**Noname manuscript No.**
(will be inserted by the editor)

# A Haploid-Diploid Evolutionary Algorithm Optimizing Nanoparticle Based Cancer Treatments

**Michail-Antisthenis Tsompanas · Larry Bull · Andrew Adamatzky · Igor Balaz**

Received: date / Accepted: date

**Abstract** This paper uses a recent explanation for the fundamental haploid-diploid lifecycle of eukaryotic organisms to present an evolutionary algorithm that differs from previous known work that is using diploid representations. A form of the Baldwin effect has been identified as inherent to the evolutionary mechanisms of eukaryotes and a simplified version is presented here which maintains such behaviour. Using a well-known abstract tuneable model, it is shown that varying fitness landscape ruggedness varies the benefit of haploid-diploid algorithms. Moreover, the methodology is applied to optimise the targeted delivery of a therapeutic compound utilizing nanoparticles to cancerous tumour cells with the multicellular simulator PhysiCell.

**Keywords** Baldwin effect · diploid · NK model · cancer · nano-particles · PhysiCell

## 1 Introduction

The vast majority of work within evolutionary computation has used an underlying haploid representation scheme; individuals are each one solution to the given problem [6]. Typically, bacteria contain one set of genes, whereas the more complex eukaryotic organisms — such as plants and animals — are predominantly diploid, containing two sets of genes [19]. A small body of work exists using a diploid representation scheme, i.e., individuals carry two solutions to the given problem. In

M.-A. Tsompanas, A. Adamatzky
Unconventional Computing Laboratory, University of the West of England, Bristol BS16 1QY, UK
E-mail: antisthenis.tsompanas@uwe.ac.uk

L. Bull
Department of Computer Science and Creative Technologies, University of the West of England, Bristol BS16 1QY, UK

I. Balaz
Laboratory for Meteorology, Physics and Biophysics, Faculty of Agriculture, Trg Dositeja Obradovica 8, University of Novi Sad, 21000, Novi Sad, Serbia

such cases recombination typically occurs between corresponding haploids/genes in each parent, essentially doubling the standard process, and a dominance scheme is utilized to reduce the diploid down to a traditional haploid solution for evaluation. That is, as individuals carry two sets of genes/variables, a heuristic is included to choose which of the genes to use (see [3] for a review).

Eukaryotes exploit a so-called haploid-diploid cycle where haploid cells are brought together to form the diploid cell/organism. At the point of reproduction by the cell/organism, the haploid genomes within the diploid each form haploid gamete cells that (may) join with a haploid gamete from another cell/organism to form a diploid (Fig. 1). Specifically, each of the two genomes in an organism is replicated, with one copy of each genome being crossed over. In this way copies of the original pair of genomes may be passed on, mutations aside, along with two versions containing a mixture of genes from each. Previous explanations for the emergence of the alternation between the haploid and diploid states are typically based upon its being driven by changes in the environment (after [14]). Recently, an explanation for the haploid-diploid cycle in eukaryotes has been presented [5] which also explained other aspects of their sexual reproduction, including the use of recombination, based upon the Baldwin effect [1]. The Baldwin effect is here defined as the existence of phenotypic plasticity that enables an organism to exhibit a different (better) fitness than its genome directly represents. Over time, as evolution is guided towards such regions under selection, higher fitness alleles/genomes which rely less upon the phenotypic plasticity can be discovered and become assimilated into the population.

The rest of the paper is arranged as follows: the next section presents the new understanding of how eukaryotic organisms evolve, a new simplified haploid-diploid algorithm is then presented, which maintains the basic mechanisms of the natural case. Finally, the new approach is applied to a high-throughput multicellular simulator to find potentially new therapeutic designs that maximise cancer tumour regression.

## 2 Background and Related Works

Key to the new explanation for the evolution of eukaryotes is to view the process from the perspective of the constituent haploids: a diploid organism may be seen to simultaneously represent two points in the underlying haploid fitness landscape. The fitness associated with those two haploids is therefore that achieved in their combined form as a diploid; each haploid genome will have the same fitness value and that will almost certainly differ from that of their corresponding haploid organism due to the interactions between the two genomes. That is, the effects of haploid genome combination into a diploid can be seen as a simple form of phenotypic plasticity for the individual haploids before they revert to a solitary state during reproduction. In this way evolution can be seen to be both assigning a single fitness value to the region of the landscape between the two points represented by a diploid's constituent haploid genomes and altering the shape of the haploid fitness landscape. In particular, the latter enables the landscape to be smoothed under a rudimentary Baldwin effect process [9], whilst the former can be seen to represent a simple form of generalization over the landscape. Note this is in direct contrast to typical cases of bacteria — and evolutionary algorithms — where in-

