Firefly Algorithm for Continuous Constrained Optimization Tasks

Szymon Łukasik and Sławomir Żak

Systems Research Institute, Polish Academy of Sciences
slukasik@ibspan.waw.pl
slzak@ibspan.waw.pl

Abstract. The paper provides an insight into the improved novel metaheuristics of the Firefly Algorithm for constrained continuous optimization tasks. The presented technique is inspired by social behavior of fireflies and the phenomenon of bioluminescent communication. The first part of the paper is devoted to the detailed description of the existing algorithm. Then some suggestions for extending the simple scheme of the technique under consideration are presented. Subsequent sections concentrate on the performed experimental parameter studies and a comparison with existing Particle Swarm Optimization strategy based on existing benchmark instances. Finally some concluding remarks on possible algorithm extensions are given, as well as some properties of the presented approach and comments on its performance in the constrained continuous optimization tasks.

Key words: firefly algorithm, constrained continuous optimization, swarm intelligence, metaheuristics

1 Introduction

Fireflies, also called lighting bugs, are one of the most special and fascinating creatures in nature. These nocturnal luminous insects of the beetle family *Lampyridae* (order *Coleoptera*), inhabit mainly tropical and temperate regions, and their population is estimated at around 1900 species [1]. They are capable of producing light thanks to special photogenic organs situated very close to the body surface behind a window of translucent cuticle [2]. Bioluminescent signals are known to serve as elements of courtship rituals, methods of prey attraction, social orientation or as a warning signal to predators (in case of immature firefly forms commonly referred to as glowworms). The phenomenon of firefly glowing is an area of continuous research considering both its biochemical [3] and social aspects [4].

Mechanisms of firefly communication via luminescent flashes and their synchronization has been imitated effectively in various techniques of wireless networks design [5], dynamic market pricing [6] and mobile robotics [7]. Firefly Algorithm developed recently by Xin-She Yang at Cambridge University and presented in the Chapter 8 of monograph [8] follows this approach. This swarm

intelligence optimization technique is based on the assumption that solution of an optimization problem can be perceived as agent (firefly) which "glows" proportionally to its quality in a considered problem setting. Consequently each brighter firefly attracts its partners (regardless of their sex), which makes the search space being explored more efficiently. Similar nature inspired metaheuristics include: Particle Swarm Optimization (PSO) [9] or Artificial Bee Colony optimization technique (ABC) [10].

This paper is devoted to the detailed study of Firefly Algorithm (FA), its experimental evaluation and possible improvements. It is organized as follows. In the next Section a comprehensive review of the existing FA scheme based on monograph [8] is given as well as some proposals for its extension. The subsequent part of the paper contains the results of parameter studies and some guidelines for its proper assignment. Next the comparison with Particle Swarm Optimization technique is performed. Ultimately, the final part of the paper presents some concluding remarks and suggestions for future work in the subject.

2 Firefly Algorithm in Practice

2.1 FA Scheme

Let us consider continuous constrained optimization problem where the task is to minimize cost function $f(x)$ for $x \in S \subset \mathbb{R}^n$ i.e. find x^* such as:

$$f(x^*) = \min_{x \in S} f(x) \; . \tag{1}$$

Assume that there exists a swarm of m agents (fireflies) solving abovementioned problem iteratively and x_i represents a solution for a firefly i in algorithm's iteration k, whereas $f(x_i)$ denotes its cost. Initially all fireflies are dislocated in S (randomly or employing some deterministic strategy). Each firefly has its distinctive attractiveness β which implies how strong it attracts other members of the swarm. As a firefly attractiveness one should select any monotonically decreasing function of the distance $r_j = d(x_i, x_j)$ to the chosen firefly j, e.g. the exponential function:

$$\beta = \beta_0 e^{-\gamma r_j} \tag{2}$$

where β_0 and γ are predetermined algorithm parameters: maximum attractiveness value and absorption coefficient, respectively [8]. Furthermore every member of the swarm is characterized by its light intensity I_i which can be directly expressed as a inverse of a cost function $f(x_i)$. To effectively explore considered search space S it is assumed that each firefly i is changing its position iteratively taking into account two factors: attractiveness of other swarm members with higher light intensity i.e. $I_j > I_i, \forall j = 1, ...m, j \neq i$ which is varying across distance and a fixed random step vector u_i. It should be noted as well that if no brighter firefly can be found only such randomized step is being used [8].

