FREE SEARCH AND DIFFERENTIAL EVOLUTION TOWARDS DIMENSIONS NUMBER CHANGE

KALIN PENEV
Southampton Solent University,
Faculty of Technology,
Southampton, UK

ABSTRACT
This paper presents an exploration of Free Search (FS) and modified Differential Evolution (DE) with enhanced adaptivity. The aim of the study is to identify how these methods can cope with changes of the number of variables of a hard design test, unaided. The results suggest that both methods can adapt successfully to the variation of the number of variables and constraint conditions. The results are presented. Contributions to the engineering design are replacement in high extent of human based search with machine based and movement of optimisation process from human guided to machine self guided search.

INTRODUCTION

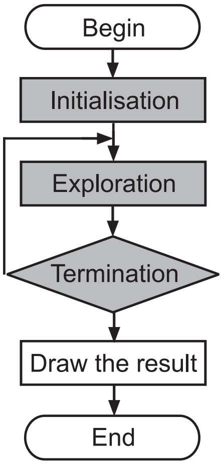
This article presents an evaluation of two population based optimisation methods namely Free Search [Penev and Littlefair, 2003][Penev, 2004A] and Differential Evolution [Storn and Price, 1995]. Objectives of the study are development of a tool for search and optimisation capable to cope with heterogeneous tasks resistant to existing methods. Particular aim of the review is to clarify how FS and DE can cope with change of the number of variables of a hard non-linear constraint optimisation problem, unaided. For a basis of the experiments the so-called bump tests is used [Keane, 1995].

Free Search can be classified as a heuristic method that relies upon trial and error rather than comprehensive theory. It attempts to model a heuristic behaviour of animals in nature and their day-by-day exploration of the surrounding environment in order to find favourable conditions. During this process they learn via trial and error and refine their behaviour accordingly. The FS model negotiates a continuous landscape in discrete steps [Penev, 2006].

Other explored method Differential Evolution can be classified as combinatorial method for fast effective optimisation [Price, 1999].

FREE SEEARC ESSENTIAL PROPERTIES

This section describes the principles for self-regulation applied in the algorithm. A new solution is generated as deviation of a current one $x = x_0 + \Delta x$, where x is a new solution, x_0 is a current solution and Δx is modification strategy. x, x_0 and Δx are vectors of real numbers. The search process begins with initialisation. A determination of the search space boundaries [$Xmin_i$ and $Xmax_i$], population size m, limit for the number of explorations G, limit for the number of steps for exploration T, minimal and maximal values for the frame of a neighbourhood space [*Rmin*, *Rmax*] is required [Penev, 2005][Penev, 2006].




```
Free Search{
      Initialisation{
        initialise Xmaxi, Xmini, n, m, G, x0ji, T, Rji
        take initial walks, ftj (x0ji+Δxt)
        generate an initial pheromone Pk
        distribute the initial pheromone Pk
        learn the initial achievements Pk
      } // end initialisation
     do{ // exploration
        generate sensibility Sj
        select start locations for a walk x'0j = xk(Sj,Pk)
        take exploration walks ftj (x'0j+Δxt)
        generate a pheromone Pk
        distribute the pheromone Pk
        learn the achievements Pk
      } while (f(xj)<fopt && g<G) // termination
      Output the result
} // end Free Search
```

Figure 1. Free Search – algorithm architecture and an example in pseudo code.

In Figure 1: $Xmin_i$ and $Xmax_i$ are search space boundaries. m is population size. $j = 1,..,m$, indicates individuals in the population. $k = 1,..,m$ indicates the location marked with pheromone from each individual after an exploration. n is number of dimensions. $i = 1,..,n$ indicates one dimension. T is step limit per walk. t is current step. $R_{ji} \in [Rmin, Rmax]$ is a variable frame for the neighbouring space for individual j within dimension i. 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.

For multi-start optimisation FS allows variation of the initialisation strategies. Upon initialisation each individual takes an exploratory walk. It 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 used in the algorithm is:

$\Delta x_{tji} = R_{ji} * (Xmax_i - Xmin_i) * random_{tji}(0,1)$, where: $i = l$ for a one-dimensional step (l indicates one dimension); $i = 1,..,n$ for a multi-dimensional step. T is the step limit per walk. t is the current step $t = 1,..,T$.