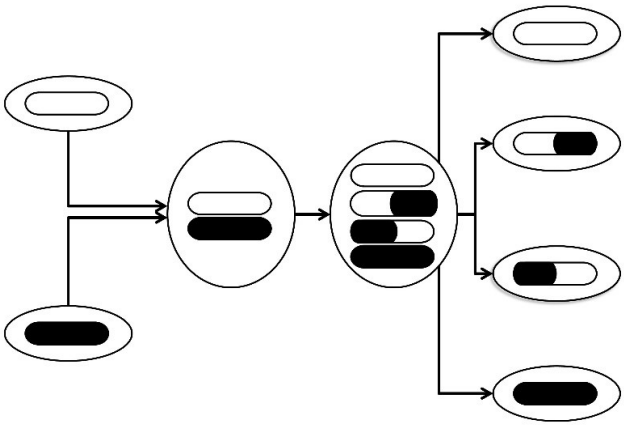
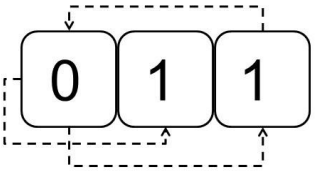


Fig. 1: Two-step meiosis with recombination under haploid-diploid reproduction as seen in most eukaryotic organisms (after [17]).

dividuals represent a single point in the (haploid) fitness landscape only and the fitness assigned is that due solely to their given gene combination. The same is also true in all known previous diploid representation schemes.

Numerous explanations exist for the benefits of recombination in both natural (e.g., [2]) and artificial systems (e.g., [18]). The latter focusing solely upon haploid genomes and neither considering the potential Baldwin effect under the haploid-diploid cycle. The role becomes clear under the new view: recombination facilitates genetic assimilation within the simple form of the Baldwin effect. If the haploid pairing is beneficial and the diploid is chosen under selection to reproduce, the recombination process can bring an assortment of those partnered genes together into new haploid genomes. In this way the fitter allele values from the pair of partnered haploids may come to exist within individual haploids more quickly than the under mutation alone (see [5] for full details). Hence, in the emergence of more complex organisms, natural evolution appears to have discovered a more sophisticated approach to navigating their fitness landscapes.

The Baldwin effect has long been used within evolutionary computation (after [9]). This paper aims to show how the benefits of a haploid-diploid cycle can be exploited as a form of evolutionary computation. However, rather than just adopt nature's scheme under which a single individual requires both haploid genomes to be evaluated (as in [5]), a simpler scheme is proposed as a form of post-processing for a traditional evolutionary algorithm. This is first explored using the NK model of fitness landscapes.




| $K_0$ $K_1$ | Fitness |
|---|---|
| 0 0 | 0.76 |
| 0 1 | 0.23 |
| **1 0** | **0.56** |
| 1 1 | 0.98 |

Fig. 2: An example NK model ($N$=3, $K$=1) showing how the fitness contribution of each gene depends on $K$ random genes (left). Therefore there are $2^{(K+1)}$ possible allele combinations per gene, each of which is assigned a random fitness. Each gene of the genome has such a table created for it (right, centre gene shown). Total fitness is the normalized sum of these values.

## 3 Methods

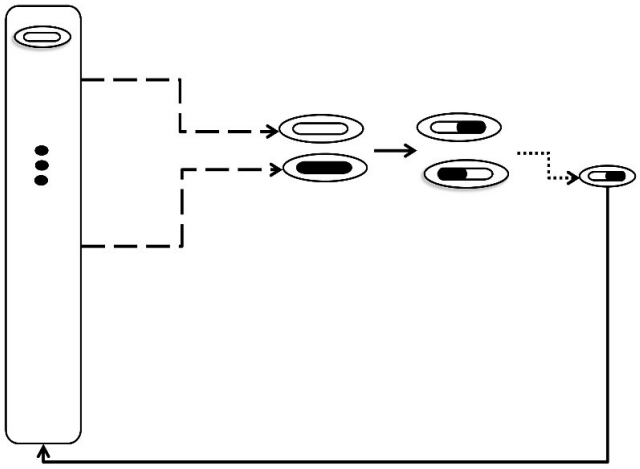
### 3.1 The NK model

The NK model [11] was introduced to allow the systematic study of various aspects of fitness landscapes (see [12] for an overview). In the standard model, the features of the fitness landscapes are specified by two parameters: $N$, the length of the genome; and $K$, the number of genes that has an effect on the fitness contribution of each (binary) gene. Thus, increasing $K$ with respect to $N$ increases the epistatic linkage, increasing the ruggedness of the fitness landscape. The increase in epistasis increases the number of optima, increases the steepness of their sides, and decreases their correlation.

