PageRank Method for Benchmarking Computational Problems and their Solvers

Joseph Gogodze

**Institute of Control System, Georgian Technical University,
Tbilisi, 0160, Georgia**

Abstract

In this short note, we propose a new tool for benchmarking computational problems and their solvers. The proposed tool, which is a version of the PageRank method, is illustrated using an example to demonstrate its viability and suitability for applications.

Keywords: *Benchmarking, Software, Solvers, PageRank.*

1. Introduction

In recent years, intensive studies have been conducted to evaluate the effectiveness of various solvers and various methods for this purpose have been proposed in the literature [1-9]. As noted in [9], most benchmarking tests utilize evaluation tables displaying the performance of each solver for each problem under a specific evaluation metric (e.g., CPU time, number of function evaluations, or number of iterations). Different methods (based on suitable "statistical" quantities) are used to interpret data from these tables, including the mean, median, and quartiles [1, 4, 5], ranking [4, 5, 7, 8], cumulative distribution function [9], etc. The selection of a benchmarking method currently depends on the subjective tastes and individual preferences of researchers, who perform evaluations using solvers/problems sets and evaluation metrics. The advantages and disadvantages of each proposed method are often a source of disagreement; however, this only stimulates further investigation in the field.

The method discussed in this paper was proposed to introduce a new benchmark that directly accounts for the natural relationship between problems and solvers, which is determined by their evaluation tables. Namely, this paper introduces the benchmarking context concept as a triple $< S, P, J >$, where S is a set of solvers, P is a set of problems, and $J : S \times P \to \mathbb{R}$ is an assessment function (a performance or evaluation metric). This concept is quite general and, furthermore, emphasizes that problem and solver benchmarking cannot be considered separately. Based on the data presented by the benchmarking context $< S, P, J >$, a special procedure was defined allowing solvers and problems to be ranked. It should also be noted that the proposed procedure is a specific version (most probably the simplest) of the Google PageRank method [10]. Various versions of PageRank have been successfully applied to numerous fields: economics [11], bibliometrics [12], and others [13]. Motivated by these applications, this study aimed to propose a PageRank procedure as an effective tool for benchmarking computational problems and their solvers.

The remainder of this paper is organized as follows: section 2 describes the proposed methodology for evaluating and comparing solver qualities and problem difficulties; section 3 considers the applications of the proposed tool in a selected benchmarking problem; and finally, section 4 contains a conclusion.

2. Method

Consider a set P of problems and a set S of solvers under the assumption that a function $J : S \times P \to \mathbb{R}$, henceforth referred to as the assessment function (performance metric), is given. Further, assume, for definiteness, that the high and low values of J correspond to the "worst" and "best" cases, respectively, and for convenience interpret $J(s, p)$ as the cost of solving the problem $p \in P$ with solver $s \in S$. Note that if $J(s, p) < J(s', p')$, it can be said that $s \in S$ solves $p \in P$ better than solver $s' \in S$ solves problem $p' \in P$ (i.e., the problem $p \in P$ was easier for solver $s \in S$ than the problem $p' \in P$ was for solver $s' \in S$). For a given $< S, P, J >$, further assume that the following assumptions hold (n_P, n_S below are given natural numbers):.

(A0) $P = \{1, \ldots, n_P\}$ and $S = \{1, \ldots, n_S\}$;

(A1) $J(s,p) \geq 0 \quad \forall (s,p) \in S \times P;$

(A2) $$\begin{cases} I_P(s) = \sum_{p \in P} J(s,p) > 0 \ \forall s \in S; \\ I_S(p) = \sum_{s \in S} J(s,p) > 0 \ \forall p \in P. \end{cases}$$

Assumption (A0) establishes that the sets P, S are finite. Assumption (A1) is not restrictive because the sets P, S are finite. Assumption (A2) can be interpreted as a "no triviality" condition of the assessments (such that, as a requirement, each solver and each problem should be tested with at least one problem and one solver, respectively). Obviously, (A2) implies that $I_{SP} = \sum_{s \in S, p \in P} J(s,p) > 0.$ The triple $< P, S, J >$, which satisfied assumptions (A0), (A1), and (A2), is henceforth referred to as the benchmarking context.

