Plant Propagation Parameterization: Offspring & Population Size

Marleen de Jonge[0000−0003−4911−2647]
Daan van den Berg[0000−0001−5060−3342]

Informatics Institute, University of Amsterdam
m.r.h.dejonge@student.uva.nl, d.vandenberg@uva.nl

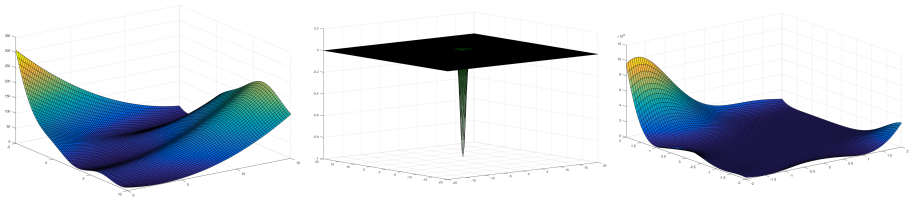
Abstract. We investigate a wide range of interdependent parameter values for the number of offspring and the population size in the plant propagation algorithm. An 'optimal window' of parameter values is found, for which the algorithm performs substantially better on five benchmark test functions. Moreover, apart from being within or outside the window, values appear to be largely interchangable, making the algorithm largely independent from specific settings of these parameters.

Keywords: Plant Propagation Algorithm · Evolutionary Algorithms · Parametrization · Metaheuristics · Combinatorial Optimization

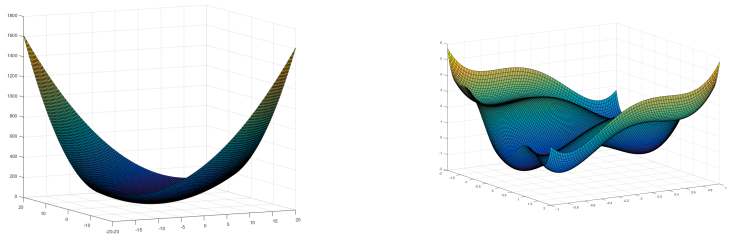
1 Introduction

In recent years, a booming interest has emerged for the implementation of nature-inspired evolutionary algorithms on combinatorial optimization problems. However, the large variety of new algorithms is often not tested thoroughly, and therefore "threatening to lead the area of metaheuristics away from scientific rigor" [7]. As one example, the vast number of possible parameter configurations often form new combinatorial optimization problems in themselves, and determining the 'perfect' settings thereby, becomes a challenging task. Unfortunately, relatively little analysis is done on optimal parameterization in many evolutionary algorithms, despite being "essential for good algorithm performance" [6].

One such algorithm is the Plant Propagation Algorithm (PPA), introduced by Abdellah Salhi and Eric S. Fraga in 2011, which works well on a broad variety of benchmark functions, as well as discrete problems such as the traveling salesman problem, university timetabling and even artistic optimization tasks [4] [8] [5] [1] [3]. After randomly initializing $popSize$ initial individuals, the objective values $f(x)$ are calculated, and normalized to [0,1] by $z(x_i) = \frac{f(x_{max})-f(x_i)}{f(x_{max})-f(x_{min})}$ after which the hyperbolic tangent $F(x_i) = \frac{1}{2}(tanh(4 \cdot z(x_i)-2)+1)$ is applied to "[provide a means of emphasising further better solutions over those which are not as good]" [4]. Then, the number of offspring per individual is proportional to its fitness as $n(x_i) = \lceil n_{max} F(x_i) r \rceil$, whereas their mutability is *inversely* proportional as $d_{r,j} = 2(1 - F(x_i))(r - 0.5)$, in which j is the respective dimension; r is a different random number for both equations. Fitter individuals will



(a) Branin (b) Easom (c) Goldstein Price



(d) Martin Gaddy (e) Six Hump Camel

Fig. 1: The five 2D benchmark test functions on which the different parameter values of $popSize$ and n_{max} were tested during a minimization task.

thereby spawn relatively many offspring with smaller mutations, whereas unfitter individuals produce fewer offspring with larger mutations. In the final step, the offspring get added to the population, from which $popSize$ fittest individuals constitute the next generation.

As Salhi and Fraga themselves point out in their seminal work "[parameter values chosen appear suitable for the problems investigated, [but] little analysis has been performed to understand the impact of these parameters]" [4]. Here, we perform a first investigation into two interdependent parameters: the population size ($popSize$) and the maximum number of offspring per individual (n_{max}). We explore 400 parameter settings of PPA on five different continuous 2D benchmark test functions (Fig.1) from the original study.

2 Experiment & Results

To rigorously test the two parameters, the algorithm was ran with 400 different parameter combinations on all five benchmark test functions with their global minimum set to zero. We choose $1 \leq popSize \leq 40$, $1 \leq n_{max} \leq 10$, and the median best objective value of ten separate runs of the algorithm was taken for each parameter combination (Fig. 2). One run consisted of 10,000 evaluations, amounting in a total of 200 million function evaluations for the whole experiment. It should be noted though, that $popSize$ and n_{max} are interdependent in the number of offspring per generation, so these parameter settings are expected to bring along different numbers of generations per combination.

