# UNIVERSITY BIRMINGHAM University of Birmingham Research at Birmingham

# Self-adaptation via multi-objectivisation: an empirical study

Qin, Xiaoyu; Lehre, Per Kristian

DOI: 10.1007/978-3-031-14714-2\_22

License: None: All rights reserved

Document Version Other version

#### Citation for published version (Harvard):

Qin, X & Lehre, PK 2022, Self-adaptation via multi-objectivisation: an empirical study. in G Rudolph, AV Kononova, H Aguirre, P Kerschke, G Ochoa & T Tušar (eds), Parallel Problem Solving from Nature – PPSN XVII: 17th International Conference, PPSN 2022, Dortmund, Germany, September 10–14, 2022, Proceedings, Part I. 1 edn, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 13398 LNCS, Springer, pp. 308–323, The seventeenth International Conference on Parallel Problem Solving from Nature, Dortmund, Germany, 10/09/22. https://doi.org/10.1007/978-3-031-14714-2\_22

Link to publication on Research at Birmingham portal

#### **General rights**

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

•Users may freely distribute the URL that is used to identify this publication.

•Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.

•User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?) •Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

#### Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

### Appendix 1 Omitted algorithms in this paper

Algorithm 3 Multi-objective sorting mechanism [33]

**Require:** Population sizes  $\lambda \in \mathbb{N}$ . Population  $P_t \in \mathcal{Y}^{\lambda}$ . Fitness function f.

Sort P<sub>t</sub> into strict non-dominated fronts F<sup>t</sup><sub>0</sub>, F<sup>t</sup><sub>1</sub>,... based on f<sub>1</sub>(x, χ) := f(x) and f<sub>2</sub>(x, χ) := χ.
for F = F<sup>t</sup><sub>0</sub>, F<sup>t</sup><sub>1</sub>,... do

3: Sort  $\mathcal{F}$  such that  $f_1(\mathcal{F}(1)) > f_1(\mathcal{F}(2)) > \dots$ 

4:  $P_t := (\mathcal{F}_0^t, \mathcal{F}_1^t, \ldots).$ 5: return  $P_t$ .

#### Algorithm 4 Strict non-dominated sorting [33]

**Require:** Population sizes  $\lambda \in \mathbb{N}$ . Population  $P \in \mathbb{Z}^{\lambda}$ , where  $\mathbb{Z}$  is a finite state space. Objective functions  $f_1, f_2, \ldots : \mathcal{Z} \to \mathbb{R}$  (assume to maximise all objective functions). 1: for each individual P(i) do 2: Set  $S_i := \emptyset$  and  $n_i := 0$ . 3: for  $i = 1, \ldots, \lambda$  do 4: for  $j = 1, \ldots, \lambda$  do if  $P(i) \prec P(j)$  based on  $f_1, f_2, \ldots$  then 5:6:  $S_i := S_i \cup \{P(i)\},\$ else if  $P(j) \prec P(i)$  based on  $f_1, f_2, \ldots$  then 7: 8:  $n_i := n_i + 1$ , 9: else if  $f_{\ell}(P(i)) = f_{\ell}(P(j))$  where  $\ell = 1, 2, \dots$  then 10:if  $P(i) \notin S_j$  then  $S_i := S_i \cup \{P(i)\}$  else  $n_i := n_i + 1$ . if  $n_i = 0$  then  $\mathcal{F}_0 = \mathcal{F}_0 \cup \{P(i)\}.$ 11: 12: Set k := 0. 13: while  $\mathcal{F}_k \neq \emptyset$  do 14: $Q := \emptyset.$ for each individual  $P(i) \in \mathcal{F}_k$  and  $P(j) \in S_i$  do 15:16:Set  $n_i := n_i - 1$ . if  $n_i = 0$  then  $Q := Q \cup \{P(j)\}.$ 17:18:Set k := k + 1,  $\mathcal{F}_k := Q$ . 19: return  $\mathcal{F}_0, \mathcal{F}_1, \ldots$ 

Algorithm 5 Multi-objective sorting mechanism (alternative)

**Require:** Population sizes  $\lambda \in \mathbb{N}$ . Population  $P_t \in \mathcal{Y}^{\lambda}$ . Fitness function f.

