HEURISTICS OPTIMISATION OF NUMERICAL FUNCTIONS

Kalin Penev
Southampton Solent University, Kalin.Penev@solent.ac.uk , UK

Abstract: The article presents an investigation of heuristic behaviour of search algorithms applied to numerical problems. The aim is to compare the abilities of Particle Swarm Optimisation, Differential Evolution and Free Search to adapt to variety of search spaces without the need for constant re-tuning of algorithms parameters. The article focuses on several advanced characteristics of Free Search and attempts to clarify specifics of its behaviour. The achieved experimental results are presented and discussed.
Key words: Adaptive Computing, Swarm Intelligence, Free Search, Optimisation

1. INTRODUCTION

The objective of this article is to introduce an algorithm called Free Search (FS) [16][17][19]. According to the discussion on the heuristic methods in the literature [13], Free Search can be classified as heuristic method based on trials and errors rather than comprehensive theory. It can be likened to the heuristic behaviour of variety of animals in nature, where they day by day explore surrounding environment trying to find some favour. They learn from the trials and refine their behaviour. FS models walking, as discrete steps within continuous space. FS generates a new solution as deviation of a current one $x = x_0 + \Delta x$. Where x is a new solution, x_0 is a current location and Δx is modification strategy. x, x_0 and Δx are vectors of real numbers. FS modifies all current solutions, which is similar to Evolutionary Programming (EP) [9], Particle Swarm Optimisation (PSO) [5] and Differential Evolution (DE) [20]. The individuals are called animals by analogy with real animals in nature. In pursuing the objective of the algorithm each individual generates and examines its action with own extent of uncertainty, taking into account the search space limits, constraints, and acquired from the previous generations knowledge. The population in Free Search can be considered as a team of individuals, which exchange knowledge and experience in contrast to the herd behaviour of a swarm or a flock. Presented in the article experimental evaluation contributes to the assessment of the reliability and to the abilities for adaptation to unknown problems. Free Search is compared to other evolutionary algorithms such as PSO [6] and DE [21].

2. FREE SEARCH

This section refines the description of the algorithm, published earlier [18]. It focuses on the abilities for adaptation and self-regulation. It attempts to clarify how a computational program could models processes similar to process of thinking. The algorithm architecture is presented in Figure 1 as a flow chart and as an example in pseudo-code. The structure of the algorithm consists of three major events initialisation, exploration and termination.

Free Search starts with initialisation. The algorithm requires definition of initial values of the search space boundaries [$Xmin_i$, $Xmax_i$], population size m, limit for the number of explorations G, limit for the number of steps within one exploration walk T, minimal and maximal values for the neighbour space frame [$Rmin$, $Rmax$]. The maximal neighbour space guarantees coverage of the whole search space from one animal. The minimal neighbour space guarantees desired granularity of the coverage from one animal. For the algorithm $Rmin$ and $Rmax$ are absolute values. An appropriate definition of these values could support good performance across variety of problems without additional external adjustments. A prior determination of the neighbour space to concrete value for particular problem can lead to slightly better performance on that problem but aggravates the performance on other problems, which is in line with the existing general assessment of the performance of the optimisation algorithms [12][24]. FS requires definition of an initialisation strategy. Acceptable initialisation strategies are:

- random values: $x_{0ji} = Xmin_i + (Xmax_i - Xmin_i)*random_{ji}(0,1)$;
- certain values: $x_{0ji} = a_{ji}$, $a_{ji} \in [Xmin_i, Xmax_i]$;
- one location: $x_{0ji} = c_i$, $c_i \in [Xmin_i, Xmax_i]$,

$random(0,1)$ is a random value between 0 and 1, a_{ji} and c_i are constants.

The ability to operate with all these strategies also supports the good performance across variety of problems without additional external adjustments. Then each animal takes individual exploration walk. The exploration walk generates coordinates of a new location x_{tji} as:

$x_{tji} = x_{0ji} - \Delta x_{tji} + 2*\Delta x_{tji}*random_{tji}(0,1)$.

