving local optima. Augen and Hansen did not report the influence of each of the five restart criteria or their interactions. A careful analysis of the influence of restart criteria on the G-CMA-ES and the Powell-ILS will be subject to future work, but the results stated in [1] are consistent with the results that we have achieved with only one stagnation criterion, see the following section.

4.3 Comparison between the Powell-ILS and the G-CMA-ES

In the following, we will compare the Powell-ILS with the G-CMA-ES experimentally. Table 1 shows the test problems this paper concentrates on, and the corresponding stagnation parameter θ. Initial solutions are generated in the interval $[-100, 100]^N$ with problem dimension N, and the step sizes are set to $\sigma_{\text{init}} = 1.0$. Each experiment is repeated 30 times. For the Powell-ILS we use the population ratios stated in Table 1, in most cases $\lambda = 8$ offspring solutions and $\mu = 2$ parental solutions. Each solution is mutated and locally optimized with Powell's method. Again, Powell's method terminates, if the improvement from one to the next iteration is lower than $\phi = 10^{-10}$, or if the optimum is found with accuracy $f_{\text{stop}} = 10^{-10}$. If the search on the ILS level stagnates, i.e., the achieved improvement is smaller than θ, the mutation strength is increased with mutation parameter $\tau = 2$. The G-CMA-ES variant starts with a population size of $4 + 3 \cdot \log N$, and $\mu = \lambda/2$. The G-CMA-ES stops when the achieved accuracy is equal or better than $f_{\text{stop}} = 10^{-10}$. In case of stagnation, i.e., if the termination condition is not reached, and the fitness does not change with precision $\theta' = 10^{-12}$ for $t_\theta = 20$ generations, the search is restarted with a double population size. Restarts and the increase of population sizes are repeated until the termination condition is reached. Furthermore, we assume a maximal budget of fitness function evaluation of $\text{ffe}_{\max} = 2.0 \cdot 10^6$. If the optimum has not been found within this time, Table 3 reports $> 2.0 \cdot 10^6$.

Table 3 shows the results of the analysis of the Powell-ILS and the G-CMA-ES on the test problems with $N = 10, 30,$ and 50 dimensions. The Wilcoxon rank-sum test validates the statistical relevance of the experiments, the table also shows the corresponding p-values. A discussion of the Wilcoxon test and its use, in particular with regard to the *CEC 2005 Special Session on Real-Parameter Optimization* [70] can be found in García et al. [19]. The results of Table 2 have shown that Powell's method is very fast on unimodal problems. Of course, the Powell-ILS shows the same capabilities and approximates the optimum in the first iteration. On the *Sphere* problem, *Doublesum*, the *Noisy Doublesum*, and the *Ellipsoidal* problem, this advantage is significant in comparison to the G-CMA-ES. The G-CMA-ES does not apply restarts on convex problems. The small p-value of the Wilcoxon test confirms that the results are statistically significant. In Sect. 4.5 we will explore the magnitude of noise that the Powell-ILS is able to tolerate, in particular in comparison to the G-CMA-ES on a noisy variant of the *Rastrigin* function.

We have already observed that Powell gets stuck in local optima of multimodal problems, e.g., *Rastrigin*. The Powell-ILS perturbates a solution when getting stuck and applies Powell's method again, with the perturbation mechanism of Eq. 9. The results show that the iterated application of Powell's method in each generation allows to approximate the global optimum. This effect becomes obvious on *Rastrigin*. The Powell-ILS is able to approximate the optimum, in comparison to its counterpart without ILS. It convergences significantly faster than the G-CMA-ES with $p = 1.3 \cdot 10^{-6}$. Also on *Griewank* a statistically significant superiority of the Powell-ILS can be observed. On *Rosenbrock* and *Ackley* no statistical significant superiority of any of the two optimization algorithms can be reported. In mean the Powell-ILS is worse than the G-CMA-ES. Obviously, the worst runs of the Powell-ILS cause a fitness deterioration in mean, but the best runs are still much faster than the best runs of the G-CMA-ES. The G-CMA-ES is more robust with smaller standard deviations, but does not offer the potential to find the optimal solution that fast. The *Ackley* problem is almost flat outside the interval $[-32, 32]^N$, hence the solutions are initialized within this interval. An increased initialization interval deteriorates the search as both methods only perform random walk on the flat areas. The CEC 2005 benchmark uses the same initialization interval. On the highly multimodal problem *Kursawe*, the Powell-ILS achieves good results, but the G-CMA-ES is significantly faster only 29, 670 fitness function evaluations in mean to reach the optimum.