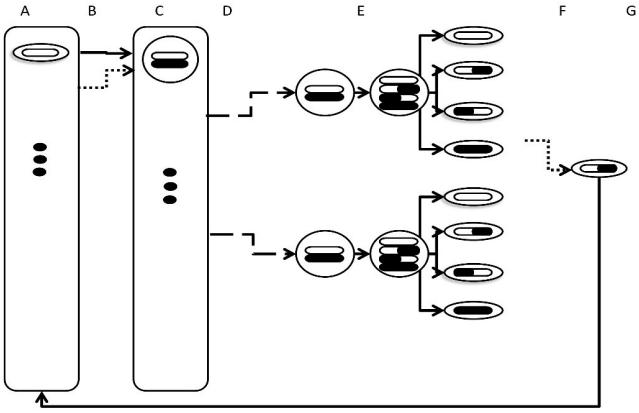
The model assumes all intragenome interactions are so complex that it is only appropriate to assign random values to their effects on fitness. Therefore for each of the possible $K$ interactions a table of $2^{(K+1)}$ fitnesses is created for each gene with all entries in the range 0.0 to 1.0, such that there is one fitness for each combination of traits (Fig. 2). The fitness contribution of each gene is found from its table. These fitnesses are then summed and normalized by $N$ to give the selective fitness of the total genome. The results reported in the next section are the average of 10 runs (random starting populations) on each of 10 NK functions, i.e. 100 runs, for 20,000 generations. Here $0 \leq K \leq 15$, for $N = 50$ and $N = 100$.

### 3.2 A Simple Haploid-Diploid Algorithm

This. Figure 3(a) shows a schematic of a traditional evolutionary algorithm (EA) which exploits binary tournament selection, one-point recombination, single-point mutation (randomly chosen gene), and creates one offspring per cycle (steady state) which is evaluated and replaces the worst individual in the population here. Figure 3(b) shows how the learning mechanism described above is implemented on top of that process. As can be seen: a traditional population of evaluated haploid individuals is maintained (A); a temporary population of diploid solutions is created from them by copying each haploid individual and then another haploid



(a)




(b)

Fig. 3: A schematic of the traditional evolutionary algorithm (a) and of the simple haploid-diploid algorithm (b).

is chosen at random (B), with the fitness of the two haploids averaged (C); binary tournament selection then uses those fitnesses to pick two diploid parents (D); the haploid-diploid reproduction cycle with two-step meiosis as shown in Fig. 1 is then used for the two chosen parents (E); one of the resulting haploids is chosen at random, mutated (single-point), and evaluated (F); the offspring haploid is inserted into the original population replacing the worst individual (G).

### 3.3 PhysiCell: A Physics-based Multicellular Simulator

Among an increasing amount of computational models [15] studying different aspects of cancer physiology, PhysiCell [8] is one of the leading ones. The open source

simulator is based on a biotransport solver (BioFVM [7]) and simulates a multicellular environment. While PhysiCell simulates cell cycling, death states, volume changes, mechanics, orientation and motility, it relies on BioFVM to simulate substrate secretion, diffusion, uptake, and decay. A significant advantage of PhysiCell is its open-source code that enables addition of new environmental substrates, cell types, and systems of cells, resulting in a general-purpose tool for investigating systems with multiple kinds of cells. This includes the ability to design cell-cell interaction rules to create a multicellular cargo delivery system that actively delivers a cancer therapeutic compound beyond regular drug transport limits to hypoxic cancer regions. We are currently exploring the use of evolutionary computing and other related techniques to optimise the design of such nano-particle (NP) delivery systems [16].

To evaluate the efficiency of the design of these NP delivery systems, the 2-D anti-cancer biorobots scenario of PhysiCell v.1.5.1 [8] was studied. This scenario utilizes three types of agents to simulate a high-throughput testing of a simple targeted drug delivery therapy. Namely, these types of agents are cancer cells, worker cells and cargo cells. Cancer cells consume oxygen and secrete a chemoattractant. The resulted gradient in oxygen concentration is employed to steer NPs, simulated as worker cells. These worker cells can be bonded with cargo cells, simulating the therapeutic compound. When a worker cell carries a cargo cell, it executes a random walk (migration) towards the gradient of the oxygen and, thus, towards accumulation of cancer cells. Whereas, when a worker cell does not carry a cargo cell it executes a random walk towards the area of the cargo cells. These random walks or migrations are controlled by input parameters of the simulator, in the range $[0, 1]$, with 0 representing Brownian motion and 1 deterministic motion.

Finally, cargo cells simulating the therapeutic compound, can attract worker cells by exuding another simulated chemoattractant (which diffuses under BioFVM rules). As described before, worker cells can carry the cargo cells and deposit them in the affinity of cancer cells, resulting in apoptosis of these cells. The specific proximity is given by the parameter defined as cargo release $O_2$ threshold.