The modification strategy is: $\Delta x_{tji} = R_{ji} * (Xmax_i - Xmin_i) * random_{tji}(0,1)$, where $i = l$ for uni-dimensional step (l indicates one dimension), $i = 1,..,n$ for multi-dimensional step. T is step limit per walk. t is current step, $t = 1,..,T$. R_{ji} indicates the size of the idealised frame of the neighbour space for animal j within dimension i. $random_{tji}(0,1)$ generates random values between 0 and 1. Δx_{tji} indicates the actual size of the neighbour space for particular problem for step t of individual j within dimension i. The modification strategy is independent from the current or the best achievements. It guarantees individual behaviour different from swarm behaviour. The strategy allows nonzero probability for access to any location of the search space and highly encourages escaping from trapping in local sub-optima. A sufficiently large $R_{ji} > 1$ guarantees coverage of the entire search space with certain probability. The walk is followed by detached assessment of the locations, individually explored. The assessment, during the exploration, is modelled as follows:

$f_{tj} = f(x_{tji})$, $f_j = max(f_{tj})$, here f_{tj} is the value of the objective function achieved from animal j for step t. f_j is the quality of the location marked with pheromone from an animal after one exploration.

The pheromone generation is generalised for whole population: $P_j = f_j / max(f_j)$, where $max(f_j)$ is the best achieved value from the population for the exploration. This is a normalisation of the explored problem to a qualitative (or cognitive) space, in which the algorithm operates. This idealised space can be considered as a simplified model of the idealised space of notions in thought. The normalisation of any particular search space to one idealised space supports successful performance across variety of problems without additional external adjustments. The sensibility generation is: $S_j = Smin + \Delta S_j$, where $\Delta S_j = (Smax - Smin)*random_j(0,1)$

$Smin$ and $Smax$ are minimal and maximal possible values of the sensibility.

$Smin = Pmin$, $Smax = Pmax$. $Pmin$ and $Pmax$ are minimal and maximal possible values of the pheromone marks. The process continues with selection of a start location for the new exploration walk. The act of selection can be considered as a model of a decision-making in thought.

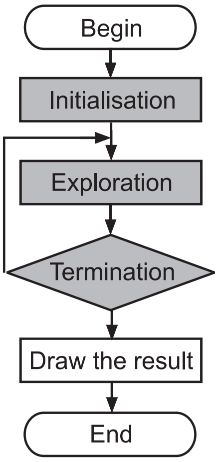
The ability for decision-making based on the achieved from the exploration (which can be in contradiction with the existing assumptions about the task during the implementation of the algorithm) supports good performance across variety of problems without additional external adjustments. The selection for a start location x_{0j} for an exploration walk is:

$x_{0j} = x_k$ $(P_k \geq S_j)$, where $j = 1,..,m$, $k = 1,..,m$, j is the animals number, k is the number of the location marked with pheromone, x_{0j} is the start location selected from animal number j.

After the exploration follows the event termination. Acceptable criteria for termination are:

- reaching the optimisation criterion: $fmax \geq fopt$, where $fmax$ is maximal achieved solution, $fopt$ is an acceptable value of the objective function.
- expiration of generation limit: $g \geq G$, where G is a limit and g - current values.
- complex criterion: $((fmax \geq fopt) \parallel (g \geq G))$.

In Figure 1 $Xmin_i$ and $Xmax_i$ are the search space boundaries, m is the population size, $j = 1,..,m$, $k = 1,..,m$, n is the number of dimensions, $i = 1,..,n$. j indicates explored locations. k indicates the locations marked with pheromone. T is step limit per walk. t is current step. R_{ji} is a variable frame for the neighbouring space $R_{ji} \in [Rmin, Rmax]$.



Free Search{
Initialisation{
initialise X_{maxi}, X_{mini}, n, m, G, x_{0ji}, T, R_{ji}
take initial walks, f_{tj} $(x_{0j}+\Delta x_{tj})$
generate an initial pheromone P_k
distribute the initial pheromone P_k
} // end initialisation
do{ // **exploration**
generate sensibility S_j
select start locations for a walk $x'_{0j} = x_k(S_j,P_k)$
take exploration walks f_{tj} $(x_{0j}+\Delta x_{tj})$
generate a pheromone P_k
distribute the pheromone P_k
} while $(f(x_j)<f_{opt}$ && g<G) // **termination**
Output the result }
// end Free Search

Figure 1. Free Search – algorithm architecture and pseudo code.