A similar behavior can be observed on the test problems with dimensions $N = 30$, and $N = 50$, see the middle and the lower parts of Table 3. For example, on the classical *Sphere* problem with $N = 50$, Powell's method only takes 524.8 function evaluations in mean, averaged over 30 repetitions. The G-CMA-ES takes about seventeen times more evaluations. This also holds true for the other unimodal test problems. The Powell-ILS is superior, and the results are

Table 3 Experimental comparison of the Powell-ILS and the G-CMA-ES variant on the test problems with $N = 10$, 30, and 50 dimensions

	Powell-ILS					G-CMA-ES					
	Best	Median	Worst	Mean	Std	Best	Median	Worst	Mean	Std	p
$N = 10$											
f_{Sp}	99	100	153	105.1	1.6E1	2,120	2,195	2,350	2,204	7.0E1	1.3E-6
f_{Dou}	89	92	178	108.6	3.6E1	2,280	2,355	2,490	2,358	6.2E1	0.0060
f_{NDou}	667	2,063	8,331	1,698.44	6.2E2	2,660	3,225	6,210	3,573	1.0E3	0.1688
f_{Ell}	97	100	101	99.33	1.80	6,500	6,810	7,390	6,844	2.5E2	1.3E-6
f_{Ros}	3,308	5,074.5	29,250	7,772.6	7.8E3	7,060	10,550	18,080	11,292	4.1E3	0.0744
f_{Ras}	2,359	14,969.5	38,550	15,682.5	9.1E3	36,360	90,540	203,120	103,456	5.7E4	1.3E-6
f_{Gri}	477	1,506	6,572	2,240	2,283.3	2,150	4,375	13,090	5,579	4,164.7	0.0111
f_{Ack}	1,064	2,901	10,338	3,486	2.8E3	3,340	3,480	3,630	3,477	1.0E2	0.2411
f_{Sch}	331,954	1.3E6	1.7E6	1.2E6	5.0E5	>2.0E6	>2.0E6	>2.0E6	>2.0E6	–	1.3E-6
f_{Kur}	4,300	196,218	316,528	165,325	9.9E4	10,780	21,960	81,370	29,670	22,001.4	0.0069
$N = 30$											
f_{Sp}	290	295.5	299	295.3	2.83	5,684	5,880	6,118	5,896.8	1.4E2	0.0068
f_{Dou}	283	286.5	479	305.6	6.0E1	7,770	8,092	8,302	8,075.2	1.6E2	1.3E-6
f_{NDou}	1,953	2,148	2,521	2,169.44	2.0E2	8,540	8,876	9,548	8,962.8	3.6E2	1.3E-6
f_{Ell}	285	296	446	310.88	5.0E1	30,940	31,479	32,522	31,547.6	5.1E2	1.3E-6
f_{Ros}	25,363	61,768	385,964	95,339	1.0E5	45,976	51,681	109,984	58,595.6	1.9E4	0.5076
f_{Ras}	58,943	78,537.5	191,489	102,429	4.4E4	360,990	699,846	721,224	576,511.6	1.7E5	1.3E-6
f_{Gri}	971	5,994.5	20,913	9,629.6	7,689.0	6,370	6,755	17,374	8,764	4,439.6	0.1168
f_{Ack}	5,194	15,516	43,602	18,502.44	1.2E4	9,128	9,457	19,096	10,505.6	3.0E3	0.0166
f_{Sch}	>2.0E6	>2.0E6	>2.0E6	>2.0E6	–	>2.0E6	>2.0E6	>2.0E6	>2.0E6	–	–
f_{Kur}	>2.0E6	>2.0E6	>2.0E6	>2.0E6	–	55,244	89,138	138,670	93,518.6	37,420.5	1.3E-6
$N = 50$											
f_{Sp}	485	489.5	843	524.8	1.1E2	8,745	9,007.5	9,375	9,019.5	2.2E2	1.3E-6
f_{Dou}	475	482.5	601	494.1	3.7E1	14,130	14,685	15,210	14,704.5	3.9E2	1.3E-6
f_{NDou}	3,006	3,139	3,357	3,193.33	1.3E2	15,585	16,005	16,800	16,030.5	3.3E2	1.3E-6
f_{Ell}	487	492	1335	613.44	2.8E2	71,400	73,162.5	74,025	72,807	9.1E2	1.3E-6
f_{Ros}	68,555	162,145.5	968,324	252,360.5	2.6E5	106,905	131,557.5	159,825	133,153.5	1.4E4	0.2026
f_{Ras}	135,576	269,424.5	492,577	294,150,6	1.0E5	570,615	1.1E6	1.2E6	1.0E6	3.0E5	1.3E-6
f_{Gri}	2,053	18,148	38,206	18,552.8	1.2E4	10,815	11,332.5	28,845	13,068	5.5E3	0.2411
f_{Ack}	20,418	52,993	56,841	42,932.22	1.5E4	13,725	14,767.5	16,260	14,766	7.1E2	0.0050
f_{Sch}	>2.0E6	>2.0E6	>2.0E6	>2.0E6	–	>2.0E6	>2.0E6	>2.0E6	>2.0E6	–	–
f_{Kur}	>2.0E6	>2.0E6	>2.0E6	>2.0E6	–	80,385	172,027.5	365,850	192,387	82,386.0	1.3E-6