For a given benchmarking context $< P, S, J >$, numerous new quantities can be defined. For any $p \in P$, $s \in S$:

$$\iota_P(s) = I_P(s) / I_{SP}, \quad \iota_S(p) = I_S(p) / I_{SP}.$$

Of further note, indicators I_P and I_S can be considered as the characteristics of solver "efficiency" (relative to P) and problem "difficulties" (relative to S), respectively. For example, if $I_P(s) < I_P(s')$, it can be said that the solver $s' \in S$ is ($P-$) worse than solver $s \in S$ for a specified criterion, such as processing time. Conversely, if $I_S(p) < I_S(p')$, problem $p' \in P$ is (S) more difficult than problem $p \in P$ (e.g. because solvers from S require more processing time to solve problem $p' \in P$ than problem $p \in P$).

Analogously, the "averaged" indicators ι_P and ι_S can be considered as the characteristics of solver "efficiency" (relative to P) and problem "difficulties" (relative to S), respectively. For example, if $\iota_S(p) < \iota_S(p')$ problem $p' \in P$ is said to be (S) more difficult than problem $p \in P$ (because solvers from S require, on average, more processing time to solve problem $p' \in P$ than problem $p \in P$).

It is possible to calculate other quantities that can be used for benchmarking with the help of the values defined above. For example, it is possible to introduce various statistics related to the vectors $I_P(\cdot), I_S(\cdot)$ into consideration, such as mean, median, and rank. As was noted in the introduction, such "statistical" quantities are frequently encountered in benchmarking research. Furthermore, indicators $\iota_P(\cdot), \iota_S(\cdot)$ can be viewed as probability measures on the sets P, S respectively, as it is evident that:

$$\begin{cases} 0 \leq \iota_P(s) \leq 1, \sum_{s \in S} \iota_P(s) = 1; \\ 0 \leq \iota_S(p) \leq 1, \sum_{p \in P} \iota_S(p) = 1; \end{cases}$$

Obviously, the corresponding cumulative distribution functions may also be used for benchmarking purposes.

Further, note that the assessment function $J : S \times P \to \mathbb{R}$ may be composed of other indicators, and can itself form new assessment functions. An important example of constructing additional assessment functions is: for a given $< P, S, J >$, consider $J_{RCA} : S \times P \to \mathbb{R}$, to be defined as

$$J_{RCA}(s,p) = \frac{J(s,p) \Big/ \sum_{p' \in P} J(s,p')}{\sum_{s \in S} J(s,p) \Big/ \sum_{s \in S, p \in P} J(s,p)} = \frac{J(s,p) / I_P(s)}{I_S(p) / I_{SP}}.$$

Here, $J_{RCA} : S \times P \to \mathbb{R}$ can be considered a new assessment function. Note also that $J_{RCA}(s,p) < 1$ implies the cost share of solving problem p with solver s (from the total cost of solving problems from P with s) is less than the total cost share of solving p using solvers from S in the total cost of solving problems from P with solvers from S. The inequality $J_{RCA}(s,p) < 1$ is interpreted as the "revealed comparative advantage (RSA)"[1] of solver s in problem p (in other words, s is said to be the significant solver of p. For comparison note also, see [11] p. 10571, that "a country can be considered to be a significant exporter of product p if its Revealed Comparative Advantage (the share of product p in the

[1] Note that the term revealed comparative advantage (RSA) was introduced in (Balassa, B., Trade Liberalization and Revealed Comparative Advantage. The Manchester School of Economic and Social Studies, 3(2), 1965, pp. 99-123) and is widely used for comparative analyses in international trade. This analogy can be clarified by assuming that $J(s,p)$ represents product export p from country s.