As per the initial example [8] and other relevant studies [16], in the 2-D anti-cancer biorobots scenario an initial 200 $\mu m$ radius tumour is simulated to grow for 7 days. Then, 450 cargo cells and 50 worker cells are added in a simulated vein close to the tumour. Note here that while in previous studies a random number of each type of cells with its mean as in the aforementioned was added, here we add exactly 450 and 50 cells for every simulation to alleviate one factor of stochasticity. The simulated drug delivery system is simulated for 3 more days and then the results are analyzed.

One paradigm of this simulation (whole 10 days) takes approximately 5 minutes of wall-clock time on an Intel® Xeon® CPU E5-2650 at 2.20GHz with 64GB RAM using 8 of the 48 cores. To accelerate the computations and further alleviate the effect of the stochastic nature of the simulator on the results, a single tumour was used for testing every possible individual in the search space. For each test, one pre-grown tumour (for 7 days) was loaded to the simulator and the treatment was applied immediately. The test was finalized after 3 days from the introduction of the treatment, resulting in a minimization of wall-clock time to approximately 1,5 minutes. A static sampling approach is used, where the average of the outputs after 5 runs of the simulator with the same set of parameters was examined. The

Table 1: Unaltered parameters of PhysiCell simulator.

| Parameter | Value |
|---|---|
| Maximum attachment distance | 18 $\mu m$ |
| Minimum attachment distance | 14 $\mu m$ |
| Worker apoptosis rate | 0 $min^{-1}$ |
| Worker migration speed | 2 $\mu m/min$ |
| Worker $O_2$ relative uptake | 0.1 $min^{-1}$ |
| Cargo $O_2$ relative uptake | 0.1 $min^{-1}$ |
| Cargo apoptosis rate | 4.065e-5 $min^{-1}$ |
| Maximum relative cell adhesion distance | 1.25 |
| Maximum elastic displacement | 50 $\mu m$ |
| Damage rate | 0.03333 $min^{-1}$ |
| Repair rate | 0.004167 $min^{-1}$ |
| Drug death rate | 0.004167 $min^{-1}$ |
| Cargo relative adhesion | 0 |
| Cargo relative repulsion | 5 |
| Elastic coefficient | 0.05 $min^{-1}$ |
| Motility shutdown detection threshold | 0.001 |
| Attachment receptor threshold | 0.1 |

objective was determined as the remaining amount of cancer cells in the simulated area after the 3 days of treatment.

The search space was defined as a 6-dimensional space, with the 6 most prolific parameters for the behaviour of worker cells (or simulated NPs). Namely, the parameters under investigation were: the attached worker migration bias [0,1]; the unattached worker migration bias [0,1]; worker relative adhesion [0,10]; worker relative repulsion [0,10]; worker motility persistence time (minutes) [0,10]; and the cargo release $O_2$ threshold ($mmHg$) [0,20]. The rest of the parameters on the simulator are not altered from the initial distribution of the simulator [8] and depicted in Table 1.

## 4 Results

First the abstract model was utilised to study the effect of fitness landscape rugednes on the efficiency of the haploid-diploid EA (HDEA). Figure 4 shows example results from running both the standard EA and the HDEA on various NK fitness landscapes. Here population size $P = 30$. As can be seen, when $K > 4$, the HDEA performs best for $N = 50$ and $K > 2$ for $N = 100$ (T-test, $p < 0.05$). Thus, as anticipated, the simple Baldwin effect process proves beneficial with increased fitness landscape ruggedness due to its ability to smooth the underlying shape. Figure 5 shows examples of how this is also true for different $P$, although the benefit is lost for higher $K$ when $P = 10$. Related to this, since the HDEA makes a temporary population of diploids containing extra copies of randomly chosen haploid solutions, it might be argued that a larger population is available to selection than in the standard EA. Moreover, as the underlying traditional haploid population converges upon higher fitness solutions, the random sampling could be increasing their number and, thereby, altering the comparative selection pressure over time. However, results from simply creating a temporary haploid population of size $2P$, in the same way as the temporary diploid population, does not alter

performance significantly (not shown here, e.g., see [10] for discussions of dynamic population sizing in general).

Whilst it is beyond the scope of this paper to exhaustively review previous diploid EAs (DEAs)(see [3]), a simple diploid version of the traditional EA has been created and explored. Here the fitness of the diploid is the average of its two constituent haploids and the corresponding haploids undergo one-point recombination. For comparison, without a dominance mechanism, populations contain either half as many individuals or are run for half as many generations as the EA and HDEA to maintain the same number of function evaluations. The results find no significant difference in performance over the traditional EA for any of the parameters explored here (not shown).

