An Empirical Study on the Parametrization of Cartesian Genetic Programming

Paul Kaufmann
Department of Computer Science
Paderborn University, Germany

Roman Kalkreuth
Department of Computer Science
University of Dortmund, Germany

Benchmarks	Functional set
$(i, i, 1)$-add, (i, i)-mul	$a \wedge b$, $a \wedge \bar{b}$, $\bar{a} \wedge b$, $a \oplus b$, $a\|b$
even parity	$a \wedge b$, $a \wedge \bar{b}$, $\bar{a} \wedge b$, $a\|b$, $a\|\bar{b}$, $\bar{a}\|b$
Koza	$+$, $-$, $*$, $/$, sin, cos, $\ln(\|n\|)$, e^n
Keijzer [3, 6]	$+$, $*$, n^{-1}, $-n$, $\sqrt{n}$

Table 1: Functional set of all benchmarks.

ABSTRACT

Since its introduction two decades ago, the way researchers parameterized and optimized Cartesian Genetic Programming (CGP) remained almost unchanged. In this work we investigate non-standard parameterizations and optimization algorithms for CGP. We show that the conventional way of using CGP, i.e. configuring it as a single line optimized by an (1+4) Evolutionary Strategies-style search scheme, is a very good choice but that rectangular CGP geometries and more elaborate metaheuristics, such as Simulated Annealing, can lead to faster convergence rates.

ACM Reference format:

Paul Kaufmann and Roman Kalkreuth. 2017. An Empirical Study on the Parametrization of Cartesian Genetic Programming. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3075980

1 METHODOLOGY

The basic methodology of our paper is that we first define a set of goal functions $f_1, f_2 \ldots$ and a set of optimization algorithms $a_1, a_2 \ldots$. Then we execute for each tuple (f_i, a_j) the automatic parameter-tuning tool iRace to identify best-performing CGP configurations [1, 2, 4, 5]. The resulting configurations are evaluated at the end in separate experiments to derive their convergence behaviors and for comparison.

2 RESULTS

The results of our experiments are presented in Tab. 2. For better interpretability we also use the Computational Effort (CE) metric at $z = 99\%$.

Following recommendations can be drawn from our experiments.

- For simple Boolean functions (1+1) Hill Climbing (HC) applied on CGP with 30 to 50 rows and 100 to 200 columns performs best.
- For complex Boolean functions Simulated Annealing (SA) applied on CGP with 3 to 10 rows and 30 to 300 columns performs best. Increasing the number of rows to 100 might help in case of heavy functions, such as the multiplication.
- For Boolean functions the best observed mutation rate interval is $[0.1, 1.6]\%$.
- For continuous functions CGP with 3 to 20 rows and 80 to 200 columns performs best.
- For continuous functions CGP with $\mu = 2 \ldots 22$ and $\lambda = 2048 \ldots 4096$ performs best. It is worth investigating $\lambda = 8 \ldots 32$ in cases where large λ values do not result in fast convergence.
- For continuous functions the mutation rate may vary from 1% to 15% with higher mutation rates being more successful for larger genotypes.

We will extend the benchmark set in our future work to more popular functions, like classification and image-processing tasks, and approach the questions regarding similarity of inner mechanisms to GP. Additionally we will try understand properly the ambivalent nature of CGP making it successful for combinatorial and continuous benchmarks.

REFERENCES

[1] P. Kaufmann. *Adapting Hardware Systems by Means of Multi-Objective Evolution.* Logos Verlag, Berlin, 2013.
[2] P. Kaufmann and M. Platzner. Multi-objective Intrinsic Hardware Evolution. In *Intl. Conf. Military Applications of Programmable Logic Devices (MAPLD)*, 2006.
[3] J. McDermott, D. R. White, S. Luke, L. Manzoni, M. Castelli, L. Vanneschi, W. Jaśkowski, K. Krawiec, R. Harper, K. D. Jong, and U.-M. O'Reilly. Genetic programming needs better benchmarks. In *Proceedings of the 14th international conference on Genetic and Evolutionary Computation Conference*, GECCO '08, pages 791–798, Philadelphia, 2012. ACM.
[4] L. Sekanina, J. A. Walker, P. Kaufmann, C. Plessl, and M. Platzner. Evolution of Electronic Circuits. In *Cartesian Genetic Programming*, Natural Computing Series, pages 125–179. Springer Berlin Heidelberg, 2011.
[5] J. A. Walker, J. F. Miller, P. Kaufmann, and M. Platzner. Problem Decomposition in Cartesian Genetic Programming. In *Cartesian Genetic Programming*, Natural Computing Series, pages 35–99. Springer Berlin Heidelberg, 2011.
[6] D. R. White, J. McDermott, M. Castelli, L. Manzoni, B. W. Goldman, G. Kronberger, W. Jaskowski, U.-M. O'Reilly, and S. Luke. Better GP Benchmarks: Community Survey Results and Proposals. *Genetic Programming and Evolvable Machines*, 14(1):3–29, Mar. 2013.

