Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

In silico optimization of cancer therapies with multiple types of nanoparticles applied at different times

Michail-Antisthenis Tsompanas[a,*], Larry Bull[b], Andrew Adamatzky[a], Igor Balaz[c]

[a] *Unconventional Computing Laboratory, University of the West of England, Bristol BS16 1QY, UK*
[b] *Department of Computer Science and Creative Technologies, University of the West of England, Bristol BS16 1QY, UK*
[c] *Laboratory for Meteorology, Physics and Biophysics, Faculty of Agriculture, Trg Dositeja Obradovica 8, University of Novi Sad, Novi Sad, 21000, Serbia*

ARTICLE INFO

Article history:
Received 12 August 2020
Accepted 23 November 2020
Available online xxx

Keywords:
Optimization
Cancer treatment
Nano-particles
PhysiCell
Metameric representation

ABSTRACT

Background and Objective: Cancer tumors constitute a complicated environment for conventional anti-cancer treatments to confront, so solutions with higher complexity and, thus, robustness to diverse conditions are required. Alternations in the tumor composition have been documented, as a result of a conventional treatment, making an ensemble of cells drug resistant. Consequently, a possible answer to this problem could be the delivery of the pharmaceutic compound with the assistance of nano-particles (NPs) that modify the delivery characteristics and biodistribution of the therapy. Nonetheless, to tackle the dynamic response of the tumor, a variety of application times of different types of NPs could be a way forward. *Methods:* The *in silico* optimization was investigated here, in terms of the design parameters of multiple NPs and their application times. The optimization methodology used an open-source simulator to provide the fitness of each possible treatment. Because the number of different NPs that will achieve the best performance is not known a priori, the evolutionary algorithm utilizes a variable length genome approach, namely a metameric representation and accordingly modified operators. *Results:* The results highlight the fact that different application times have a significant effect on the robustness of a treatment. Whereas, applying all NPs at earlier time slots and without the ordered sequence unveiled by the optimization process, proved to be less effective. *Conclusions:* The design and development of a dynamic tool that will navigate through the large search space of possible combinations can provide efficient solutions that prove to be beyond human intuition.

1. Introduction

Cancerous tumors have proved to have high heterogeneity in types of cancer cells [1,2]. Due to that fact, the effectiveness of therapeutic compounds, which are usually carefully designed in terms of targeting specific cells, can not be equally sufficient in the dynamic scenarios they face. Moreover, as the ensemble of cells are subjected to a therapy, they are stressed and this can result to higher heterogeneity [3,4], thus, higher drug resistance is developed [2,5].

Because of the aforementioned observations, higher complexity should be pursued in future investigated treatments. For instance, the functionalization of nano-particles (NPs) with the therapeutic compound is enhancing the biodistribution, tumor penetration and cellular uptake in targeted tissues [6–11]. Diverse behaviour of NPs have also been examined as a possible solution to tackle the adaptability of the evolving environment of a tumor towards a “static” treatment, either by developing multi-functional or multi-stimuli-responsive nanoparticles [12–14]. Moreover, the differentiation of the application times of each type of NPs, can prove to enhance the effectiveness, as different stages of the evolving tumor [15] can be targeted each time, namely a multistage treatment approach [16–18].

Consequently, the differentiation of application times of NPs within the same treatment against a tumor was investigated. The *in silico* optimization of a possible treatment was examined with an open-source multi agent cell simulator, namely PhysiCell [19]. Note here that there are several works on the mathematical mod-

[*] Corresponding author.
E-mail address: antisthenis.tsompanas@uwe.ac.uk (M.-A. Tsompanas).

https://doi.org/10.1016/j.cmpb.2020.105886

eling of the application of nanomedicine on cancerous tumors [20,21], that were utilized to design more effective therapies [6,22]. Specifically, PhysiCell has been previously used as a testbed for optimization techniques of different types [23–25]. The consideration of *in silico* optimization of anti-cancer therapies was previously considered with single-NP [23,26,27] and multi-NP therapies [28], however, without differentiation of the application times, similar to the way it was implemented in the present study.

An optimization problem was formalized with a search space defined by the application times and the physiological parameters of the NPs in the simulation, and with the fitness function as the remaining cancer cells. The treatments investigated as individuals in the evolutionary algorithm, were characterized by a variable-length genome, as the required complexity of the treatment (i.e. the amount of different NPs) is not known beforehand. This variable-length alternative is termed as metameric representation [29,30] and has been applied to a wide variety of optimization problems, where the solutions can be fragmented into identical segments, like locating turbines in wind farms [31] or positioning nodes for a network to optimize coverage [32]. Moreover, the operators of the evolutionary algorithm had to be adjusted to be able to apply on the metameric representation.