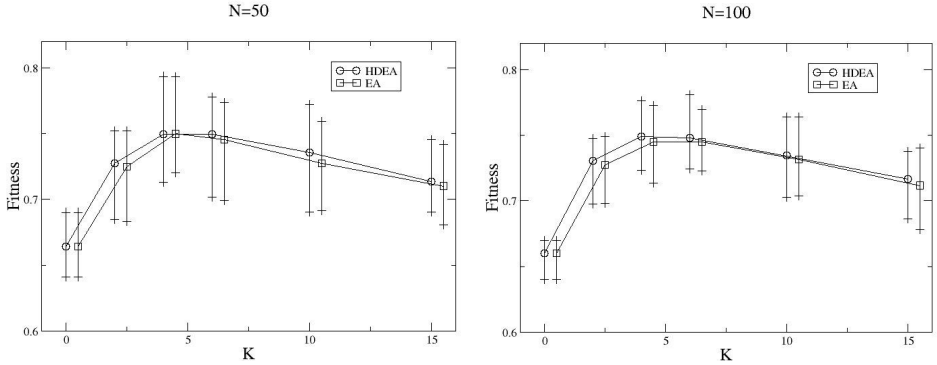



Fig. 4: Showing examples of the fitness reached after 20,000 generations on landscapes of various size ($N$) and ruggedness ($K$). Error bars show min and max values.

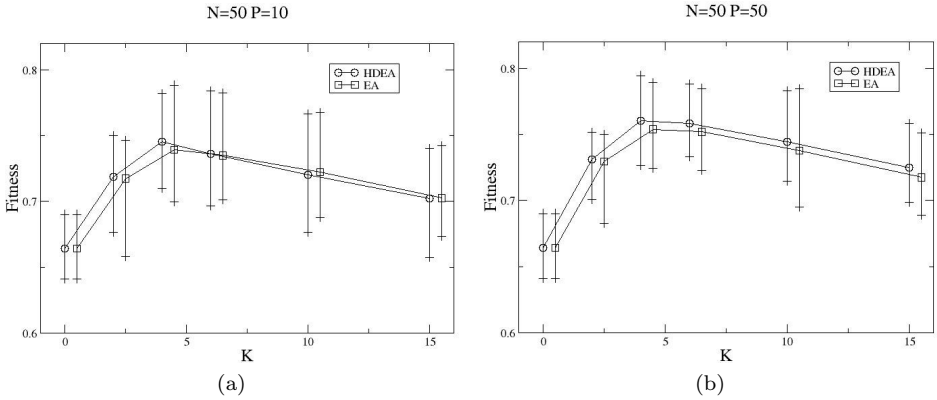



Fig. 5: Showing examples of the fitness reached after 20,000 generations with differing population sizes ($P$).

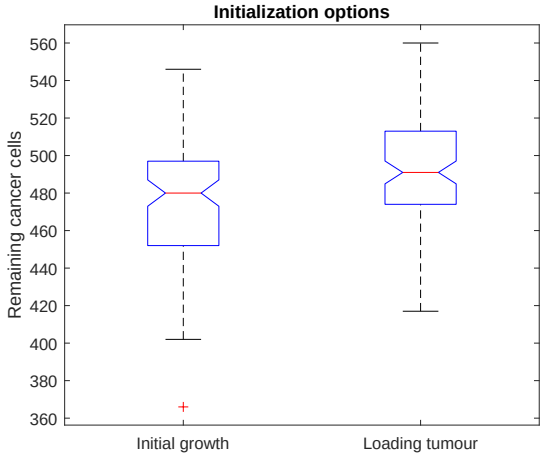



Fig. 6: Boxplot of 100 samples for each of the initialization options.

The findings with this abstract model are now explored in the context of simulating nano-particle therapy delivery for cancer tumour regression within PhysiCell v.1.5.1 (see [15] for an overview of computational modelling in cancer biology).

Initially, as described in Section 3.3, the option to load a tumour rather than simulate its growth for 7 days was investigated. In Fig. 6 the boxplot of 100 simulations for each initialization option with the same input parameters is illustrated. When comparing the initial growth (7 days tumour growth and 3 days treatment simulation, mean=475.06, SD=32.9, median=480, kurtosis=3.3515) with the loading tumour alternative (loading a tumour and 3 days treatment simulation, mean=494.12, SD=29.11, median=491, kurtosis=2.7698), the latter produces more consistent results (based on smaller standard deviation and kurtosis). Additional to the aforementioned acceleration of computations (from 5 minutes to 1,5 minutes) the loading tumour was selected for the tests presented in the following.

To study the performance of HDEA, another control algorithm was utilized to optimize the behaviour/design of worker cells, namely a steady-state genetic EA. The population size was set to $P = 50$, the selection and replacement tournament size to $T = 3$, a uniform crossover probability to $X = 80\%$ and a per allele mutation rate to $\mu = 20\%$ with a uniform random step size of range $s = [-5, 5]\%$. The HDEA was set up with the same parameters as the EA in order for the comparison to be meaningful. All comparison runs started by evaluating a randomly produced, same for each run, initial population ($P = 50$) under PhysiCell simulator, and then using the corresponding EA to evolve the design of worker cells, with a computational budget of 100 individual evaluations (100 individuals $\times$ 5 samples = 500 PhysiCell simulations). In total, 30 comparison runs were executed.