Δx_{tji} indicates the actual size of the neighbourhood space for a particular problem for step t of individual j within dimension i.

The exploration performs heuristic trials based on stochastic divergence from one location, followed by an individual assessment of the explored locations.

The best location is marked with pheromone. It indicates the quality of the locations and may be considered as result or cognition from previous activities. The assessment, during the exploration, is defined as: $f_{tj} = f(x_{tji})$, $f_j = \max (f_{tj})$.

The value of the objective function achieved from animal j for step t is $f_{tj} = f(x_{tji})$. The quality of the location marked with pheromone from an individual after one exploration is $f_j = \max(f_{tj})$. The pheromone generation is generalised for the whole population: $P_j = f_j / \max(f_j)$, where $\max(f_j)$ is the best achieved value from the population for the exploration.

Then a generation and a refining of the sensibility follow. 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 a new exploratory walk defined as: $x_{0j} = x_k\ (P_k \geq S_j)$, where $j = 1,..,m$, j is the number of the individual; $k = 1,..,m$, , 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 termination. Acceptable criteria for termination are:

- reaching the optimisation criteria: $fmax \geq fopt$, where *fmax* is the maximal achieved solution, *fopt* is an acceptable value of the objective function.

- expiration of the generation limit: $g \geq G$, where G is the limit and g - current value

- complex criterion: $((fmax \geq fopt) \,||\, (g \geq G))$.

The Free Search structure consists of generalised events initialisation, exploration and termination, which reduces the well-known general description of evolutionary algorithms [Eiben and Smith, 2003] [Corne *et all*, 1999].

DIFFERENTIAL EVOLUTION A BRIEF OVERVIEW

Differential Evolution is an approach for optimisation of non-linear and non-differentiable continuous search space [Price and Storn, 1997].

The individuals in DE are called vectors [Price, 1999]. A specific conceptual feature of DE is the implicit assumption that the individuals are not of equal value. Proposed modification strategies in DE are based on the difference between these vectors. The assumption excludes all possible populations with equal value individuals. If all individuals belong to one location of the space, then generated new individuals belong to the same location. Therefore for the experiment the original start condition of the bump problem: start from $x_i = 5$; is changed to the: start from $x_i = 4 + r_i$, $i = 1,...,n$ where r_i is random and $r_i \in (0, 2)$. An exclusion of populations with equal individuals is classified as a negative restriction of the DE. If during the optimisation process all the individuals become on an equal value and differential becomes zero, follows that this value is the optimum. This negative effect can be observed on the optimisation of flat problems [Rosenbrock, 1960]. In fact it stops the optimisation process. FS avoids such restriction [Penev and Littlefair, 2005].

A typical feature of DE is generation of new individuals. DE selects from the current population target, donor and differential vectors. From these vectors DE generates new trial vector, which replaces the target vector, if it is better, within the new population. The authors proposed several strategies for generation of trial vectors [Price and Storn, 1997]. It has been found that the concept for donor vector reflects positively on the search process. The concept for differential vector is, also, very powerful. It positive characteristic is an implicit adjustment of the differential vector component to the range of the modified component. What range is appropriate for the modification differential? DE does not require the answer to this question. It automatically adjusts the modification differential to the range of the modified component. For example, for the bump problem, for n = 20, $0 < x_i < 10$, i =1,…,20. It can be observed that for the optimal value the range of x_1 is about $x_1 = 3$, and the range of the x_{20} is about $x_{20} = 0.4$. An appropriate differential for x_1 is $\Delta x_1 \in (0, 1)$ and for x_{20} is $\Delta x_{20} \in (0, 0.1)$. Values for $\Delta x_{20} \in (0.1, 1)$ lead to non-optimal results. It can be observed that Differential Evolution can adjust adaptively an appropriate range of the differential vector [Lampinen and Zelinka, 1999].

A modified DE with enhanced adaptivity is implemented and explored with originally proposed strategies [Price, 1999][Lopez-Cruz *et all*, 2001] [Feoktistov and Janaqi, 2004]. Mutation factor varies from 0.5 to 1.5 with step 0.1. Crossover probability is 0.5. Population is 10 individuals for all experiments.