The optimized treatments found by the evolutionary algorithm proved to reach the highest possible complexity available by the simulator, however, this can prove to be challenging in practice. The fabrication of many different types of NPs would require different techniques and their combination may prove to have unexpected toxic effects. Thus, the evolutionary algorithm was updated with parsimony pressure to reduce the complexity of the discovered treatments, i.e. the different types of NPs utilized. Comparing the results of the two methodologies, there was no statistically different fitness provided by the discovered treatments, whereas the therapies provided by the updated technique were significantly less complex.

The results, also, revealed that when designing drug delivery systems consisting of different types of NPs, their robustness is susceptible to multiple application times. After conducting statistical analysis on results provided with multiple NPs applied at earlier, arbitrary chosen time slots and results provided with time slots indicated by the optimization procedure, the latter proved to have higher robustness. Proving that the selection of randomly chronologically ordered time slots, that happen to be early in the simulation procedure is not a viable technique.

2. In silico treatment evaluation

To test each different possible treatment, a sample project implemented with PhysiCell [19] simulator, namely “anti-cancer biorobots”, is utilized. The basic scenario provided under this sample project, initializes a cancerous tumor of radius 200 μm, by accumulating c. 570 simulated cancer cell agents attached to each other, located in the center of a simulated area. The division of these cell agents is applied for one week of simulated time, in order to demonstrate tumor growth. After that, the therapy is introduced in close proximity to the tumor, by adding 500 new agents; 10% of them are labeled as worker agents and represent the NPs, whereas the rest, 90% of them are labeled as cargo agents and represent the drug. For the rest of the simulation, that is defined as 3 days, the worker agents adhere to cargo agents, move towards the center of the simulated tumor (following an oxygen concentration gradient) and deposit the cargo agents. The fact that cargo agents are in the vicinity of cancer cell agents cause them to decay and eventually die, thus simulating the effect that the drug has to actual cancer cells.

The simulator was executed on a system with an Intel® Xeon® CPU E5-2650 at 2.20GHz (using 8 of the 48 cores) and 64GB RAM, and the results of 10 days simulated time were provided after 5 min. To accelerate the execution of the simulator and minimize the effects of its stochastic mechanisms, appropriate additions were applied in the source code [19] to load a tumor that was previously subjected to a simulated growth of one week. As a result, each possible treatment will be tested on the same initial tumor and the simulated time is now 3 days, leading to an execution of 1.5 min on the aforementioned computational system.

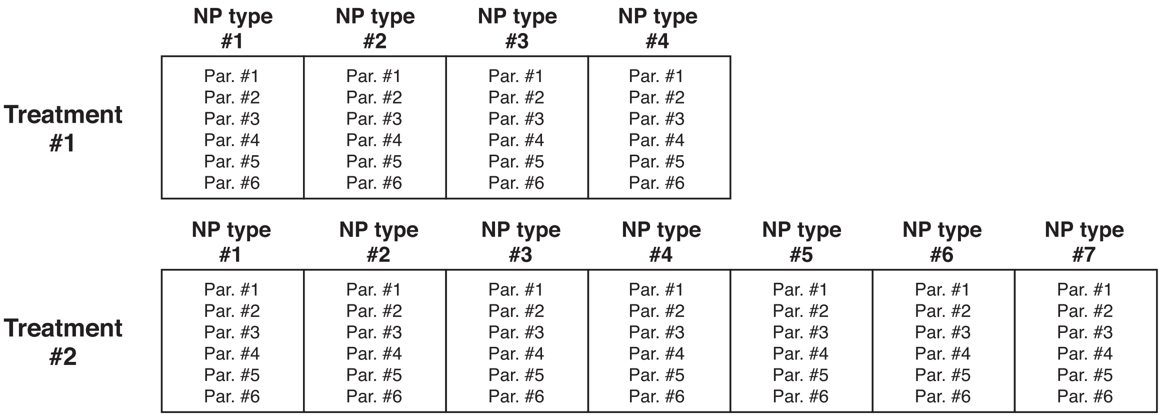
Table 1
Unaltered parameters of PhysiCell simulator.