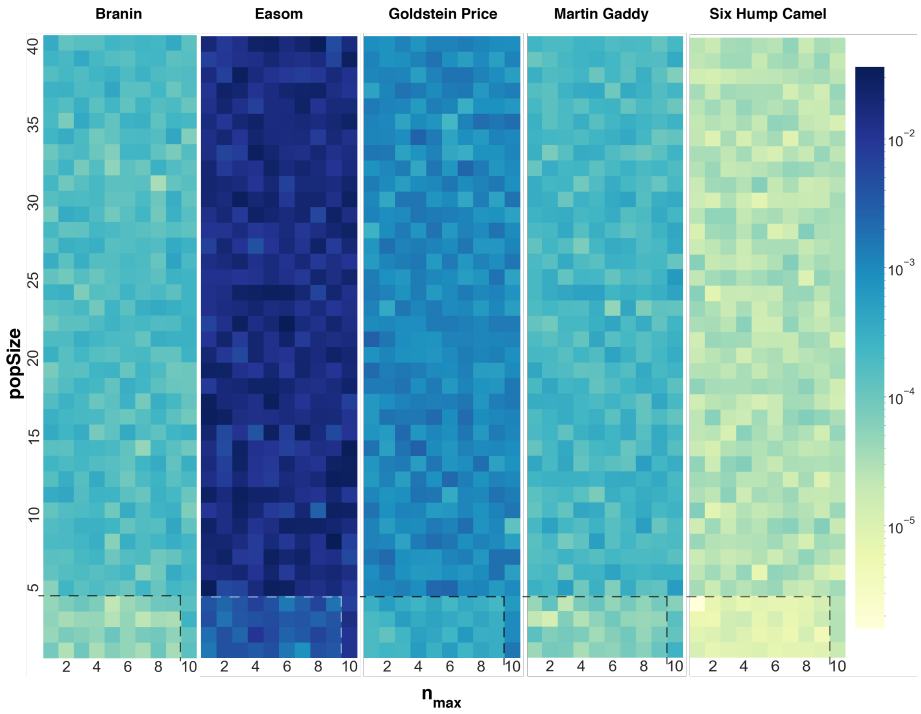


Fig. 2: Results for a wide range of parameter values for $popSize$ and n_{max} in PPA on five different benchmark test functions with a zero-normalized global minimum. Each cell represents the median best objective value over 10 runs of 10,000 function evaluations. A consistent window of parameter values (dashed lines) provides significantly better results for all five functions.

The results show a window of parameter configurations that consistently work better than others. For all five of the tested benchmark functions, parameter values $1 \leq popSize \leq 4$ and $1 \leq n_{max} \leq 9$ result in significantly better found objective values (Table 1). But moreover, the differences in standard deviation are small both inside and outside the window, which leaves room for a tantalizing point of discussion.

3 Discussion & Future work

Ceteris paribus, it looks like the plant propagation algorithm is largely unsensitive to parameter settings. As can be seen from the heatmap, the only real difference appears whether one chooses a setting inside the window or outside the window, as for both these subsets, their respective standard deviation on the best performance is low.

	Inside window		Outside window	
Function	μ	σ	μ	σ
Branin	5.79e-05	2.46e-05	1.94e-04	7.70e-05
Easom	3.83e-03	1.84e-03	1.66e-02	6.87e-03
Goldstein Price	3.41e-04	1.25e-04	1.11e-03	4.53e-04
Martin Gaddy	6.52e-05	2.94e-05	2.34e-04	9.57e-05
Six Hump camel	8.90e-06	3.22e-06	2.96e-05	1.22e-05

Table 1: Overview of the mean and standard deviation per benchmark test function, both within and outside the window of optimal configurations. For each of the functions, the mean best value inside is better than outside this window range. But the standard deviations are low, so apart from the inside-outside-window choice, the algorithm is quite robust against different parameter settings.

Whether these results are consistent across a broader range of benchmark test functions, or more real-worldly problems such as timetabling or the traveling salesman problem, remains to be seen. Finally, we encourage other teams to check, replicate and extend our results, either with or without our publicly accessible repository [2].

References

1. Geleijn, R., van der Meer, M., van der Post, Q., van den Berg, D.: The plant propagation algorithm on timetables: First results. EVO* 2019 p. 2 (2019)
2. de Jonge, M.: Source code of algorithm used in the experiment. https://github.com/marleentheyoung/PPA_parametrization, accessed: 2020-30-3
3. Paauw, M., Van den Berg, D.: Paintings, polygons and plant propagation. In: International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar). pp. 84–97. Springer (2019)
4. Salhi, A., Fraga, E.S.: Nature-inspired optimisation approaches and the new plant propagation algorithm (2011)
5. Selamoğlu, B.İ., Salhi, A.: The plant propagation algorithm for discrete optimisation: The case of the travelling salesman problem. In: Nature-inspired computation in engineering, pp. 43–61. Springer (2016)
6. Smit, S.K., Eiben, A.E.: Comparing parameter tuning methods for evolutionary algorithms. In: 2009 IEEE congress on evolutionary computation. pp. 399–406. IEEE (2009)
7. Sörensen, K.: Metaheuristics—the metaphor exposed. International Transactions in Operational Research **22**(1), 3–18 (2015)
8. Vrielink, W., van den Berg, D.: Fireworks algorithm versus plant propagation algorithm. In: IJCCI. pp. 101–112 (2019)