In Fig. 7 the evolution of the best individuals found by the two algorithms is illustrated. Specifically, the average and confidence level at 95% for the best individuals in all 30 runs are considered. Throughout the evolution steps it is apparent that the HDEA algorithm is generally finding better solutions faster (it learns faster). Moreover, the final average of best solutions found by HDEA is better than the one by the genetic algorithm. The smaller range of the 95%

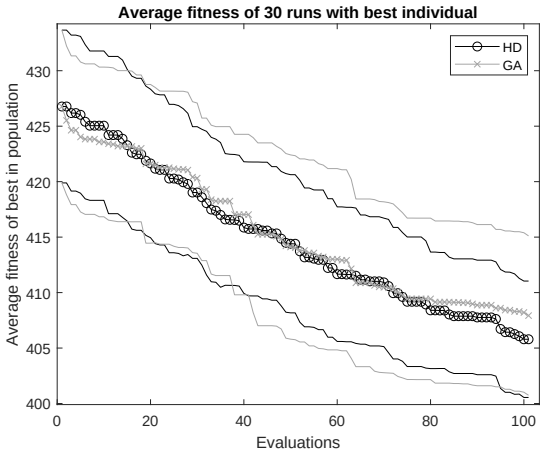



Fig. 7: Average and confidence levels (95%) of the best individuals per evolution step for both algorithms for all 30 runs.

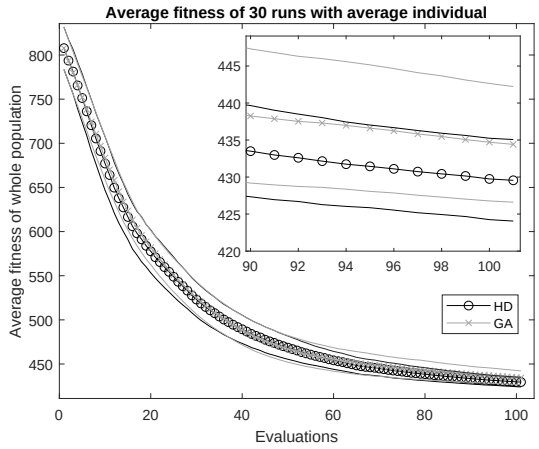



Fig. 8: Average and confidence levels (95%) of all the individuals per evolution step for both algorithms for all 30 runs.

confidence levels of HDEA reveal a better consistency in the solutions found by this algorithm.

Figure 8 shows the relative performance of the average solutions over time for both approaches. As can be seen, the HDEA finds fitter solutions. Note the zoomed in region of evaluations 90 to 100 for a clearer comparison. Although, after 100 evaluations, the best solutions (Fig. 7) are not statistically significantly better (Wilcoxon signed-rank test, $p = 0.3763$), the average solutions (Fig. 8) are (Wilcoxon signed-rank test, $p = 0.0256$). It can be noted that the best solution found by the HDEA was significantly better (Wilcoxon signed-rank test, $p = 0.0215$) for the first ten runs of the thirty shown here.

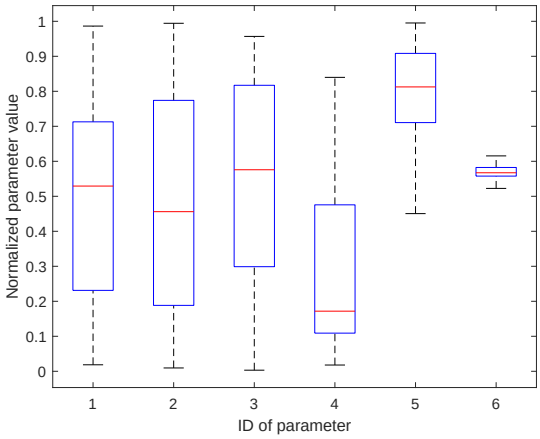



Fig. 9: Boxplot of parameters of best individuals found by GA (in normalized ranges). Parameters: 1) attached worker migration bias 2) the unattached worker migration bias 3) worker relative adhesion 4) worker relative repulsion 5) worker motility persistence time 6) and the cargo release $O_2$ threshold.

In Figs. 9 and 10 the boxplots of the parameters of the best individual discovered during the 30 runs by GA and HDEA, respectively, are presented. In Figs. 11 and 12 the scatter plots of the parameters of the best individual discovered during the 30 runs by GA and HDEA, respectively, are depicted. It is clear that the most prolific parameter value for optimizing the design of NPs is the cargo release $O_2$ threshold parameter. The majority of solutions are quite close to 11 $mmHg$, similar to findings from previous works [8,16]. Although, for three of the parameters the results can not be conclusive (namely, attached and unatached worker migration bias and worker relative adhesion having almost uniform distribution like boxplots), the graphs for the other two parameters can convey the fact of the solutions being skewed towards smaller values for worker relative repulsion and higher values for worker motility persistence time.