To summarize, when taking into consideration all above statements, the algorithm scheme established in [8] can be presented in the following pseudo-code form:

Firefly Algorithm for Constrained Optimization

Input:
$f(z)$, $z = [z_1, z_2, ..., z_n]^T$ {*cost function*}
$S = [a_k, b_k], \forall k = 1, ..., n$ {*given constraints*}
$m, \beta_0, \gamma, \min u_i, \max u_i$ {*algorithm's parameters*}
Output:
$x_{i^{min}}$ {*obtained minimum location*}

begin
  for *i=1* **to** m **do**
    $x_i \leftarrow$ `Generate_Initial_Solution` ()
  end
  repeat
    $i^{min} \leftarrow \arg\min_i f(x_i)$
    $x_{i^{min}} \leftarrow \arg\min_{x_i} f(x_i)$
    for *i=1* **to** m **do**
      for *j=1* **to** m **do**
        if $f(x_j) < f(x_i)$ **then** {*move firefly i towards j*}
          $r_j \leftarrow$ `Calculate_Distance` (x_i, x_j)
          $\beta \leftarrow \beta_0 e^{-\gamma r_j}$ {*obtain attractiveness*}
          $u_i \leftarrow$ `Generate_Random_Vector` $(\min u_i, \max u_i)$
          for *k=1* **to** n **do**
            $x_{i,k} \leftarrow (1-\beta)x_{i,k} + \beta x_{j,k} + u_{i,k}$
          end
        end
      end
    end
    $u_{i^{min}} \leftarrow$ `Generate_Random_Vector` $(\min u_i, \max u_i)$
    for *k=1* **to** n **do**
      $x_{i^{min},k} \leftarrow x_{i^{min},k} + u_{i^{min},k}$ {*best firefly should move randomly*}
    end
  until *stop condition true*
end

In the next part of the paper some technical details of the algorithm will be considered. A closer look will be taken at such important issues as the sensitivity of the parameters, their influences on the convergence rate of the algorithm, the potential improvement, and further development. For more detailed theoretical considerations, MATLAB code and convincing two dimensional demonstrations of the algorithm performance one could refer to the pioneer publication already mentioned [8].

2.2 Technical Details

The algorithm presented here makes use of a synergic local search. Each member of the swarm explores the problem space taking into account results obtained by others, still applying its own randomized moves as well. The influence of other solutions is controlled by value of attractiveness (2). It can be adjusted by modifying two parameters: its maximum value β_0 and an absorption coefficient γ.

The first parameter describes attractiveness at $r_j = 0$ i.e. when two fireflies are found at the same point of search space S. In general $\beta_0 \in [0, 1]$ should be used and two limiting cases can be defined: when $\beta_0 = 0$, that is only non-cooperative distributed random search is applied and when $\beta_0 = 1$ which is equivalent to the scheme of cooperative local search with the brightest firefly strongly determining other fireflies positions, especially in its neighborhood [8].

On the other hand, the value of γ determines the variation of attractiveness with increasing distance from communicated firefly. Using $\gamma = 0$ corresponds to no variation or constant attractiveness and conversely setting $\gamma \to \infty$ results in attractiveness being close to zero which again is equivalent to the complete random search. In general $\gamma \in [0, 10]$ could be suggested [8]. It is more convenient, however, to derive γ value specifically for the considered problem. Such customized absorption coefficient should be based on the "characteristic length" of the optimized search space. It is proposed here to use:

$$\gamma = \frac{\gamma_0}{r_{max}} \tag{3}$$

or:

$$\gamma = \frac{\gamma_0}{r_{max}^2} \tag{4}$$

wheras $\gamma_0 \in [0, 1]$ and:

$$r_{max} = \max d(x_i, x_j), \forall x_i, x_j \in S \; . \tag{5}$$

Efficiency of both techniques introduced here will be experimentally evaluated in the next Section.

Finally one has to set random step size i.e. its lower and upper bounds ($\min u_i$, $\max u_i$) and define the method of its generation. In [8] it was proposed to use $\min u_i = -0.5\alpha$ and $\max u_i = 0.5\alpha$, with $\alpha \in [0, 1]$ being algorithm's parameter. In consequence u_i for each search space dimension k is supposed to be generated according to:

$$u_{i,k} = \alpha(rand - \frac{1}{2}) \; . \tag{6}$$

with $rand \sim U(0, 1)$ - a random number obtained from the uniform distribution. Here it is suggested to use alternative approach i.e. to define random vector as a fraction of firefly distance to search space boundaries:

$$u_{i,k} = \begin{cases} \alpha \, rand_2(b_k - x_{i,k}) & if \;\; \mathrm{sgn}(rand_1 - 0.5) < 0 \\ -\alpha \, rand_2(x_{i,k} - a_k) & if \;\; \mathrm{sgn}(rand_1 - 0.5) \geq 0 \end{cases} \tag{7}$$

with two uniform random numbers $rand_1$, $rand_2$ obtained similarly as above.