GECCO '17 Companion, Berlin, Germany
 978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3075980

Table 2: Evaluation of CGP parameters. Not optimized parameters are marked with an "–". The comparison prefers conventional (1+4) CGP, as iRace budget is set to 2000 for all configurations and challengers have more parameters to optimize. The results are measured in number of fitness evaluations. Best results are printed in *bold*. n_c and n_r - number of CGP columns and rows; m - mutation rate; $T_{\textbf{start}}$ and $T_{\textbf{stop}}$ - starting and stopping temperatures for SA. CE at $z = 99\%$.

goal function	algo-rithm	evolved parameters							termination[no. evaluations]			Comp. Effort	restart at eval.
		n_c	n_r	μ	λ	m[%]	T_{start}	T_{stop}	1Q	median	3Q		
(2,2,1) add	1+4 CGP	200	–	–	–	2.1112	–	–	14916	26532	49840	160753	91840
	1+λ CGP	100	200	–	3	0.3215	–	–	11316	18933	28797	89280	34350
	μ+λ CGP	200	50	1	1	0.3803	–	–	8114	**13129**	21723	**67860**	19849
	SA	200	2	–	–	1.8976	1299	0.0348	12242	20052	35411	109530	42284
(3,3,1) add	1+4 CGP	200	–	–	–	2.1512	–	–	113168	194120	326156	**689115**	689112
	1+λ CGP	150	1	–	3	1.9464	–	–	105789	178344	302211	929794	581961
	μ+λ CGP	100	4	1	3	0.8396	–	–	122460	190539	330936	1018919	451407
	SA	70	4	–	–	1.3706	4671	0.4366	88335	**149817**	246126	750368	621896
(4,4,1) add	1+4 CGP	200	–	–	–	1.2341	–	–	424924	697152	1182452	2830424	2404400
	1+λ CGP	300	2	–	2	0.6852	–	–	303080	501550	698950	2206982	1680482
	μ+λ CGP	100	4	1	1	1.1503	–	–	364545	545438	936699	2469195	2097544
	SA	150	3	–	–	0.6693	3610	0.6437	283038	**400832**	723341	**2034761**	1422236
(2,2) mul	1+4 CGP	100	–	–	–	2.9542	–	–	3452	5564	9136	28434	14864
	1+λ CGP	100	100	–	3	0.8680	–	–	2121	3417	5474	**16512**	9009
	μ+λ CGP	100	30	1	1	1.4332	–	–	2079	**3322**	5465	17349	7279
	SA	30	14	–	–	2.4941	58	0.0889	2661	4183	6801	21275	9959
(3,3) mul	1+4 CGP	2000	–	–	–	0.5008	–	–	274228	447220	722280	2103815	1787156
	1+λ CGP	200	20	–	2	0.2988	–	–	149824	288368	459822	1203021	1203020
	μ+λ CGP	150	30	1	2	0.2971	–	–	130250	224178	498888	1382722	361496
	SA	200	100	–	–	0.1622	3336	0.0870	84844	**148145**	356305	**949607**	169289
7-parity	1+4 CGP	300	–	–	–	1.2582	–	–	175628	271048	427788	1347746	645976
	1+λ CGP	300	8	–	2	0.7142	–	–	100408	186250	262668	762572	381284
	μ+λ CGP	300	2	1	2	0.9089	–	–	118996	186674	291118	696589	696588
	SA	150	8	–	–	0.7584	1528	0.2000	87773	**140463**	238599	**539214**	458054
8-parity	1+4 CGP	2000	–	–	–	0.9057	–	–	336420	461948	739504	2113156	1374636
	1+λ CGP	200	6	–	3	1.0381	–	–	310524	486894	798396	2408346	932859
	μ+λ CGP	300	6	1	1	0.5578	–	–	192417	**323192**	455204	1404562	702280
	SA	300	4	–	–	0.6733	417	0.3479	213877	329472	479532	**1196482**	1196482