export basket of product p in world trade) is greater than 1.") and the following quantities are introduced:

$$M_J(s,p)=\begin{cases}1, & J_{RCA}(s,p)<1\\ 0, & J_{RCA}(s,p)\ge 1\end{cases},$$

$$u_J(s)=\sum_{p\in P} M_J(s,p), \quad a_J(p)=\sum_{s\in S} M_J(s,p).$$

The quantity $u_J(s)$ (henceforth called the universality of solver $s\in S$) represents the number of problems for which solver $s\in S$ is significant. Analogously, $a_J(p)$ (henceforth called the accessibility of problem $p\in P$) represents the number of solvers that are significant to the problem $p\in P$. Note now that for the given benchmarking context $<P,S,J>$, the matrixes

$$W_P=[w_{sp}^P],\, w_{sp}^P=M_J(s,p)/u_J(s) \quad \forall(s,p)\in S\times P,$$
$$W_S=[w_{ps}^S],\, w_{ps}^S=M_J(s,p)/a_J(s) \quad \forall(p,s)\in P\times S,$$

can be introduced and it is easy to verify that

$$W_P\ge 0, W_P 1_P=1_S; \quad W_S\ge 0, W_S 1_S=1_P$$

where

$$1_P=\{\underbrace{1,\ldots,1}_{n_P}\}, 1_S=\{\underbrace{1,\ldots,1}_{n_S}\}.$$

Now we can assume that the vectors $e=e(\cdot)\in\mathbb{R}^{n_S}, d=d(\cdot)\in\mathbb{R}^{n_P}$ defined by equalities

$$\begin{cases} e(s)=\kappa_e\sum_{p\in P} w_{sp}^P d(p)\\ d(p)=\kappa_d\sum_{p\in S} w_{ps}^S e(s)\end{cases},$$

where $\kappa_e,\kappa_d>0$ represent some scaling coefficients, estimate the effectiveness of solvers (henceforth called the S-score) and difficulties of problems (henceforth called the P-score), respectively. We note that these equations reflect a simple idea: the effectiveness of solvers is directly proportional to the weighted sum of problem difficulties and the difficulties of problems is directly proportional to the weighted sum of solver effectiveness. Using vector notations, we thus have $e=\kappa_e W_P d, \quad d=\kappa_d W_S e$ and consequently $W_{PS}e=\lambda e, \quad W_{SP}d=\lambda d,$ where

$$\lambda=(\kappa_e\kappa_d)^{-1}, W_{PS}=W_P W_S, W_{SP}=W_S W_P.$$

This means that λ is an eigenvalue and e,d are the corresponding eigenvectors. Following [11], we select the eigenvectors corresponding to the second largest eigenvalue and standardize them using the Z-score.

The described benchmarking method is quite general and can be used to compare solvers and problems in various areas. It should also be noted that, as was mentioned in the introduction, the proposed method can be considered as the simplest version of the Google PageRank method, and, of course, many variations thereof are possible. For the sake of comparison note that after the seminal publication of [11] approximately seven hundred different measures for defining and benchmarking economic complexities have been proposed [14].

3. Case Study: Benchmarking of Differential Evolution Algorithms

Recently, researchers [15] have conducted a performance analysis of differential evolution (DE) algorithms using a well-known set of test functions. In this section, we use these results to illustrate the proposed benchmarking method.

3.1 Data

The previous study [15] considered the nine optimization algorithms listed in Annex Table A1 and 25 test functions listed in Annex Table A2. The sources cited in these tables present detailed information on the selected algorithms and test functions. Utilizing these algorithms and test functions, the sets of 9 solvers and 50 problems were defined (see Annex Tables A3 and A4).