## 5 Conclusion

In the standard evolutionary computing approach each individual solution can be seen to represent a single point in the fitness landscape. Typically, the same is true of bacteria in natural evolution. It has recently been suggested that natural evolution is using a more sophisticated approach with eukaryotes, exploiting a generalization process, whereby each individual represents a region in the fitness landscape [5]. Of course, landscape smoothing can be achieved by numerous mechanisms (after [9]) but they all require extra fitness evaluations. The scheme presented in this paper is intended to exploit the Baldwin effect through what is essentially simple population manipulation rather than through altering the underlying representation and evaluations of the standard evolutionary computing approach.

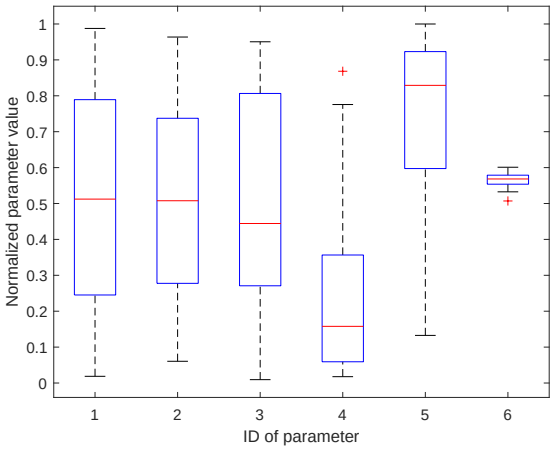



Fig. 10: Boxplot of parameters of best individuals found by HDEA (in normalized ranges). Parameters: 1) attached worker migration bias 2) the unattached worker migration bias 3) worker relative adhesion 4) worker relative repulsion 5) worker motility persistence time 6) and the cargo release $O_2$ threshold.

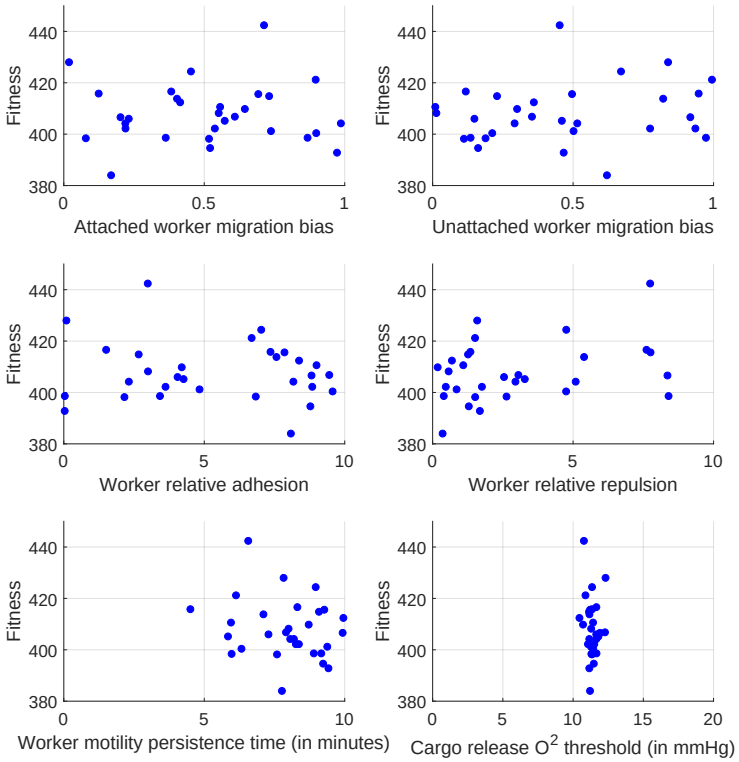



Fig. 11: Scatter plot of parameters of best individuals found by GA for all 30 runs.