The Free Search structure is similar to the general description of the evolutionary algorithms [4][8]. The FS architecture is simplified and consists of generalised events initialisation, exploration and termination. However, the semantic of the events is enriched with new bio-inspired content. A major element of this content is sense. Sense is time dependent variable, adjusted during the process of search in an adaptable frame. The frame obtains certainty when the algorithm is applied to the concrete problem. The algorithm determines the sensibility during the process of search as function of the explored problem. The same algorithm makes different adjustments of the sense during the exploration of different problems. This is considered to be a model of adaptation.

The sensibility creates a specific interpretation of the individuals in FS called animals. They can be described by the abstraction – entities, which can move, which can evaluate by particular criteria locations from a search space, and which can indicate the quality of the evaluated locations as memory from previous activities. Then they can identify the indicators from previous activities, and can use them to decide where and how to move. The sense models individualism. Its value is unique for each animal and it plays a role of a tool for regulation of the divergence and the convergence within the search process and a tool for guiding of the space exploration. Three general states of the sensibility distribution can be considered – uniform, enhanced and reduced. Within a stochastic environment and during a stochastic process, it is assumed that any deviation could lead to non-uniform changes of the process. The achieved results play a role of deviator. The animals with enhanced sensibility will select and can differentiate more precisely locations marked with higher level of pheromone and will ignore the locations with lower level of pheromone. By reducing of the sensibility, the animal can be allowed to explore around locations marked with a low level of pheromone, which indirectly decreases the probability for selection of locations marked with high level of pheromone. The sense is an original concept implemented in Free Search. It has no analogue in other evolutionary population-based methods.

When related to the pheromone marks it can be interpreted as individual knowledge used in decision-making for selection of a location for search. This relation leads to adaptive self-regulation of the sense, the pheromone marks, and the action. How is this adaptive self-regulation organised? An achievement of better solutions increases the maximal value of the pheromone and enhances the maximal allowed sensibility of the animals. This is an adaptive regulation between pheromone and sensibility. In fact it is an abstract approach for learning.

The enhanced sensibility within the defined frame implicitly regulates the action by indirect reduction of the probability for selection of the locations marked with low level of pheromone. In this manner FS implements a computational model of abstraction, cognition, decision-making and action analogous perhaps to the processes of perception, learning and thinking in biological systems [18] [19]. The algorithm is implemented as a computer program. In this study FS together with DE and PSO are applied to the following tests.

3. TEST PROBLEMS

For the test are used search spaces defined by non-linear multi-modal functions. For all experiments the aim is to find the maximum therefore the test functions are transformed in a relevant manner.

Step sphere test function is proposed by Bäck [2]. It introduces plateaus to the topology, and excludes a local correlation of the space. Maximise:

$$f(x_i) = 10 - \sum_{i=1}^{n} \lfloor x_i + 0.5 \rfloor^2 ,$$ where $x_i \in$ [-2.5, 2.5].

For the 20 dimensional variant used in this study, the maximal are all locations, which belongs to the plateau $x_i \in$ [-0.5, 0.5). The maximum is f max =10.

Rastrigin test function is known from the literature [23].

$$f(x) = nA + \sum_{i=1}^{n} (x_i^2 - A\cos(2\pi x_i))$$, where A=10 and $-5.12<x_i<5.12$.

The maximum is $f(0) = 0$.

Norwegian test function test function is published on the domain of the Norwegian University of Science and Technology (therefore it is called Norwegian) by Brekke [3] and Federici [10].

$$\prod_{i=1}^{n} \left(\cos(\pi x_i^3) \left(\frac{99 + x_i}{100} \right) \right)$$, where search space borders are defined by $-1.1<x_i<1.1$.

The maximum is f(1.0)=1.0.

Himmelblau test function [11] has four maxima equal height (200) at (3.584, -1.848), (3.0, 2.0), (2.805, 3.1313) and (-3.779, -3.283).

$$f(x, y0 = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2$$

The search space is restricted to $-10 < x, y < 10$.

4. EXPERIMENTAL RESULTS

FS, PSO and DE are applied to the above-mentioned functions as follows – Each algorithm is evaluated four times per test function – (1) start from stochastic initial population with limit 100 iterations, (2) start from stochastic initial population with limit 2000 iterations, (3) start from one initial location with limit 100 iterations, (4) start from one initial location with limit 2000 iterations.