The figures show the number of fitness function evaluations until the difference between the fitness of the best solution and the optimum is smaller than $f_{\text{stop}} = 10^{-10}$. This termination condition has been reached in every run. The p-value of the Wilcoxon test is an indicator for statistical significance of the comparison

statistically relevant. On the multimodal test problems in higher dimensions similar results like for $N = 10$ can be observed: The Powell-ILS is statistically better on *Rastrigin*. The G-CMA-ES's mean and median are better on *Rosenbrock*, *Griewank* and *Ackley*, but the Powell-ILS again performed the fastest run, with the exception of *Ackley* with $N = 50$. Experiments on *Schwefel* in higher dimensions did not lead to reasonable results, but lead to the maximum budget termination criterion. On *Ackley* and on *Kursawe* the G-CMA-ES is significantly faster. On *Kursawe*, the Powell-ILS is not able to approximate the optimum, while on *Ackley*, the G-CMA-ES is about two times faster in mean for $N = 30$, and about three times faster for $N = 50$.

Figure 5 shows fitness curves and step sizes of typical runs on both problems. On the left hand side of the figure we can observe the fitness development on Schwefel's problem (left, upper part). It can clearly be observed that the search successively gets stuck and leaves local optimal. The strategy of Powell moves the candidate solutions into local optima quite fast. In average Powell needs 200