A description of the assessment function used in [15] follows. First, note that the expected running time (ERT), a widely used performance metric for optimization algorithms [16], is defined as

$$ERT(\tau)=mean(M_\tau)+\frac{1-q}{q}N_{\max}, \quad q=\frac{N_{succes}}{N_{total}},$$

where τ is a reference threshold value, M_τ is the number of function evaluations required to reach an objective value better than τ (such as successful runs), $N_{\max}$ is the maximum number of function evaluations per optimization

IJCSI International Journal of Computer Science Issues, Volume 15, Issue 3, May 2018
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.5281/zenodo.1292395

run, N_{succes} is the number of successful runs, N_{total} is the total number of runs, and q is the named success rate. Note now that in order to compare qualitative performances using ERT, it is necessary that all compared algorithms meet the success criterion at least a few times. Accordingly, the special quantity — the random sampling equivalent–expected run time (ERT_{RSE}) — may be introduced as a performance metric. To clarify the meaning of this quantity, we note that, for example, ERT_{RSE}=300 "means that the corresponding algorithm requires 300 function evaluations to obtain a function evaluation better than the threshold, (which was defined as the expected best objective value for 1000 uniform random samples in the problem domain)" (see [15], p.8). Annex Table A5 presents the ERT_{RSE} values for all problem–solver pairs used to define an assessment function $J : S \times P \to \mathbb{R}$. Obviously, assumptions (A0), (A1), and (A2) hold and, hence, the benchmarking context $< P, S, J >$ is fully determined for the case under consideration.

3.2. Results

All necessary calculations were conducted using R software within the RStudio framework. Results of the S- and P-score calculations for the considered case are represented in Tables 1 and 2, respectively. Fig. 1 presents scatter plots reflecting the universality vs. S-score and accessibility vs. P-score, and shows that these dependencies can be considered as monotonically increasing.

Table 1 S-score

Solver	Score	Solver	Score
S03	-0,7522	S09	-0,4403
S05	-0,7330	S08	0,5611
S04	-0,7010	S02	1,5966
S06	-0,6340	S01	1,6324
S07	-0,5297	-	-

Table 2 R-score

Problem	Score	Problem	Score
P14	-2,0285	P30	0,5271
P15	-2,0285	P04	0,5271
P08	-1,5207	P29	0,5345
P11	-1,5207	P10	0,5953
P12	-1,5207	P21	0,5953
P18	-1,5207	P42	0,5953
P19	-1,5207	P01	0,5953
P20	-1,5207	P03	0,5953
P33	-1,5207	P13	0,6018
P36	-1,5207	P26	0,6018
P37	-1,5207	P44	0,6018
P40	-1,5207	P45	0,6018
P39	-0,8346	P46	0,6018
P05	-0,5214	P24	0,7599
P25	-0,5102	P49	0,7599
P50	-0,5102	P48	0,9322
P16	-0,1358	P38	0,9601
P28	0,1555	P06	0,9739
P02	0,38252	P09	0,9739
P07	0,40813	P23	0,9738
P17	0,40813	P31	0,9739
P22	0,40813	P34	0,9739
P32	0,40813	P35	0,9739
P47	0,40813	P41	0,9739
P27	0,41967	P43	0,9739

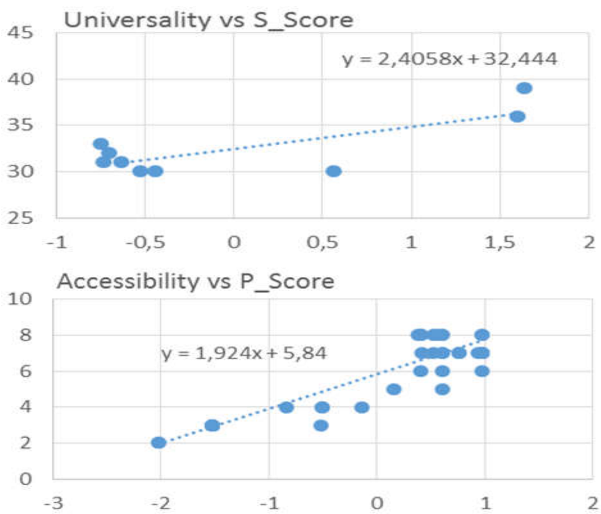

Fig.1 Universality vs S-score and Accessibility vs P-score

Our calculations show that the solver S01 can be considered as the "best" in the framework of the given benchmarking context. In addition, the 16 problems with negative P-scores (see Table 2) can be considered as the most inaccessible (difficult/complex) in the framework of the given benchmarking context.