In the end it could be noted that computational complexity of the algorithm under consideration is $O(m^2)$, so using larger population size leads to substantial increase in calculation time. It can, however, bring significant benefits in terms of algorithm's performance, especially when some deterministic technique of initial swarm displacement is being employed. In the paper, simple random dislocation of fireflies in S, instead of such strategy, is being assumed.

3 Numerical Experiments and Parameter Studies

The performance of the presented technique was verified experimentally using its MATLAB implementation and a set of 14 benchmark problems (for a detailed list please refer to the Appendix). All tests were conducted for a fixed number of algorithm iterations l and repeated in 100 independent trials with different random number generator seeds. As problems are characterized by different scales on the cost function it was more convenient to use ranking of different algorithm's variants instead of direct analysis of quality indexes $|f_{min} - f(x_{i^{min}})|$. It means that each problem was considered separately with tested configurations being ranked by their performance. Then the final comparison was carried out using medians of obtained ranks. Due to space limitations only most representative results are presented in the paper. The full set of simulation data can be found on the first author's web site (`http://www.ibspan.waw.pl/~slukasik`).

3.1 Population Size

Firstly, the influence of swarm size on the algorithm efficiency was analyzed. For such purpose a fixed number of cost function evaluations (2000) and FA variants with $m = \{6, 8, 10, ..., 200\}$ were assumed and suitably decreasing number of iterations l were compared. To make such a comparison more representative variants characterized by significant rounding errors of $m\,l$ product i.e. $|\lfloor m\,l \rfloor - m\,l| > 20$ were rejected. All tests were conducted for $\beta_0 = 1$, $\alpha = 0.01$ and fixed $\gamma = 1$. Although general remarks on optimal number of fireflies cannot be made, two tendencies can be observed. For difficult optimization tasks, such as instances no 2, 3, 4, 8, 9, 13, 14 it is always a better option to use a maximum number of fireflies. It is an observation based on medians of ranks for those problems depicted on Fig. 1. Still, there exists a set of relatively easy optimization problems like 1, 5, 6, 7, 10, 11, 12, 13 where optimal number of fireflies could be found, e.g. the problem of Sphere function minimization is solved most effectively by a set of 28 fireflies as shown on Fig. 2.

The aforementioned remarks should be accompanied by an additional look on associated calculation time which, as noted before, increases significantly with size of the swarm. Taking it into account it is advisable to use reasonable population of 40-50 fireflies and refrain from applying such rule only for more complicated optimization tasks. It is worth noting that similar remarks have been made in [9] with reference to the Particle Swarm Optimization technique.

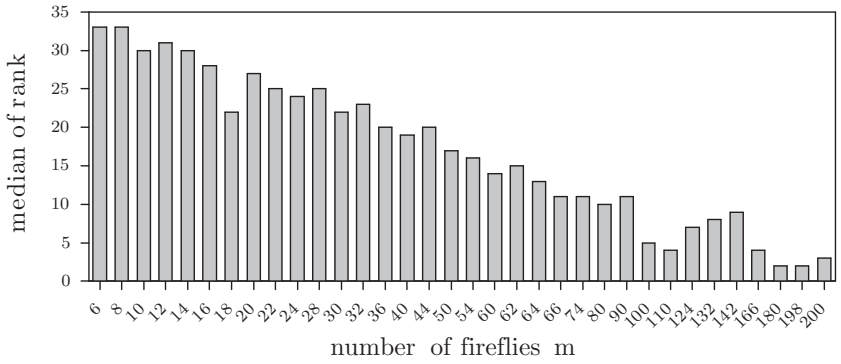

Fig. 1. Median of performance ranks for varying population size (problems no 2, 3, 4, 8, 9, 13, 14