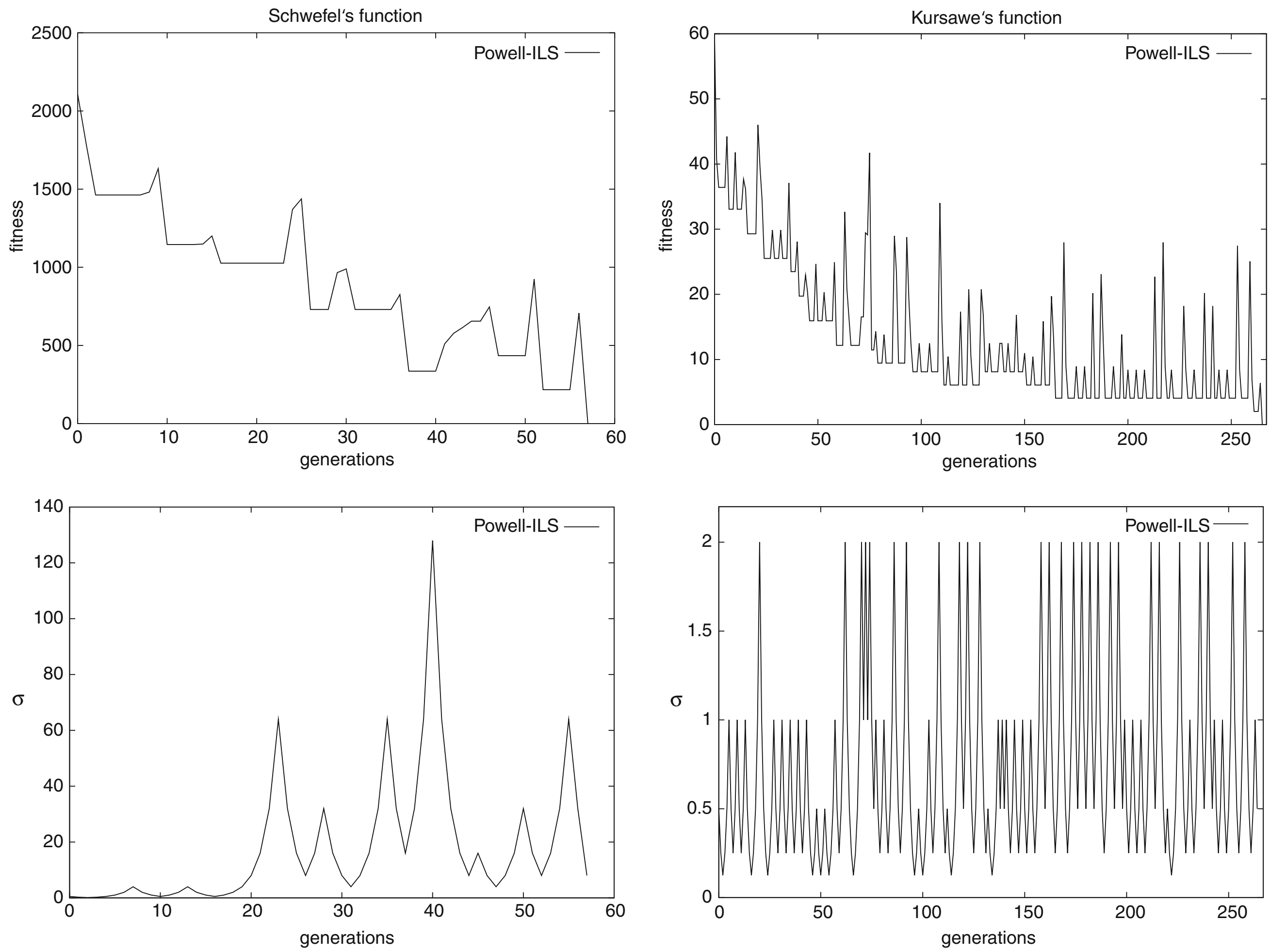

Fig. 5 *Left*: development of fitness (*upper figure*) and step size σ (*lower figure*) on the the multimodal problem *Schwefel* with $N = 10$. When the search gets stuck in local optima, the perturbation mechanism increases σ and lets the Powell-ILS escape from basins of attraction. *Right*: development of fitness (*upper figure*) on Kursawe's problem. The mutation strength σ (*lower figure*) of the perturbation mechanism is fluctuating in the highly multimodal fitness landscape

evaluations to terminate, i.e., to find a local optimum. The corresponding step sizes (left, lower part) increase within a local optimum and decrease after the local optimum has been left. A similar behavior can be observed on Kursawe's problem (right part of Fig. 5). When the search gets stuck in a local optimum, the strategy increases σ until the local optimum is successfully left and a better local optimum is found. The approach moves from one local optimum to another controlling σ, until the global optimum is found. The fitness development reveals that the search has to accept worse solutions from time to time to approximate the optimal solution. The figures confirm the basic idea of the algorithm. The ILS controls the global search, while Powell's method drives the search into local optima. Frequently, the hybrid is only able to leave local optima by controlling the strength σ of the Gaussian perturbation mechanism. The ILS conducts a search in the space of local optima.

The outcome of the experiments can be summarized as follows:

– The strategy of Powell, and also the Powell-ILS outperform the CMA-ES on unimodal problems like *Sphere*, *Doublesum*, *Noisy Doublesum*, and the *Ellipsoidal* test function.
– On highly multimodal problems like *Schwefel*, *Rastrigin* or *Kursawe* the strategy of Powell gets stuck in local optima. But the ILS-approach is able to leave these local optima and approximate the optimal solution with the help of an adaptive perturbation mechanism.
– On *Rastrigin*, *Griewank*, and *Schwefel* the Powell-ILS is significantly faster than the G-CMA-ES for $N = 10$.
– On *Kursawe* the G-CMA-ES is superior for all numbers of dimensions.

Table 4 Analysis of the Powell-ILS perturbation parameter τ and the population sizes on *Rastrigin* with $N = 30$ using the same initial settings as in the previous experiments

	$\tau = 1.5$			$\tau = 2.0$		
	Best	Median	Worst	Best	Median	Worst
(1,4)	53,913	92,544	130,675	31,686	78,214.5	121,170
(2,8)	56,074	100,835.5	143,642	65,540	112,643	242,983
(4,16)	149,350	162,083.5	210,384	77,481	117,972.5	163,693
(8,32)	156,517	295,457.5	370,320	193,259	209,725	244,325
	$\tau = 5.0$			$\tau = 10.0$		
	Best	Median	Worst	Best	Median	Worst
(1,4)	53,465	105,513	406,495	$> 2 \cdot 10^9$	$> 2 \cdot 10^9$	$> 2 \cdot 10^9$
(2,8)	48,274	104,461.5	285,651	32,773	680,363	1,473,097
(4,16)	67,241	103,142.5	202,447	52,991	208,088.5	338,922
(8,32)	109,820	189,676	221,069	123,838	309,169	802,285