4. Conclusions

This short note introduced a new method of benchmarking computational problems and their solvers. The proposed method is quite general and can be viewed as the simplest version of the PageRank method. Of course, other versions of the PageRank method can also be used for benchmarking purposes. Furthermore, we considered an illustrative example to demonstrate the viability and suitability of the proposed method for applications.

Appendix

Table A1. Algorithms (source [15])

Code	Short Description
DE	"Rand/1/bin" Differential Evolution [18]
DE2	"Best/2/bin" Differential Evolution [19]
jDE	Self-adapting Differential Evolution [20]
JADE	Adaptive Differential Evolution [21]
SaDE	Strategy adaptation Differential Evolution [22]
Code	Composite vector strategy Differential Evolution [23]
epsDE	Ensemble parameters Differential Evolution [24]
SQG	Stochastic Quasi-Gradient search [25]
SQG-DE	Stochastic Quasi-Gradient Differential Evolution [15]

Table A2. Test Functions (source [15])

Code	Short Description (see [17])
F01	Shifted Sphere Function
F02	Shifted Schwefel's Problem 1.2
F03	Shifted Rotated High Conditioned Elliptic Function
F04	Shifted Schwefel's Problem 1.2 with noise in fitness function
F05	Schwefel's Problem 2.6 with the global optimum on the bounds
F06	Shifted Rosenbrock's Function
F07	Shifted Rotated Griewank's Function
F08	Shifted Rotated Ackley's Function with the global optimum on the bounds
F09	Shifted Rastrigin's Function
F10	Shifted Rotated Rastrigin's Function
F11	Shifted Rotated Weierstrass Function
F12	Schwefel's Problem 2.13
F13	Expanded Extended Griewank's plus Rosenbrock's Function
F14	Shifted Rotated Expanded Scaffer's F6
F15	Hybrid Composition Function
F16	Rotated Hybrid Composition Function
F17	Rotated Hybrid Composition Function
F18	Rotated Hybrid Composition Function
F19	Rotated Hybrid Composition Functions with noise in fitness function
F20	Rotated Hybrid Composition Function with a narrow basin for the global optimum
F21	Rotated Hybrid Composition Function
F22	Rotated Hybrid Composition Function with a high condition number matrix
F23	Non-Continuous Rotated Hybrid Composition Function
F24	Rotated Hybrid Composition Function
F25	Rotated Hybrid Composition Function

Table A3. Case Stady: Solvers

Solver	Algorithm	Solver	Algorithm
S01	DE	S06	Code
S02	DE2	S07	epsDE
S03	jDE	S08	SQG
S04	JADE	S09	SQG-DE
S05	SADE	-	-

Table A3.Case Study: Problems

Problem	Description		Problem	Description	
	Dimension	Function		Dimension	Function
P01	30	F01	P26	50	F01
P02	30	F02	P27	50	F02
P03	30	F03	P28	50	F03
P04	30	F04	P29	50	F04
P05	30	F05	P30	50	F05
P06	30	F06	P31	50	F06
P07	30	F07	P32	50	F07
P08	30	F08	P33	50	F08
P09	30	F09	P34	50	F09
P10	30	F10	P35	50	F10
P11	30	F11	P36	50	F11
P12	30	F12	P37	50	F12
P13	30	F13	P38	50	F13
P14	30	F14	P39	50	F14
P15	30	F15	P40	50	F15
P16	30	F16	P41	50	F16
P17	30	F17	P42	50	F17
P18	30	F18	P43	50	F18
P19	30	F19	P44	50	F19
P20	30	F20	P45	50	F20
P21	30	F21	P46	50	F21
P22	30	F22	P47	50	F22
P23	30	F23	P48	50	F23
P24	30	F24	P49	50	F24
P25	30	F25	P50	50	F25

Table A5. Case Study: ERT_{RSE} Metric (source [15])