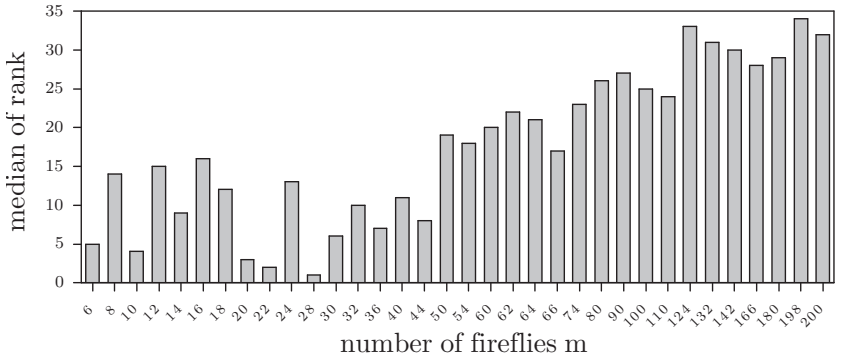

Fig. 2. Median of performance ranks for varying population size (problem: 12)

3.2 Maximum of Attractiveness Function

In the second series of computational experiments the influence of β_0 value on the algorithm performance was studied. Testing runs were conducted for $\beta_0 = \{0, 0.1, 0.2, ..., 1.0\}$ with other parameters fixed i.e. $m = 40, l = 250, \alpha = 0.01$ and $\gamma = 1$. Again, each of the tested configurations was ranked by its performance in the considered problem and median of such rank is reported (see Fig. 3).

It is observed that the best option is to use maximum attractiveness value $\beta_0 = 1$ which implies the strongest dependence of fireflies' positions on their brighter neighbors location.

3.3 Absorption Coefficient and Random Step Size

Finally changes in the algorithm's performance with varying absorption coefficient γ and random step size α were under investigation. Maximum attractiveness $\beta_0 = 1$ was used, with population size $m = 40$ and iteration number $l = 250$.

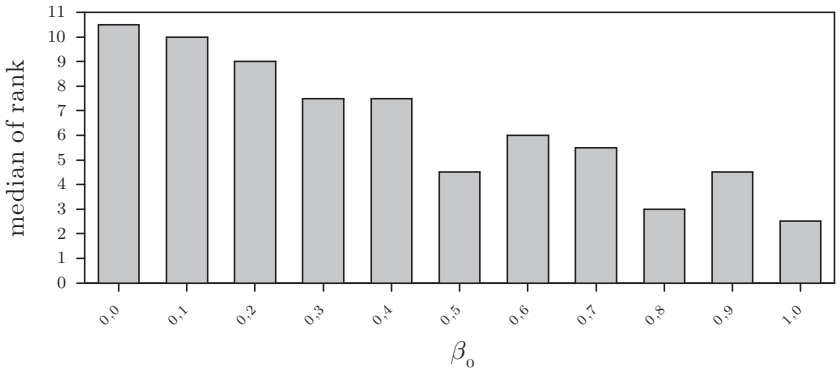

Fig. 3. Median of performance ranks with varying maximum of attractiveness function

Firefly Algorithm variants with $\alpha = \{0.001, 0.01, 0.1\}$ and $\gamma = \{0.1, 1.0, 10.0\}$ were tested. Additionally two problem-related techniques of obtaining absorption coefficient (Eq. (3) and (4)) were considered (with $\gamma_0 = \{0.1, 0.2, ..., 1.0\}$), so the overall number of examined configurations reached 75.

The obtained results indicate that for the examined optimization problems variants of the algorithm with $\alpha = 0.01$ are the best in terms of performance. Furthermore it could be advisable to use adaptable absorption coefficient according to (3) with $\gamma_0 = 0.8$ as this configuration achieved best results in the course of executed test runs. Although proposed technique of γ adaptation in individual cases often performs worse than fixed γ values it has an advantage to be automatic and "tailored" to the considered problem.

4 Comparison with Particle Swarm Optimization

Particle Swarm Optimization is a swarm-based technique introduced by Kennedy and Eberhart [11]. It has been intensively developed recently with research studies resulting in numerous interesting both theoretical and practical contributions [9]. The Particle Swarm Optimizer was studied in continuous optimization context in [12], with suggested variants and recommended parameter values being explicitly given.