Performance measure and termination condition are chosen like in the previous experiments

4.4 Perturbation mechanism and population sizes

For deeper insights into the perturbation mechanism and the interaction with population sizes, we conduct further experiments on *Rastrigin*, where the Powell-ILS has shown successful results. The strength of the perturbation mechanism plays an essential role for the ILS. In case of stagnation the step size is increased, like described in Eq. 9 with $\tau > 1$, to let the search escape from local optima. Frequently, a successive increase of the perturbation strength is necessary to prevent stagnation. In case of an improvement, the step size is decreased with the same Eq. 9, but $\tau < 1$. The idea of the step size reduction is to prevent the search process from jumping over promising regions of the solution space. In the following, we analyze the perturbation mechanism and the population sizes on *Rastrigin*. What are useful parameter settings for τ and for population parameters μ and λ? To answer this question we test various settings experimentally.

Table 4 shows the experimental results. The best result has been achieved with $\tau = 2.0$ and population sizes $(1, 4)$. Also the best median has been achieved with this setting, while the second best has been achieved with $\tau = 1.5$ and population sizes $(1, 4)$. With parameter setting $\tau = 10.0$ the Powell-ILS achieves a good best solution, but the deviation of the results is high, e.g., the worst solution is quite bad. In general, the results for $\tau = 10.0$ are quite weak, for $(1, 4)$ the algorithm does not converge within reasonable time. For low mutation strengths the best results can be observed for low population sizes. In turn, for high mutation strengths, i.e. $\tau = 5.0$, high population sizes are necessary to compensate the explorative effect. Further experiments on other problems led to the decision, that a $(2, 8)$-Powell-ILS is a good compromise between exploration and efficiency, while a $(4, 16)$-Powell-ILS is a rather conservative, but stable choice with reliable results.

4.5 The Powell-ILS and noise

In the first experiment of Powell's method on the test suite, see Table 2, we have observed that noise deteriorates the optimization process. In this section we briefly investigate the magnitude of noise under that Powell's method is still able to find the optimum, and compare the results to the G-CMA-ES. For this purpose, we use the *Noisy Double-sum* with various noise parameters, i.e., various values for parameter δ. Table 5 summarizes the results, in particular in comparison to the G-CMA-ES. While the approximation capabilities of the G-CMA-ES are almost not influenced from $\delta = 0$ to $\delta = 0.02$, the Powell-ILS gets worse when the noise is increased. For noise higher than $\delta = 0.01$ the results deteriorate significantly, for noise higher than $\delta \geq 0.1$ Powell is not able to approximate the optimum at all, and does not converge. The G-CMA-ES is robust until $\delta = 0.02$, and suddenly collapses from $\delta \geq 0.1$. Although the G-CMA-ES is able to approximate the optimum with more noise than Powell's method, for noise with $\delta \leq 0.001$ Powell is faster.

5 Conclusion

ILS is a successful hybridization technique in combinatorial solution spaces. This work shows that Powell's method is an excellent example for a meta-heuristic in real-valued solution spaces. We proposed to combine the strategy of Powell and elements from evolutionary search in an ILS framework. The hybrid outperforms the standard G-CMA-ES on unimodal problems, independent of the dimension. With

Table 5 Experimental comparison of Powell's method, the Powell-ILS, respectively, and the G-CMA-ES on the *Noisy Doublesum* problem with $N = 10$

δ	Powell / Powell-ILS					G-CMA-ES				
	Best	Median	Worst	Mean	Std	Best	Median	Worst	Mean	Std
0	92	93	93	92.8	0.42	2,390	2,500	2,590	2,488	71.30
1.00E-08	160	187	231	187.7	17.95	2,260	2,385	2,500	2,380	85.50
1.00E-07	168	190	240	192.9	18.00	2,160	2,380	2,740	2,392	157.88
1.00E-06	186	214.5	331	226.3	45.15	2,340	2,445	2,720	2,462	106.22
1.00E-05	254	327.5	382	317	41.46	2,200	2,475	2,600	2,437	136.95
1.00E-04	290	339	384	339.5	31.11	2,310	2,445	2,660	2,459	120.22
1.00E-03	412	439	461	438.1	19.42	2,350	2,480	2,580	2,464	73.36
0.01	823	1,064	3,073	1,407.1	723.25	2,290	2,430	2,620	2,441	90.73
0.02	1,391	3,744	9,434	4,369.6	2,552.12	2,300	2,435	2,780	2,465	162.08
0.1	–	–	–	–	–	2,830	3,135	3,060	3,135	3,210
0.5	–	–	–	–	–	–	–	–	–	–