TEST PROBLEM

As a basis for the tests the so-called bump problem [Keane, 1996] is used. The problem is maximise:

$$\mathrm{f}(x_i) = \left|\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)\right| / \sqrt{\sum_{i=1}^{n} i x_i^2}$$

for: $0 < x_i < 10$ and i = 1, ..., n, subject to: $\prod_{i=1}^{n} x_i > 0.75$, $\sum_{i=1}^{n} x_i < 15n/2$,

starting from $x_i = 4 + r_i$, $i = 1,..., n$, where r_i is random and $r_i \in (0, 2)$.

The number of dimension n varies form 2 to 20 with step 1. The criterion for termination is expiration of number of iterations g for each space.

EXPERIMENTAL RESULTS

	g= 100		g=2000		g=20000	
n	FS	DE	FS	DE	FS	DE
2	0.357733	0.353180	0.364830	0.364936	0.364979	0.364975
3	0.509981	0.515773	0.515767	0.515785	0.515783	0.515785
4	0.614223	0.507847	0.622250	0.621809	0.622250	0.622280
5	0.524064	0.631584	0.632544	0.634448	0.634076	0.634448
6	0.599470	0.596194	0.692749	0.683830	0.693474	0.693847
7	0.561774	0.647054	0.697660	0.694243	0.703647	0.704614
8	0.569637	0.688547	0.718222	0.727581	0.727139	0.727623
9	0.641508	0.508571	0.710156	0.741210	0.738965	0.741230
10	0.578157	0.544831	0.735708	0.731991	0.743541	0.747297
11	0.593189	0.497803	0.751095	0.758743	0.758576	0.760296
12	0.543493	0.514308	0.756067	0.758824	0.758970	0.762519
13	0.565841	0.606033	0.748267	0.760477	0.766523	0.769229
14	0.510785	0.445228	0.753575	0.767970	0.774202	0.774161
15	0.535867	0.500196	0.769648	0.782026	0.777287	0.782395
16	0.478487	0.512809	0.751070	0.772553	0.786176	0.773667
17	0.464643	0.364923	0.723047	0.790685	0.788844	0.791031
18	0.480829	0.453693	0.780681	0.784572	0.794709	0.781361
19	0.507314	0.415786	0.764852	0.718407	0.795602	0.779572
20	0.448555	0.571672	0.760953	0.753888	0.800154	0.785628

Table 1. The best achieved results.

With both algorithms three series of experiments limited to 100, 2000 and 20000 iterations, and 10 experiments per each space, are made. FS and DE have population of 10 individuals. The best achieved values of the objective function per space and per method are presented in Table 1. *g* denotes the iterations limit and *n* is the number of dimension. The FS results are placed within the columns indicated with FS, and DE results within the columns indicated with DE.

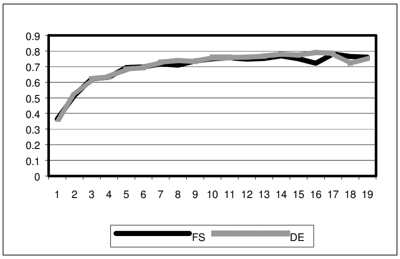


Figure 2: Best results 2000 iterations.

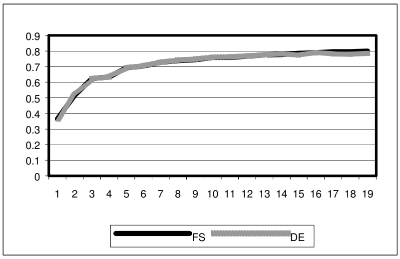


Figure 3: Best results 20 000 iterations.

The results reached within 20000 iterations correspond to the published in the literature [Keane, 1996][Jiandong *et all*, 2001][Michalewicz and Fogel, 2002][Penev, 2004], and can be acsepted as optimal with certain precision.

FS is explored additionaly for high number of iterations. The result for 20 dimensions is *Fmax20* = 0.803619104125586458664542988. The constraint value is *p* = 0.75000000000000111022302463. The same maximal value of the function for twenty dimensions is achieved for several other locations. Other authors publish similar results up to the eight digit after decimal point [Jiandong *et all*, 2001]. *Fmax50*=0.83526234835811175 is achieved for 50 dimensions. The constraint is *p*=0.75000000000000122. This result exceeds the results published by other authors [Keane, 1996] [Michalewicz and Fogel, 2002] [Jiandong *et all*, 2001]. A recent achievement of Free Search for 200 dimensions is: *Fmax200* = 0.8501375. The constraint is p = 0.750000006378. It requires reconsideration of the assumptions and results published by other authors [Jiandong *et all*, 2001].