1: Sort  $P_t$  into  $P_t^1, P_t^1, \ldots$  where  $P_t^1$  containing all individuals with the highest fitness  $f, P_t^2$  containing all individuals with the 2nd highest fitness  $f, \ldots$ 

2: for  $i = 1, ..., \lambda$  do Set  $\hat{\chi} := -\infty$ . 3: for  $Q = P_t^1, P_t^1, ...$  do 4: Find  $(x', \chi')$  which is the element with the highest  $\chi$  in Q. 5:6: if  $Q \neq \emptyset$  and  $\chi' > \hat{\chi}$  then  $P_t(i) := (x', \chi') \text{ and } \hat{\chi} := \chi'.$ 7: Pop  $(x', \chi')$  from Q. 8: Break. 9: 10: return  $P_t$ .

Algorithm 6	$(\mu, \lambda$	) selection
-------------	-----------------	-------------

**Require:** Population size  $\lambda \in \mathbb{N}$ . Parameter  $\mu \in [\lambda]^3$ . 1:  $I_t \sim \text{Unif}([\mu]).$ 2: return  $I_t$ .

Algorithm 7 Fitness-first sorting mechanism [7]

**Require:** Population sizes  $\lambda \in \mathbb{N}$ . Population  $P_t \in \mathcal{Y}^{\lambda}$ . Fitness function f. 1: Sort  $P_t$  such that  $P_t(1) \succeq \cdots \succeq P_t(\lambda)$ , according to  $(x,\chi) \succeq (x',\chi') \Leftrightarrow f(x) > f(x') \lor (f(x) = f(x') \land \chi \ge \chi').$ 2: 3: return  $P_t$ .

<sup>&</sup>lt;sup>3</sup> For any  $n \in \mathbb{N}$ , we define  $[n] := \{1, \ldots, n\}$ 

## Appendix 2 Omitted statistical results of experiments

Table 2: Statistical results of experiments on random NK-LANDSCAPE problems. The *p*-values of each algorithm come from Wilcoxon rank-sum tests between the algorithm and MOSA-EA.

MOSA-EA	3-tour.EA	$(\mu, \lambda)\mathbf{EA}$	$1 + (\lambda, \lambda))GA$	FastGA (1	(1 + 1)EA	$\mathbf{SA-}(1,\lambda)\mathbf{EA}$	RLS	UMDA	$_{\rm cGA}$	RS	k Stat.
79.2846	79.2846	78.2089	79.2846	76.9230	76.6613	74.8418	71.3547	74.8631	72.9964	66.6591	ع Median
-	0.8805	0.7985	0.9299	0.4215	0.2668	0.0226	6.5e-08	0.0213	2.3e-04	2.1e-22	p-value
82.5270	78.7832	79.2680	77.1520	76.1340	75.5792	71.0248	68.3100	73.2968	69.5499	66.4442	10 Median
-	0.0063	0.0030	2.2e-09	1.1e-12	5.0e-18	3.5e-34	2.6e-34	2.0e-15	1.5e-26	2.6e-34	p-value
80.4417	76.9053	76.0777	74.6407	74.2253	73.7253	67.8968	66.4446	70.9576	66.5517	66.2055	15 Median
-	1.1e-17	1.3e-20	5.2e-33	1.8e-33	2.6e-34	2.6e-34	2.6e-34	5.5e-22	2.6e-34	2.6e-34	<sup>13</sup> p-value
78.5247	75.3662	74.2580	73.0882	72.8783	72.8025	66.0533	64.9865	69.6786	64.4191	66.1233	oo Median
-	1.2e-31	4.0e-33	2.6e-34	2.6e-34	2.6e-34	2.6e-34	2.6e-34	7.0e-31	2.6e-34	2.6e-34	<sup>20</sup> p-value
77.5024	74.8115	73.4398	71.9623	71.7564	70.8648	65.1886	64.3685	68.5683	63.1222	66.2207	or Median
-	1.4e-33	2.6e-34	2.6e-34	2.6e-34	2.6e-34	2.6e-34	2.6e-34	2.6e-34	2.6e-34	2.6e-34	<sup>20</sup> p-value



Fig. 11: The *p*-values of Wilcoxon rank-sum tests between the algorithms and the MOSA-EA on 100 random k-SAT instances. The y-axis is log-scaled.



Fig. 12: The *p*-value of Wilcoxon rank-sum test between Open-WBO and the MOSA-EA on 100 random k-SAT instances. The y-axis is log-scaled.