Problems	Solvers			
	S01	S02	S03	S04
P01	11657	24049	491	264
P02	7401	11567	739	309
P03	6536	32763	606	356
P04	6182	13326	570	285
P05	4342	4766	536	311
P06	7404	11728	491	234
P07	256	124	482	310
P08	2877	3011	1783	2084
P09	15735	24058	467	269
P10	11658	24054	474	256
P11	2414	1555	1967	1502
P12	1072	934	509	342
P13	15709	10150	524	224
P14	7497	19174	2735	1735
P15	437	373	668	366
P16	2999	11671	490	335
P17	6925	11670	514	362
P18	1017	1036	501	314
P19	1271	1045	533	317
P20	1190	1098	491	315
P21	9466	24251	498	253
P22	2433	2889	489	281
P23	13650	24194	474	254
P24	7785	4449	544	272
P25	220	115	445	304
P26	19030	9054	439	205
P27	9055	8145	572	316
P28	3819	10317	514	289
P29	19059	10175	538	341
P30	9173	19040	568	294
P31	11557	13327	491	235
P32	394	187	443	289
P33	1732	2347	2098	1792
P34	15719	19060	482	245
P35	24071	24059	456	235
P36	2024	2568	2187	1324
P37	1183	1263	560	340
P38	7381	9057	514	227
P39	7476	15824	1792	1310
P40	341	291	557	354
P41	7506	11568	495	320
P42	11585	32369	640	327
P43	10264	19135	631	319
P44	24351	7387	511	290
P45	19307	13407	533	321
P46	13571	9089	486	243
P47	3311	5068	564	291
P48	7468	7544	483	248
P49	15709	13335	498	231
P50	304	163	488	315

Table A5. (cont.)

Problems	Solvers				
	S05	S06	S077	S08	S09
P01	329	904	606	202	168
P02	379	894	3653	137	318
P03	519	1148	1045	135	210
P04	342	662	1712	694	310
P05	527	871	564	459	160
P06	315	807	603	124	168
P07	596	879	169	32365	107
P08	2673	2720	2082	132	2771
P09	380	818	726	126	193
P10	320	850	630	201	172
P11	1321	1541	1979	368	1949
P12	416	931	594	128	225
P13	231	723	898	161	289
P14	1509	2366	6379	6899	1658
P15	565	857	458	1016	185
P16	615	689	616	219	179
P17	572	845	693	9096	178
P18	460	741	341	263	143
P19	551	836	389	259	148
P20	545	1039	384	284	154
P21	327	901	597	301	175
P22	389	730	602	13320	202
P23	325	802	586	664	181
P24	376	911	573	10183	187
P25	554	855	167	5321	109
P26	254	674	579	192	178
P27	388	908	2135	127	296
P28	455	764	566	121	145
P29	434	920	2931	1332	310
P30	366	964	1418	451	214
P31	244	831	609	132	182
P32	536	884	231	24041	109
P33	2724	2522	2740	117	1960
P34	295	845	738	123	194
P35	306	739	661	169	188
P36	1814	1825	1675	347	1331
P37	418	1132	774	119	224
P38	193	604	1139	145	336
P39	1198	1296	3482	4706	1808
P40	556	852	331	205	173
P41	487	738	593	260	156
P42	515	794	809	5743	192
P43	380	1071	892	352	226
P44	389	866	728	307	210
P45	372	915	794	380	227
P46	320	871	605	325	176
P47	431	755	724	19048	153
P48	333	880	590	537	170
P49	245	784	688	8170	201
P50	552	1038	210	13368	109

References

[1] Benson, H.Y., Shanno, D.F., Vanderbei, R.J.: Technical Report ORFE-00-02, Princeton University, Princeton, New Jersey, 2000.

[2] Billups, S.C., Dirkse, S.P., Ferris, M.C.: A comparison of algorithms for large-scale mixed complementarity problems. Comput. Optim. Appl., 7 , 1997, pp. 3–25.

[3] Bondarenko, A.S., Bortz, D.M., More, J.J.: COPS: Large-scale nonlinearly constrained optimization problems. Technical Memorandum ANL/MCS-TM-237, Argonne National Laboratory, Argonne, Illinois, 1998 (Revised October 1999).