The single initial location is defined as: $x_0 = x_{min} + 0.1(x_{max} - x_{min})$.

Each evaluation is 320 experiments. Population size is 10 (ten) individuals for all algorithms for all experiments. DE and PSO are implemented according to the original models proposed in the literature [5][6][7][20][21]. To reduce probability for trapping and to enhance ability for adaptation DE and PSO are freed by variation of their parameters. For DE differential factor F varies from 0.5 to 1.5. For PSO inertia W varies from 0.5 to 1.5. Respectively for FS neighbour space R varies from 0.5 to 1.5. These configurations of DE, PSO and FS are applied to the tests described in section 3.

The results are accepted as successful if: for Step sphere test function equal to 10 (The optimum is 10 for $x_i \in$ [-0.5, 0.5)); for Himmelblau test function higher than 199.9, precision 0.0005; for Rastrigin test function higher than -0.1(the next high optima are less than –1.0); for

Norwegian test function higher than 0.99 (the next high optimum is less than –0.985);

	Start from stochastic locations						Start from single location					
	FS		DE		PSO		FS		DE		PSO	
	100	2000	100	2000	100	2000	100	2000	100	2000	100	2000
F1	94	320	85	218	25	185	15	320	0	0	0	0
F2	187	267	202	225	153	235	144	257	0	0	0	0
F3	34	253	4	6	10	15	22	269	0	0	0	0
F4	320	320	294	315	268	313	320	320	0	0	0	0
Overall	635	1160	585	764	456	748	501	1166	0	0	0	0

Table 1. Experimental results

In Table 1 F1 - 20 dimensional Step sphere test function. F2 - two-dimensional Rastrigin test function. F3 - two-dimensional Norwegian test function. F4 - two-dimensional Himmelblau test function.

5. CONCLUSIONS

The results from the experiments with a start from stochastic initial population demonstrate that the evaluated algorithms can adapt to the explored tests without external adjustments for given tests. The experiments confirm the excellent exploration abilities of Particle Swarm Optimisation and Differential Evolution published earlier in the literature [1][18][21][22]. For 100 iterations PSO can cope for 50% of the experiments with Rastrigin test function and DE and FS for more than 50%. However, the algorithms have difficulties in escaping from trapping in sub-optimal peaks. After 2000 iterations DE cannot reach the optimum for 30% PSO for 27% and FS for 17% from the experiments. The algorithms reach all four equal optima of the Himmelblau test function in 100 iterations for more than 80% of the experiments. It illustrates the excellent exploration abilities of PSO, DE and FS. Himmelblau test function is used in the literature for evaluation of clustering and niche methods [14]. FS outperforms DE and PSO on Norwegian test problem, which confirms that they have some difficulties in search near to the search space boundaries published earlier [18]. In Table 1 the results for strategy (2) for DE are presented [20]. For the experiments with a start from one location DE and PSO are not applicable. This is a consequence from the requirements for non-equal individuals for adaptive settings of the optimisation parameters from DE and PSO.

A higher number, of successful results achieved from FS, suggests that the algorithm could reach the best solution in fewer starts. The ability to find the optimum starting from one location is essential for solving heterogeneous tasks when a prior knowledge about the task does not exist. It gives a credit how reliable could be one algorithm applied to unknown problems. FS overcomes common disadvantages of existing evolutionary population-based algorithms such as dependence on the initial population and inability for orientation within the search space, which is supported with the experimental results. Free Search can advance a wide range of disciplines in the efforts to cope with complex, uncertain problems, such as engineering, physics, chemistry, economics, business, finance, and operations research. Further investigations can focus on evaluation with dynamic and time dependent search space, including implementation in autonomous systems. A pragmatic area for further research is application to real-world problems.

REFERENCES

1. Angeline P., 1998, Evolutionary Optimisation versus Particle Swarm Optimisation: Philosophy and Performance Difference, The 7-th Annual Conference on Evolutionary Programming, San Diego, USA.

2. Bäck T., 1996, Evolutionary Algorithms in Theory and Practice, ISBN: 0195099710, New York: Oxford University Press.

3. Brekke E. F., 2004, Complex Behaviour in Dynamical Systems, Norwegian University of Science and Technology, http://www.stud.ntnu.no/~edmundfo/tma4700report.pdf pp 37-38, last visited on 03.02.2006.