Performance measure and termination condition are chosen like in the previous experiments

Powell's method the search converges fast towards local optimal solutions of convex parts of the fitness landscape. In multimodal solution spaces, the ILS controls the walk in the space of local optima. Its perturbation mechanism helps the search process to free from bad basins of attraction, making use of an adaptive step size rule that increases step sizes in case of stagnation. The experimental comparison with the G-CMA-ES reveals a competitive performance. Noise deteriorates the approximation capabilities. A deeper analysis of the perturbation mechanism revealed attractive settings for the strategy parameter τ and showed that the search can get stuck with too high mutation parameter values. A recommendable balance between exploration and performance are the settings $\mu = 2, \lambda = 8$, and $\mu = 4, \lambda = 16$.

To conclude, combining both worlds of optimization, the evolutionary and the deterministic, is a promising undertaking. It reflects the original idea of EAs: If we do not know anything about the problem, stochastic algorithms are an appropriate choice. In multimodal fitness landscapes we typically know nothing about the *landscape of local optima*. The Powell-ILS only assumes that attractive local optima lie closely together and thus the search might jump from one basin of attraction into a neighbored one. To move into local optima Powell's method turns out to be fairly successful. Furthermore, the adaptation of the perturbation strength is a natural enhancement in real-valued solution spaces. A population-based implementation allows to run multiple Powell searches in parallel and will lead to a crucial speedup in distributed computing environments. An analysis of further restart criteria like proposed by Auger and Hansen [1] will be interesting in the future.

A next step will be the integration of constraint handling methods to the ILS-framework. Many optimization problems are subject to constraints. Even simple objective functions are subject to interval constraints. One possible approach is to integrate constraint handling on the level of Powell's method. Another possibility will be the integration of penalty functions that deteriorate infeasible solutions.

References

1. Auger A, Hansen N (2005) A restart CMA evolution strategy with increasing population size. In: Proceedings of the IEEE congress on evolutionary computation—CEC 2005, vol. 2, pp 1769–1776
2. Bellman R (1957) Dynamic programming. Princeton University Press, Princeton
3. Beyer H-G, Melkozerov A (2008) σ-self-adaptive weighted multirecombination evolution strategy with scaled weights on the noisy sphere. In: Proceedings of the 10th international conference on parallel problem solving from nature (PPSN). Springer, Berlin, Heidelberg, pp 11–20
4. Beyer H-G, Schwefel H-P (2002) Evolution strategies—a comprehensive introduction. Nat Comput 1:3–52
5. Blum C, Roli A (2008) Hybrid metaheuristics: an introduction. In: Hybrid Metaheuristics. Springer, pp 1–30
6. Broyden CG (1970) The convergence of a class of double-rank minimization algorithms. I: general considerations. J Inst Math 6:76–90
7. Dai Y, Yuan Y (1998) Convergence properties of the beale-powell restart algorithm
8. Dai Y-H, Liao L-Z, Li D (2004) On restart procedures for the conjugate gradient method. Numer Algorithms 35(1–4):249–260
9. Das S, Abraham A, Chakraborty UK, Konar A (2009) Differential evolution using a neighborhood-based mutation operator. IEEE Trans Evol Comput 13(3):526–553
10. Davidon W (1991) Variable metric methods for minimization. SIAM J Optim 1:1–17