Parameter	Value
Damage rate	0.03333 min^{-1}
Repair rate	0.004167 min^{-1}
Drug death rate	0.004167 min^{-1}
Elastic coefficient	0.05 min^{-1}
Cargo O_2 relative uptake	0.1 min^{-1}
Cargo apoptosis rate	4.065e-5 min^{-1}
Cargo relative adhesion	0
Cargo relative repulsion	5
Cargo release O_2 threshold	10 mmHg
Maximum relative cell adhesion distance	1.25
Maximum elastic displacement	50 μm
Maximum attachment distance	18 μm
Minimum attachment distance	14 μm
Motility shutdown detection threshold	0.001
Attachment receptor threshold	0.1
Worker migration speed	2 μm/min
Worker apoptosis rate	0 min^{-1}
Worker O_2 relative uptake	0.1 min^{-1}

The simulation of the release of multiple different NPs within a treatment, during different application times was developed. The application time of each type of NP in the simulated therapy was defined to be within the range of 0 to 2880 min (2 days). Note here, that the time from the initial application of the compound to the termination of the simulation is 3 days. Thus, even types of NPs with late application times, are given at least one day of simulated time to perform.

Here, the maximum of different types of NPs that can be inserted was set to 10. As mentioned previously, a total of 50 worker agents are introduced during the simulation, despite the amount of different types of NPs, in order to have comparable results. That amount of agents is equally divided among the diverse types of agents for every case. For instance, with 2 types of agents/NPs tests, 25 of each one are introduced; with 5 types of agents/NPs, 10 of each one are introduced and so forth.

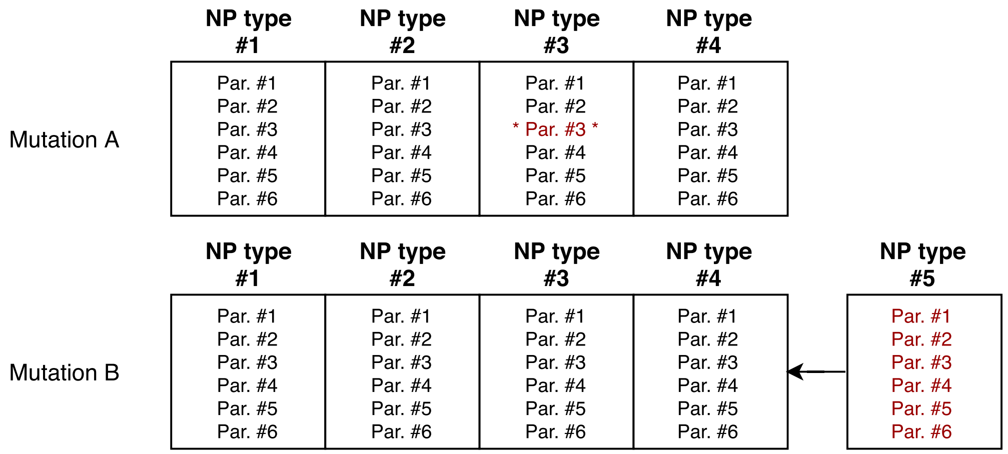
The parameters of the simulator are not changed from the original version of the simulator [19] and are given in Table 1. To formulate the design of the treatment as an optimization problem, the parameters that define each type of NPs can be located in a 6-dimensional space. Specifically the parameters under study are the following (and their ranges are in brackets): attached worker migration bias [0,1], unattached worker migration bias [0,1], worker relative adhesion [0,10], worker relative repulsion [0,10], worker motility persistence time (min) [0,10] and application time (min) [0,2880]. The attached and unattached worker migration bias determines the method of movement/migration of the worker agents, ranging from deterministic (value 1) to Brownian (value 0), when the are attached and unattached to a cargo agent. Worker motility persistence time controls the amount of time that worker agents move towards one direction before electing a new migration direction. Worker relative adhesion and repulsion describe the physical interaction of the worker agents with cargo and cancer cell agents, when they are in close proximity. Moreover, the fitness function needed to evaluate each solution is regarded as the remaining amount of cancer cell agents in the simulated area, after the simulated period of 3 days.



	NP type #1	NP type #2	NP type #3	NP type #4
Treatment #1	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6

	NP type #1	NP type #2	NP type #3	NP type #4	NP type #5	NP type #6	NP type #7
Treatment #2	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6

Fig. 1. Representation of two therapies.



	NP type #1	NP type #2	NP type #3	NP type #4
Mutation A	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 * Par. #3 * Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6

	NP type #1	NP type #2	NP type #3	NP type #4	NP type #5
Mutation B	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6	Par. #1 Par. #2 Par. #3 Par. #4 Par. #5 Par. #6

Fig. 2. Mutation operator.