4. Corne D., Dorigo M., and Glover F., 1999, New Ideas in Optimisation. ISBN 007 7095065, McGraw-Hill International (UK) Limited.

5. Eberhart R. and Kennedy J., 1995, Particle Swarm Optimisation, Proceedings of the 1995 IEEE International Conference on Neural Networks., vol. 4, 1942-1948. IEEE Press.

6. Eberhart R. and Shi Y., 1998, Comparison between Genetic Algorithms and Particle Swarm Optimisation, The 7-th Annual Conference on Evolutionary Programming, San Diego, USA.

7. Eberhart R. and Shi Y., 2000, Comparing inertia weights and construction factors in Particle Swarm Optimisation. Proceedings of the Congress on Evolutionary Computation, pp 84-89.

8. Eiben A.E., Smith J.E., 2003, Introduction to Evolutionary Computing, Springer-Verlag, ISBN 3-540-40184-9.

9. Fogel D.B., 2000, Evolutionary Computation – Toward a New Philosophy of Machine Intelligence, IEEE Press, ISBN 0-7803-5379-X.

10. Federici D., 2005, Norwegian University of Science and Technology, http://www.idi.ntnu.no/~federici/subai/index1.html last visited on 03.02.2006.

11. Himmelblau D., 1972, Applied Non-linear Programming, McGraw-Hill, New York.

12. Igel C., Toussaint M., 2004, A No-Free-Lunch Theorem for Non-Uniform Distribution of Target Functions, Journal of Mathematical Modelling and Algorithms, 3: 313–322, 2004.© 2004 Kluwer Academic Publishers.

13. Michalewicz, Z. and Fogel, D., 2002, How to Solve It: Modern Heuristics, ISBN 3-540-66061-5 Springer-Verlag, Berlin, Heidelberg, New York.

14. Parmee I.C., 2001, Evolutionary and adaptive computing in engineering design, ISBN 1852330295, Springer -Verlag London Limited.

15. Parsopoulos, K.E., & Vrahatis, M.N., (2001), Particle Swarm Optimizer in Noisy and Continuously Changing Environments, M.H. Hamza (ed.), Artificial Intelligence and Soft Computing, IASTED/ACTA Press (Anaheim, CA, USA), ISBN: 0-88986-283-4, pp. 289-294.

16. Penev, K., and Littlefair, G., 2003, Free Search – a Novel Heuristic Method, Proceedings of PREP 2003, 14-16 April, Exeter, UK, (pp 133-134).

17. Penev K., 2004, Adaptive Computing in Support of Traffic Management, In I.C. Parmee editor, Adaptive Computing in Design and Manufacturing, ISBN 1852338296, ACDM 2004, 20- 22 April, Bristol, UK, pp 295-306.

18. Penev K., and Littlefair G., 2005, Free Search – A Comparative Analysis, Information Sciences Journal, Elsevier, Volume 172, Issues 1-2, pp 173-193.

19. Penev, K., 2005, Adaptive Search Heuristics Applied to Numerical Optimisation, Thesis for a degree of Doctor of Philosophy, The Nottingham Trent University, UK,

20. Price K., and Storn R., 1997, "Differential Evolution", Dr. Dobb's Journal 22 (4), 18-24.

21. Price K., Chisholm K., Lampinen J., Storn R., Zelinka I., 1999, Part Two Differential Evolution, in Editors Corne D.,M.Dorigo, and F.Glover, New Ideas in Optimisation. ISBN 007 7095065, McGraw-Hill Int. (pp 77-158).

22. Rogalsky, T., Derksen, R., & Kocabiyic, S., (1999), Differential Evolution in Aerodynamic Optimisation, Proceedings of the 46-th Annual Conference of the Canadian Aeronautics and Space Institute, (May 2-5). (pp. 29-36).

23. Torn A., Zilinskas A., 1989, Global Optimization, Lecture Notes in Computer Science 350, Berlin, Springer.

24. Wolpert D.H., and Macready W.G., 1997, No Free Lunch Theorems for Optimisation, IEEE Trans. Evolutionary Computation, Vol. 1:1, pp. 67-82.