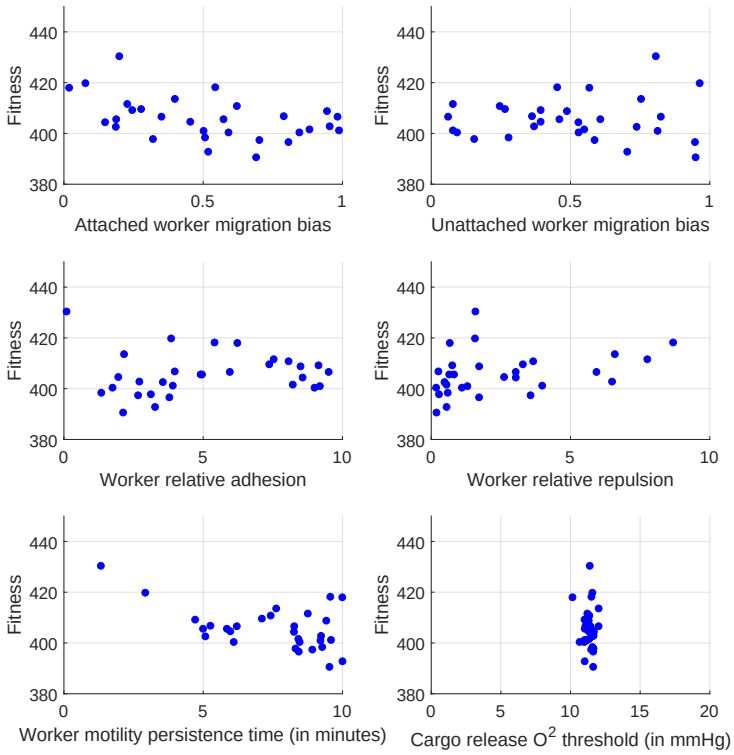



Fig. 12: Scatter plot of parameters of best individuals found by HDEA for all 30 runs.

It can also be noted that the shape of the fitness landscape varies based upon the haploid genomes, which exist within a given population at any time and how they are paired. This is significant since, as has been pointed out for coevolutionary fitness landscapes [4], such movement potentially enables the temporary creation of neutral paths, where the benefits of (static) landscape neutrality are well-established (after [13]).

The proposed HDEA method was also compared with a simple haploid EA in a more complicated simulator (PhysiCell). Again, the HDEA seems to perform better than the traditional and well-established haploid method. After analyzing the results of the methodology on the cancer treatment simulator, it can be concluded that it reaches fitter solutions faster, despite the high stochasticity injected into the fitness landscape to capture some of the dynamics of the biology. Current work is exploring the inclusion of a sexual selection-like process into the HDEA to further improve performance.

**Acknowledgements** This work was supported by the European Research Council under the European Union's Horizon 2020 research and innovation programme under grant agreement No. 800983.

## References


1. Baldwin, J.M.: A new factor in evolution. The american naturalist **30**(354), 441–451 (1896)
2. Bernstein, H., Bernstein, C.: Evolutionary origin of recombination during meiosis. BioScience **60**(7), 498–505 (2010)
3. Bhasin, H., Mehta, S.: On the applicability of diploid genetic algorithms. AI & society **31**(2), 265–274 (2016)
4. Bull, L.: On coevolutionary genetic algorithms. Soft Computing **5**(3), 201–207 (2001)
5. Bull, L.: The evolution of sex through the baldwin effect. Artificial life **23**(4), 481–492 (2017)
6. Eiben, A.E., Smith, J.E., et al.: Introduction to evolutionary computing, vol. 53. Springer (2003)
7. Ghaffarizadeh, A., Friedman, S.H., Macklin, P.: Biofvm: an efficient, parallelized diffusive transport solver for 3-d biological simulations. Bioinformatics **32**(8), 1256–1258 (2015)
8. Ghaffarizadeh, A., Heiland, R., Friedman, S.H., Mumenthaler, S.M., Macklin, P.: Physicell: An open source physics-based cell simulator for 3-d multicellular systems. PLoS computational biology **14**(2), e1005991 (2018)
9. Hinton, G.E., Nowlan, S.J.: How learning can guide evolution. Complex systems **1**(3), 495–502 (1987)
10. Karafotias, G., Hoogendoorn, M., Eiben, Á.E.: Parameter control in evolutionary algorithms: Trends and challenges. IEEE Transactions on Evolutionary Computation **19**(2), 167–187 (2014)
11. Kauffman, S., Levin, S.: Towards a general theory of adaptive walks on rugged landscapes. Journal of theoretical Biology **128**(1), 11–45 (1987)
12. Kauffman, S.A.: The origins of order: Self-organization and selection in evolution. OUP USA (1993)
13. Kimura, M.: The neutral theory of molecular evolution. Cambridge University Press (1983)
14. Margulis, L., Sagan, D.: Origins of sex: three billion years of genetic recombination (1986)
15. Metzcar, J., Wang, Y., Heiland, R., Macklin, P.: A review of cell-based computational modeling in cancer biology. JCO clinical cancer informatics **2**, 1–13 (2019)
16. Preen, R.J., Bull, L., Adamatzky, A.: Towards an evolvable cancer treatment simulator. BioSystems **182**, 1–7 (2019)
17. Smith, J.M., Szathmary, E.: The major transitions in evolution. Oxford University Press (1997)
18. Spears, W.M.: Evolutionary algorithms: the role of mutation and recombination. Springer Science & Business Media (2013)
19. Trun, N., Trempy, J.: Fundamental bacterial genetics. John Wiley & Sons (2009)