3. Optimization methodology

Given that multiple types of NPs can be included in a candidate solution, but the best amount of different types is not known a priori, conventional optimization methodology and operators can not be utilized. On the contrary, the operators need to be adjusted to a representation of a solution with multiple segments (each one representing one type of NPs), without the amount of these segments being static. Consequently, a metameric representation [29,30] was implemented, with a limitation of a maximum of 10 different types of NPs introduced, for algorithmic reasons and because the higher complexity of these therapies is considered prohibitive for the fabrication of the treatment in the lab. For instance, two different treatments are illustrated in Fig. 1, one with four types of NPs and one with seven types of NPs.

In the adopted optimization methodology, the evolution of the solutions is achieved through a modified steady state GA methodology. The population size of possible solutions is set to $P = 20$, whereas a tournament of size $T = 2$ is set for the selection and replacement operators. A modified mutation operator was implemented. The appropriate changes in this operator are that once a parent is selected, then with 50% probability, either one of the parameters is altered (one randomly selected gene modified with random step size of $s = [-5; 5]\%$ - depicted in Fig. 2 as Mutation A, where parameter #3 of NP type #3 is altered), or one additional type of NP is added to the treatment (depicted in Fig. 2 as Mutation B), with random parameters. A custom crossover operator for metameric representations was not incorporated in this study.

In a second variation of the evolutionary methodology, instead of only adding one type of NP with the mutation operator, one type of NP can be removed with equal probability. Moreover, to control an anticipated emergence of bloat in the results parsimony pressure [33] was applied to the replacement operator. Namely, the offspring produced by the mutation operator, will have to be 10% fitter than an individual in the previous population to replace it in the next generation. This is true only for mutated offspring with higher genome length (different types of NPs), otherwise, if the two individuals are of similar genome lengths (or the offspring represents fewer types of NPs) only the fitness is used as a comparison measure without any adjustment.

Note here, that the initialization of the population is set with solutions of only one type of NPs, in order to have the lowest possible complexity. This option can contribute to bloat control (especially in early generations) and the limited search space that is created, may enhance the convergence rate, in contrast to cases with more complex initial populations [30,34].

The computational budget of each test of evolutionary optimization was set to 1000 evaluations with PhysiCell. Also, a 5-run static sampling technique was adopted, thus, the population is evolved for 200 generations (or evaluations of possible offspring solutions). The total of 1000 evaluations result in 1500 min of wall-clock time on the aforementioned computer system, which trans-

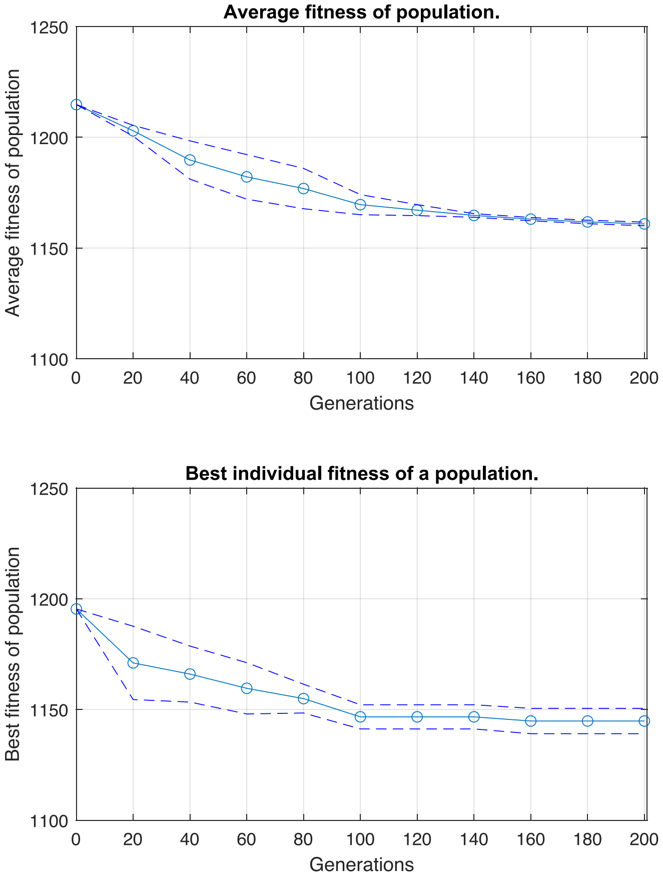


Fig. 3. Average and confidence levels (95%) results from 5 runs of GA. (a) Evolution of average fitness of the population and (b) evolution of the best individual in the population.

lates to 25 h per evolutionary optimization test. All tests were executed on the same initial population of 20 individuals/solutions, to render a valid statistical analysis and a fair comparison between different methods.