11. Deb K, Anand A, Joshi D (2002) A computationally efficient evolutionary algorithm for real-parameter optimization. Evol Comput 10(4):371–395
12. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. Evol Comput IEEE Trans 6(2):182–197
13. Duarte AR, Ribeiro CC, Urrutia S (2007) A hybrid ILS heuristic to the referee assignment problem with an embedded MIP strategy. In: Hybrid Metaheuristics, pp 82–95
14. Eiben AE, Hinterding R, Michalewicz Z (1999) Parameter control in evolutionary algorithms. IEEE Trans Evol Comput 3(2): 124–141
15. Emmerich MTM, Deutz AH, Beume N (2007) Gradient-based/evolutionary relay hybrid for computing pareto front approximations maximizing the S-metric. In: Hybrid Metaheuristics, pp 140–156
16. Fliege J, Drummond LMG, Svaiter B (2008) Newton's method for multiobjective optimization. Optimization Online
17. Fliege J, Svaiter B (2000) Steepest descent methods for multicriteria optimization. Math Methods of Oper Res 51(3):479–494
18. Fogel DB (1966) Artificial intelligence through simulated evolution. Wiley, New York
19. García S, Molina D, Lozano M, Herrera F (2009) 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 15:617–644
20. Beyer HG, Arnold DV (2003) Qualms regarding the optimality of cumulative path length control in CSA/CMA-evolution strategies. Evol Comput 11(1):19–28
21. Glover F, Laguna M (1997) Tabu search. Kluwer Academic Publishers, Boston
22. Griewank A (1981) Generalized decent for global optimization. JOTA 34:11–39
23. Gutin G, Karapetyan D (2009) A selection of useful theoretical tools for the design and analysis of optimization heuristics. Memet Comput 1(1):25–34
24. Herrera F, Lozano M (2000) Two-loop real-coded genetic algorithms with adaptive control of mutation step sizes. Appl Intell 13(3):187–204
25. Herrera F, Lozano M, Verdegay JL (1998) Tackling real-coded genetic algorithms: operators and tools for behavioural analysis. Artif Intell Rev 12:265–319
26. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
27. Katayama K, Narihisa H (1999) Iterated local search approach using genetic transformation to the traveling salesman problem. In: Proceedings of the 1st annual conference on Genetic and evolutionary computation (GECCO). New York, NY, USA, ACM
28. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, pp 1942–1948
29. Koch P, Kramer O, Rudolph G, Beume N (2009) On the hybridization of SMS-EMOA and local search for continuous multiobjective optimization. In: Proceedings of the 11th annual conference on Genetic and evolutionary computation (GECCO). New York, NY, USA, ACM, pp 603–610
30. Koumoutsakos P, Muller SD (2006) Flow optimization using stochastic algorithms. Lect Notes Control Inf Sci 330:213–229
31. Kramer O (2008) Self-adaptive heuristics for evolutionary computation. Springer, Berlin
32. Kramer O (2009) Fast blackbox optimization: iterated local search and the strategy of powell. In: Proceedings of GEM 2009. CSREA Press, pp 159–163
33. Land AH, Doig AG (1960) An automatic method of solving discrete programming problems. Econometrica 28(3):497–520
34. Le MN, Ong Y-S, Jin Y, Sendhoff B (2009) Lamarckian memetic algorithms: local optimum and connectivity structure analysis. Memet Comput 1(3):175–190
35. Lewis R, Torczon V, Trosset M (2000) Direct search methods: then and now. J Comput Appl Math 124(1–2):191–207
36. Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. Evol Comput IEEE Trans 10(3):281–295
37. Lourenço HR, Martin O, Stützle T (2001) A beginner's introduction to iterated local search. In: Proceedings of the fourth metaheuristics conference, vol. 2, Porto, Portugal, pp 1–6
38. Lourenço HR, Martin O, Stützle T (2003) Iterated local search. In: Handbook of Metaheuristics, pp 321–352
39. Lozano M, Herrera F, Krasnogor N, Molina D (2004) Real-coded memetic algorithms with crossover hill-climbing. Evol Comput 12(3):273–302
40. Martínez SZ, Coello CAC (2008) A proposal to hybridize multi-objective evolutionary algorithms with non-gradient mathematical programming techniques. In: Proceedings of the 10th international conference on parallel problem solving from nature (PPSN). Springer, Berlin, Heidelberg, pp 837–846
41. Mersch B, Glasmachers T, Meinicke P, Igel C (2006) Evolutionary optimization of sequence kernels for detection of bacterial gene starts. In: ICANN (2), pp 827–836
42. Meyer-Nieberg S, Beyer H-G (2007) Self-adaptation in evolutionary algorithms. In: Lobo FG, Lima CF, Michalewicz Z (eds) Parameter setting in evolutionary algorithms. Springer, Berlin
43. Mladenovic N, Drazic M, Kovacevic-Vujcic V, Cangalovic M (2008) General variable neighborhood search for the continuous optimization. Eur J Oper Res 191(3):753–770
44. Mladenovic N, Hansen P (1997) Variable neighborhood search. Comput Oper Res 24:1097–1100
45. Nelder J, Mead R (1964) A simplex method for function minimization. Comput J 7:308–313
46. Neri F, Tirronen V (2009) Scale factor local search in differential evolution. Memet Comput 1(2):153–171
47. Nguyen QH, Ong YS, Lim MH (2009) A probabilistic memetic framework. Evol Comput IEEE Trans 13(3):604–623
48. Noman N, Iba H (2008) Accelerating differential evolution using an adaptive local search. IEEE Trans Evol Comput 12(1):107–125
49. Ong Y-S, Keane AJ (2004) Meta-lamarckian learning in memetic algorithms. IEEE Trans Evol Comput 8(2):99–110
50. Ong Y-S, Lim M-H, Zhu N, Wong K-W (2006) Classification of adaptive memetic algorithms: a comparative study. IEEE Transactions on Systems, Man, and Cybernetics, part b: Cybernetics 36(1)
51. Ostermeier A, Gawelczyk A, Hansen N (1994) A derandomized approach to self adaptation of evolution strategies. Evol Comput 2(4):369–380
52. Ostermeier A, Gawelczyk A, Hansen N (1995) A derandomized approach to self adaptation of evolution strategies. Evol Comput 2(4):369–380
53. Li LP, Wang L (2009) Hybrid algorithms based on harmony search and differential evolution for global optimization. In: GEC '09: Proceedings of the first ACM/SIGEVO summit on genetic and evolutionary computation. ACM, New York, NY, USA, pp 271–278
54. Powell M (1964) An efficient method for finding the minimum of a function of several variables without calculating derivatives. Comput J 7(2):155–162
55. Powell MJD (1977) Restart procedures for the conjugate gradient method. Math Program V12(1):241–254
56. Price KV, Storn RM, Lampinen JA (2005) Differential evolution. A practical approach to global optimization natural computing series. Springer, Berlin

57. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13(2):398–417
58. Raidl GR (2006) A unified view on hybrid metaheuristics. In: Almeida F, Aguilera MJB, Blum C (eds) Hybrid Metaheuristics, Lecture Notes in Computer Science vol. 4030. Springer, pp 1–12
59. Rechenberg I (1973) Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution. Frommann-Holzboog, Stuttgart
60. Rosenbrock H (1960) An automatic method for finding the greatest or least value of a function. Comput J 3(3):175–184
61. Sánchez AM, Lozano M, Villar P, Herrera F (2009) Hybrid crossover operators with multiple descendents for real-coded genetic algorithms: combining neighborhood-based crossover operators. Int J Intell Syst 24(5):540–567
62. Schwefel H-P (1975) Evolutionsstrategie und numerische Optimierung. PhD thesis, TU Berlin
63. Schwefel H-P (1995) Evolution and optimum seeking. Sixth-generation computer technology. Wiley Interscience, New York
64. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: Proceedings of the international conference on evolutionary computation, pp 69–73
65. Shukla PK (2007) Gradient based stochastic mutation operators in evolutionary multi-objective optimization. In: Adaptive and natural computing algorithms, pp 58–66
66. Sindhya K, Deb K, Miettinen K (2008) A local search based evolutionary multi-objective optimization approach for fast and accurate convergence. In: Proceedings of the 10th international conference on parallel problem solving from nature (PPSN). Springer, Berlin, Heidelberg, pp 815–824
67. Stützle T (1999) Local search algorithms for combinatorial problems: analysis, improvements, and new applications, DISKI vol. 220. Infix Publishers, Sankt Augustin, Germany
68. Stützle T, Hoos HH (1999) Analyzing the run-time behaviour of iterated local search for the TSP. In: III Metaheuristics international conference. Kluwer Academic Publishers
69. Stutzle T (2006) Iterated local search for the quadratic assignment problem. Eur J Oper Res 174(3):1519–1539
70. Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Yp, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical Report, Nanyang Technological University
71. Talbi E-G (2002) A taxonomy of hybrid metaheuristics. J Heuristics 8(5):541–564
72. Ting C-K, Ko C-F, Huang C-H (2009) Selecting survivors in genetic algorithm using tabu search strategies. Memet Comput 1(3):191–203
73. Toksari MD, Güner E (2007) Solving the unconstrained optimization problem by a variable neighborhood search. J Math Anal Appl 328(2):1178–1187
74. Vrugt JA, Robinson BA, Hyman JM (2009) Self-adaptive multimethod search for global optimization in real-parameter spaces. IEEE Trans Evol Comput 13(2):243–259
75. Yu X, Tang K, Chen T, Yao X (2009) Empirical analysis of evolutionary algorithms with immigrants schemes for dynamic optimization. Memet Comput 1(1):3–24