9-parity	1+4 CGP	2000	–	–	–	0.8718	–	–	628536	1011220	1718660	5380705	1487336
	1+λ CGP	150	3	–	2	0.7050	–	–	617418	959194	1570728	2859287	2859286
	μ+λ CGP	300	3	1	1	0.8519	–	–	512420	755543	1239866	3073095	1774561
	SA	300	10	–	–	0.3784	2209	0.2907	392406	**579111**	910828	**2209561**	1876989
Koza-2	1 + 4 CGP	150	–	–	–	5	–	–	0.0095	0.0099	0.0325	–	–
	1 + λ CGP	150	3	–	128	2	–	–	0.0091	0.0098	0.0364	–	–
	μ + λ **CGP**	**150**	**3**	**18**	**2048**	**10**	–	–	**0.0085**	**0.0099**	**0.0140**	–	–
Koza-3	1 + 4 CGP	150	–	–	–	7	–	–	0.0104	0.0325	0.0328	–	–
	1 + λ CGP	120	10	–	16	2	–	–	0.0087	0.0099	0.0325	–	–
	μ + λ **CGP**	**80**	**20**	**14**	**4096**	**5**	–	–	**0.0091**	**0.0100**	**0.0327**	–	–
Nguyen-4	1 + 4 CGP	120	–	–	–	10	–	–	0.0120	0.0324	0.0487	–	–
	1 + λ CGP	40	8	–	64	15	–	–	0.0129	0.022	0.0395	–	–
	μ + λ CGP	**60**	**6**	**18**	**2048**	**10**	–	–	**0.0101**	**0.0283**	**0.0498**	–	–
Nguyen-5	1 + 4 CGP	60	–	–	–	7	–	–	0.0090	0.0100	0.0240	–	–
	1 + λ CGP	150	10	–	16	2	–	–	0.0099	0.0099	0.0229	–	–
	μ + λ **CGP**	**150**	**20**	**22**	**4096**	**1**	–	–	**0.0085**	**0.0096**	**0.0100**	–	–
Nguyen-6	1 + 4 CGP	100	–	–	–	2	–	–	0.0270	0.0382	0.0392	–	–
	1 + λ **CGP**	**60**	**20**	–	**8**	**1**	–	–	**0.0091**	**0.0191**	**0.0381**	–	–
	μ + λ CGP	80	14	–	4096	5	–	–	0.0100	0.0381	0.0407	–	–
Nguyen-7	1 + 4 CGP	200	–	–	–	7	–	–	0.0157	0.0262	0.0534	–	–
	1 + λ **CGP**	**120**	**8**	–	**4096**	**7**	–	–	**0.0099**	**0.01866**	**0.0382**	–	–
	μ + λ CGP	150	6	2	32	2	–	–	0.0116	0.0216	0.0288	–	–
Nguyen-8	1 + 4 CGP	150	–	–	–	15	–	–	0.0084	0.0111	0.0415	–	–
	1 + λ CGP	80	10	–	16	2	–	–	0.0072	0.0084	0.0098	–	–
	μ + λ **CGP**	**150**	**6**	**2**	**32**	**2**	–	–	**0.0072**	**0.0088**	**0.0095**	–	–
Nguyen-9	1 + 4 **CGP**	**150**	–	–	–	**15**	–	–	**0.2475**	**0.4184**	**1.2077**	–	–
	1 + λ CGP	200	4	–	16	7	–	–	0.2707	0.6189	1.0801	–	–
	μ + λ CGP	120	20	22	4096	15	–	–	0.5325	0.7245	1.0079	–	–
Nguyen-10	1 + 4 CGP	60	–	–	–	20	–	–	0.5728	0.9185	1.1150	–	–
	1 + λ CGP	120	10	–	4096	20	–	–	0.3718	0.5727	0.7346	–	–
	μ + λ **CGP**	**150**	**20**	**8**	**4096**	**15**	–	–	**0.2975**	**0.4020**	**0.5921**	–	–
Keijzer-4	1 + 4 CGP	22	–	–	–	5	–	–	3.6828	3.6828	3.6828	–	–
	1 + λ CGP	200	20	–	16	7	–	–	2.1038	2.3413	2.4953	–	–
	μ + λ **CGP**	**120**	**20**	**22**	**1024**	**10**	–	–	**2.0837**	**2.2254**	**2.3484**	–	–
Keijzer-6	1 + 4 CGP	100	–	–	–	2	–	–	0.3229	0.4883	0.6438	–	–
	1 + λ CGP	60	20	–	64	10	–	–	0.1538	0.2184	0.3445	–	–
	μ + λ **CGP**	**200**	**20**	**6**	**256**		–	–	**0.0516**	**0.1008**	**0.2390**	–	–
Pagie-1	1 + 4 CGP	150	–	–	–	20	–	–	31.5965	34.0846	35.2309	–	–
	1 + λ CGP	200	20	–	512	10	–	–	14.9535	21.4781	30.7461	–	–
	μ + λ **CGP**	**200**	**20**	**14**	**256**	**15**	–	–	**14.7931**	**21.3225**	**30.1226**	–	–