Experiments reported here involved a performance comparison of Firefly Algorithm with such advanced PSO algorithm defined with constriction factor and the best parameters set suggested in conclusion of [12]. Both algorithm were executed with the same population size $m = 40$, iteration number $l = 250$ and the test was repeated 100 times for its results to be representative. The obtained results are presented in Tab. 1. It contains algorithms' performance indices given as average difference between a result obtained by both techniques $f(x_{i^{min}})$ and actual minimum of cost function $f(x^*)$ with standard deviations given for reference. For the Firefly Algorithm results obtained by the best configuration

selected in Section 3.3 are presented, as well as the best result obtained for each test problem by one of the 75 algorithm variants considered in the same Section.

Table 1. Performance comparison of Firefly Algorithm and Particle Swarm Optimization technique

Problem	$avg.\|f(x^*) - f(x_{imin})\| \pm std.dev.\|f(x^*) - f(x_{imin})\|$		
	PSO	FA($\gamma_0 = 0.8$)	FA(best)
1	5.75E-21 ± 5.10E-02	2.35E-04 ± 3.76E-04	1.01E-04 ± 1.78E-04
2	4.98E+02 ± 1.59E+02	6.59E+02 ± 2.25E+02	4.14E+02 ± 2.98E+02
3	0.00E+00 ± 0.00E+00	1.42E-01 ± 3.48E-01	3.05E-02 ± 1.71E-01
4	3.04E+01 ± 1.09E+01	1.63E+01 ± 5.78E+00	2.78E+00 ± 5.39E-01
5	4.63E-02 ± 2.59E-02	1.56E-01 ± 4.56E-02	1.48E-01 ± 4.17E-02
6	2.31E-01 ± 6.17E-01	1.10E+00 ± 6.97E-01	5.66E-01 ± 3.36E-01
7	1.16E-17 ± 6.71E-17	7.90E-05 ± 6.82E-05	4.14E-05 ± 2.17E-05
8	2.15E-06 ± 6.51E-15	2.80E-02 ± 7.78E-02	2.21E-03 ± 1.11E-03
9	5.84E-02 ± 5.99E-02	2.18E-01 ± 1.67E-01	2.18E-02 ± 2.33E-02
10	2.86E+00 ± 3.52E+00	3.09E+00 ± 3.72E+00	1.69E-01 ± 1.08E+00
11	4.00E-18 ± 9.15E-18	1.63E-06 ± 1.60E-06	1.44E-06 ± 1.47E-06
12	7.31E-22 ± 1.61E-21	1.59E-06 ± 1.73E-06	6.17E-07 ± 6.23E-07
13	4.61E+00 ± 1.09E-01	6.46E-03 ± 6.46E-02	2.75E-06 ± 5.24E-06
14	3.63E-03 ± 6.76E-03	2.59E-02 ± 3.96E-02	1.00E-02 ± 1.28E-02

It is noticeable that Firefly Algorithm is outperformed repeatedly by Particle Swarm Optimizer (PSO was found to perform better for 11 benchmark instances out of 14 being used). It is also found to be less stable in terms of standard deviation. It is important to observe though that the advantage of PSO is vanishing significantly (to 8 instances for which PSO performed better) when one relates it to the best configuration of firefly inspired heuristic algorithm. Consequently, such comparison should be repeated after further steps in the development of the algorithm being described here will be made. Some possible improvements contributing to the Firefly Algorithm performance progress will be presented in the final Section of the paper.

5 Conclusion

Firefly Algorithm described here could be considered as an unconventional swarm-based heuristic algorithm for constrained optimization tasks. The algorithm constitutes a population-based iterative procedure with numerous agents (perceived as fireflies) concurrently solving a considered optimization problem. Agents communicate with each other via bioluminescent glowing which enables them to explore cost function space more effectively than in standard distributed random search.

Most heuristic algorithms face the problem of inconclusive parameters settings. As shown in the paper, coherent suggestions considering population size

and maximum of absorption coefficient could be derived for the Firefly Algorithm. Still the algorithm could benefit from additional research in the adaptive establishment of absorption coefficient and random step size. Furthermore some additional features like decreasing random step size and more sophisticated procedure of initial solution generation could bring further improvements in the algorithm performance. The algorithm could be hybridized together with other heuristic local search based technique like Adaptive Simulated Annealing [13]. Firefly communication scheme should be exploited then on the higher level of the optimization procedure.

Acknowledgments. Authors would like to express their gratitude to Dr Xin-She Yang of Cambridge University for his suggestions and comments on the initial manuscript.