CONCLUSIONS

In summary FS and DE demonstrate good capability to cope with this hard constrained test. They both can adapt to the changes of the number of variables and constraints conditions, unaided. The experimental results suggest that adaptive methods can solve complex problem with standard operators and common general configuration of the optimisation parameters.

A speed with which both methods reach optimal solution with acceptable level of precision could contribute to the acceleration of design process of complex engineering tasks and deserves attention.

This study confirms that Free Search replaces in high extent human-based search with machine-based and moves the optimisation process from human guided to machine self guided search. A contribution to engineering design is a powerful tool for search and optimisation, which moves up level of abstraction of human activities to the definition and description of the design problem, rather than performing and guiding search process. In that sense FS can advance a wide range of disciplines in the efforts to cope with complex optimisation and search tasks.

Further investigations can focus on evaluation with dynamic and time dependent search space. A pragmatic area for further research is application to real-world design and engineering tasks.

REFERENCES

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

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

Feoktistov V., and Janaqi S., 2004, New Strategies in Differential Evolution – Design Principle, *in I.C. Parmee editor, Adaptive computing in design and manufacture VI*, ISBN 1-85233-829-6 Springer - Verlag London Limited. pages (335-346).

Jiandong Li, Zhuo Kang, Yan Li, Hongqing Cao, Pu Liu, 2001, Automatic Data Mining by Asynchronous Parallel Evolutionary Algorithms, *Tools,* p. 0099, 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS39).

Keane A. J., 1995, Genetic algorithm optimization of multi-peak problems: studies in convergence and robustness, *Artificial Intelligence in Engineering* 9(2) pages (75-83).

Keane, A.J., 1996, A brief Comparison of Some Evolutionary Optimization Methods, *in Modern Heuristic Search Methods, ed. V. Rayaward-Smith, I. Osman, C. Reeves and G. D. Smith, J Wiley*, ISBN 0-471 96280 5, (pp 255-272).

Lampinen J. and Zelinka I., 1999, Mechanical Engineering Design Optimisation by Differential Evolution, *Chapter Eight In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimisation.* ISBN 007 7095065, McGraw-Hill International (UK) Limited, pages (127- 146).

Lopez-Cruz I.L., Van Willigenburg L.G., and Van Straten G., 2001, Parameter strategy in Differential Evolution algorithm for optimal control, *in Proceedings of the IASTED International Conference Artificial Intelligence and Soft Computing*, May 21-24, Cancun, Mexico, pages (211-216).

Michalewicz, Z., & Schoenauer, M., 1996, Evolutionary Algorithms for Constrained Parameter Optimisation Problems. *Evolutionary Computation*, Vol. 4, No. 1, (pp 1-32).

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

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

Penev K., 2004, Adaptive Computing in Support of Traffic Management, In Parmee I.C. editor, *Adaptive Computing in Design and Manufacturing VI*, ISBN 1852338296, pp 295-306.

Penev, K., 2004A, Adaptive Search Heuristics Applied to Numerical Optimisation, PhD thesis submitted in partial fulfilment of the requirements of SI and The NTU, UK.

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

Penev, K., 2006, Free Search - Extensive Evaluation, In Parmee I.C. editor, *Adaptive Computing in Design and Manufacturing VII*, ISBN 978-0-9552885-0-0, pp 157-165.

Price, K., & Storn, R., 1997, Differential Evolution- A simple evolution strategy for fast optimisation, *Dr. Dobb's Journal 22 (4), (April)* (pp. 18-24).

Price, K., 1999, An Introduction in Differential Evolution, *Chapter Six in D. Corne, et al., New Ideas in Optimisation,* ISBN 007 7095065, McGraw-Hill International (UK) Ltd., (pp. 77-108).

Rosenbrock, H.H., 1960, An automate method for finding the greatest or least value of a function. *Comput. J. 3 (1960),* (pp.175-184).

Storn R. and Price K., 1995, Differential Evolution – A simple and efficient adaptive scheme for global optimisation over continuous spaces, TR-95-012, International Computer Science Institute, 1947 Center Street, Berkeley, CA 94704-1198, Suite 600.