[4] Bongartz, I., Conn, A.R., Gould, N.I.M., Saunders, M.A., Toint, P.L.: A numerical comparison between the LANCELOT and MINOS packages for large-scale numerical optimization. Report 97/13, Namur University, 1997.

[5] Conn, A.R., Gould, N.I.M., Toint, P.L.: Numerical experiments with the LANCELOT package (Release A) for large-scale nonlinear optimization. Math. Program. 73,1996,PP.73-110.

[6] Mittelmann, H.: Benchmarking interior point LP/QP solvers, Optim. Methods Softw. 12(1999),pp.655-670.

[7] Nash, S.G., Nocedal, J.: A numerical study of the limited memory BFGS method and the truncated Newton method for large scale optimization. SIAM J. Optim., 1 (1991), pp.358-372.

[8] Vanderbei, R.J., Shanno, D.F.: An interior-point algorithm for nonconvex nonlinear programming. Comput. Optim. Appl., 13, 1999, pp. 231-252.

[9] Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical programming 91, 2000, pp. 201-213

[[10] Langville, A.N., Meyer, C.D.: Deeper inside pagerank. Internet Mathematics, 1(3), 2004, pp. 335-380.

[11] Hidalgo, C.A., Hausmann, R.: The building blocks of economic complexity. Proceedings of the National Academy of Sciences, 106, 2009, pp. 10570-10575.

[12] Bollen, J., Rodriquez, M.A., Van de Sompel, H.: Journal status. Scientometrics, 69(3), 2006, pp. 669-687

[13] Gleich, D. F.: PageRank beyond the Web. SIAM Review, 57(3), 2015, pp. 321-363.

[14] Saleh, A., Kaltenberg, M., Alsaleh, M., Hidalgo, C.A.: 729 new measures of economic complexity (Addendum to Improving the Economic Complexity Index). arXiv preprint arXiv: 1708.04107,2017.

[15] Sala, R., Baldanzini N., Pierini M.: SQG-Differential Evolution for difficult optimization problems under a tight function evaluation budget. arXiv, 2017, preprint arXiv: 1710.06770.

[16] Auger, A., Hansen, N.: Performance evaluation of an advanced local search evolutionary algorithm. In Proceedings of the IEEE Congress on Evolutionary Computation,Vol. 2, IEEE, 2005, pp. 1777-1784.

[17] Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC, 2005, special session on real-parameter optimization. KanGAL report, 2005005.

[18] Storn, R., Price, K.: Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, ICSI, 1995.

[19] Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J. of global optimization, 11(4), 1997,pp. 341-359 .

[20] Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE transactions on evolutionary computation, 10(6), 2006, pp.646-657.

[21] Zhang, J., Sanderson, A. C.: JADE: adaptive differential evolution with optional external archive. IEEE Transactions on evolutionary computation, 13(5), 2009, pp. 945-958.

[22] Qin, A. K., Huang, V. L., Suganthan, P. N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE transactions on Evolutionary Computation, 13(2), 2009, pp.398-417.

[23] Wang, Y., Cai, Z., Zhang, Q.: Differential evolution with composite trial vector generation strategies and control parameters. IEEE Transactions on Evolutionary Computation, 15(1), 2011, pp. 55-66.

[24] Mallipeddi, R., Suganthan, P. N., Pan, Q. K., Tasgetiren, M. F.: Differential evolution algorithm with ensemble of parameters and mutation strategies. Applied Soft Computing, 11(2), 2011, pp. 1679-1696.

[25] Ermoliev, Y. M.: Methods of solution of nonlinear extremal problems. Cybernetics, 2(4), 1966, pp. 1-14.

.

Joseph Gogodze received his PhD degree from V.Steklov Mathematical Institute, Moscow, in 1980. At present, he is working as a Senior Researcher at the Institute of Control System of Georgian Technical University, Georgia, and a Chief Researcher at Institute TECHINFORMI of the same university. For a number of years, he is participated in theoretical and practical research projects in Georgia. Areas of his research interest are optimization theory, calculus of variation, operation research and mathematical modelling.