Appendix: Benchmark Instances

No.	Name	n	S	$f(x^*)$	Remarks
1	Himmelblau [14]	2	(-6,6)	0	four identical local minima
2	Schwefel [15]	10	(-500,500)	0	several local minima
3	Easom [16]	2	(-100,100)	-1	a singleton maximum in a horizontal valley
4	Rastrigin [17]	20	(-5.12,5.12)	0	highly multimodal and difficult
5	Griewank [18]	5	(-600,600)	0	several local minima
6	Rosenbrock [19]	4	(-2.048,2.048)	0	long curved only slightly decreasing valley, unimodal
7	Permutation [20]	2	(-2,2)	0	function parameter β=10
8	Hartman3 [21]	3	(0,1)	-3.862	4 local minima
9	Hartman6 [21]	6	(0,1)	-3.322	6 local minima
10	Shekel [22]	4	(0,10)	-10.536	10 local minima
11	Levy 10 [23]	5	(-10,10)	0	10^5 local minima
12	Sphere [24]	3	(-5.12,5.12)	0	unimodal
13	Michalewicz [25]	5	(0,π)	-4.687	$n!$ local minima
14	Powersum [26]	4	(0,2)	0	singular minimum among very flat valleys

References

1. Encyclopædia Britannica: Firefly. In: Encyclopædia Britannica. Ultimate Reference Suite. Chicago: Encyclopædia Britannica (2009)
2. Babu, B.G., Kannan, M.: Lightning bugs. Resonance **7**(9) (2002) 49–55
3. Fraga, H.: Firefly luminescence: A historical perspective and recent developments. Journal of Photochemical & Photobiological Sciences **7** (2008) 146–158
4. Lewis, S., Cratsley, C.: Flash signal evolution, mate choice, and predation in fireflies. Annual Review of Entomology **53** (2008) 293–321

5. Leidenfrost, R., Elmenreich, W.: Establishing wireless time-triggered communication using a firefly clock synchronization approach. In: Proceedings of the 2008 International Workshop on Intelligent Solutions in Embedded Systems. (2008) 1–18
6. Jumadinova, J., Dasgupta, P.: Firefly-inspired synchronization for improved dynamic pricing in online markets. In: Proceedings of the 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems. (2008) 403–412
7. Krishnanand, K., Ghose, D.: Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications. Multiagent and Grid Systems **2**(3) (2006) 209–222
8. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press (2008)
9. Eberhart, R.C., Shi, Y.: Computational Intelligence: Concepts to Implementations. Morgan Kaufmann (2007)
10. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization **39**(3) (2007) 459–471
11. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Neural Networks, 1995. Proceedings., IEEE International Conference on. (1995) 1942–1948 vol.4
12. Schutte, J.F., Groenwold, A.A.: A study of global optimization using particle swarms. Journal of Global Optimization **31**(1) (2005) 93–108
13. Ingber, L.: Adaptive simulated annealing (ASA): lessons learned. Control & Cybernetics **25**(1) (1996) 33–55
14. Himmelblau, D.M.: Applied Nonlinear Programming. McGraw-Hill (1972)
15. Schwefel, H.P.: Numerical Optimization of Computer Models. John Wiley & Sons, Inc. (1981)
16. Easom, E.: A survey of global optimization techniques. Master's thesis, University of Louisville (1990)
17. Mühlenbein, H., Schomisch, D., Born, J.: The Parallel Genetic Algorithm as Function Optimizer. Parallel Computing **17**(6-7) (1991) 619–632
18. Griewank, A.: Generalized descent for global optimization. Journal of Optimization Theory and Applications **34** (1981) 11–39
19. Rosenbrock, H.H.: State-Space and Multivariable Theory. Thomas Nelson & Sons Ltd (1970)
20. Neumaier, A.: Permutation function. `http://www.mat.univie.ac.at/~neum/glopt/my_problems.html`
21. Törn, A., Žilinskas, A.: Global Optimization. Springer (1989)
22. Shekel, J.: Test functions for multimodal search techniques. In: Proceedings of the 5th Princeton Conference on Infomration Science and Systems. (1971) 354–359
23. Jansson, C., Knüppel, O.: Numerical results for a self-validating global optimization method. Technical Report 94.1, Technical University of Hamburg-Harburg (1994)
24. Bilchev, G., Parmee, I.: Inductive search. In: Proceedings of IEEE International Conference on Evolutionary Computation. (1996) 832–836
25. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer (1998)
26. Neumaier, A.: Powersum function. `http://www.mat.univie.ac.at/~neum/glopt/my_problems.html`