4. Results

4.1. Conventional mutation-only evolution

Initially, a steady state genetic algorithm with a conventional mutation operator (Genetic Algorithm with Conventional Mutation *GA-CM*) and no crossover operator was tested as a benchmark. The conventional mutation operator only changes one parameter of the 6-dimentional space specified in a previous section, representing the behaviour of one type of NPs (and not altering the amount of types of NPs simulated - only Mutation A as depicted in Fig. 2).

After, multiple executions (5 runs) of the *GA-CM* on the same initial population the results are presented in Fig. 3. The average and 95% confidence levels are illustrated. In Fig. 3(a) the average fitness of the population is presented (initial value 1215 and final value 1160 - a reduction of 4.4%), while in Fig. 3 (b) the best fitness is presented (initial value 1195 and final value 1145 - a reduction of 4.2%). Note here that no additional executions were run (more than 5) as the results are consistent - as depicted from the small range of the confidence levels in Fig. 3 - and the improvement in the final solutions is small. This is attributed to the low rate of exploration in the parameter space with the conventional mutation operator.

The association of the time of application of the NPs and the fitness of the treatment is depicted in Fig. 4. As expected, the earliest the NPs are introduced as a therapy, the better the results in terms of remaining cancer cell agents. In Fig. 4 all the new individuals tested throughout the evolution of all 5 runs were considered, and a linear regression model was fitted to illustrate the dependence of application times and fitness, without considering the rest of the parameters being changed with the mutation operator. The higher density on early application times unveils the fact that the GA converges fast to this area of application times search space and, then, slowly fine-tunes all the other parameters to minimize the fitness. The slope of the fitted line reveals the intuitive notion of the connection between earlier application times and better fitness.

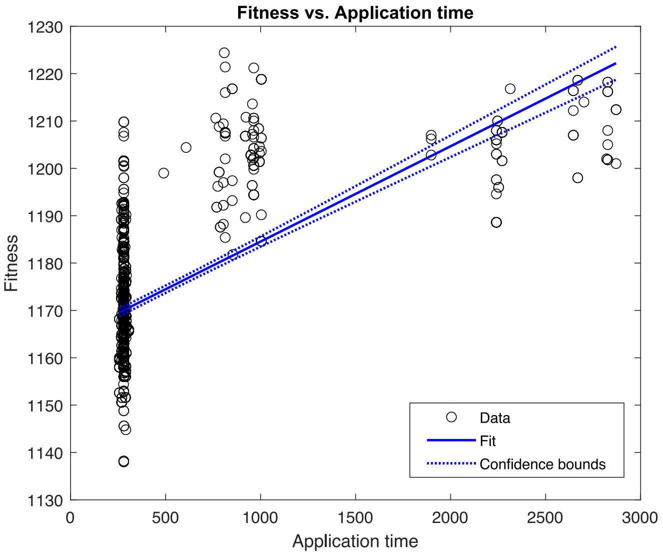


Fig. 4. Scatter plot, linear regression fit and confidence bounds of application times for types of NPs and their fitness as a treatment for all new individuals tested throughout all 5 runs.

4.2. Mutation with addition evolution

The application of the genetic algorithm with mutation operator of updated functionality (adding random types of NPs, thus, using both Mutations A and B in Fig. 2) is tested next (Genetic Algorithm with Mutation Additions *GA-MA*). The results after 10 runs with the updated mutation operator are summarized in Fig. 5. The data points presented are the average of the 10 runs, while the dashed lines refer to the 95% confidence levels. The average fitness of the population evolving throughout the generations is depicted in Fig. 5(a) with the average values indicating a gain in fitness of 60% (initial average at 1215 and final at 477). Moreover, the fitness of best individuals found throughout the evolution is displayed in Fig. 5(b) where a total advance of 61% in fitness is realised (initial best at 1195 and final at 458).

The composition (amount of different types of NPs) of the best solutions found throughout the evolution generations are illustrated in Fig. 6. The data points displayed are the average of the 10 runs, while the dashed lines refer to the 95% confidence levels. There is a steady increase in the amount of NPs added throughout the evolution, with the final composition converging to the maximum of 10 types. However, when comparing the fitness of the best individuals (Fig. 5(b)) for generation 100 onward, we realize that only slight improvements are achieved when more types of NPs are added. This situation is known as bloat and there are ways to avoid it, like parsimony control that is implemented in the following variation of the GA algorithm.

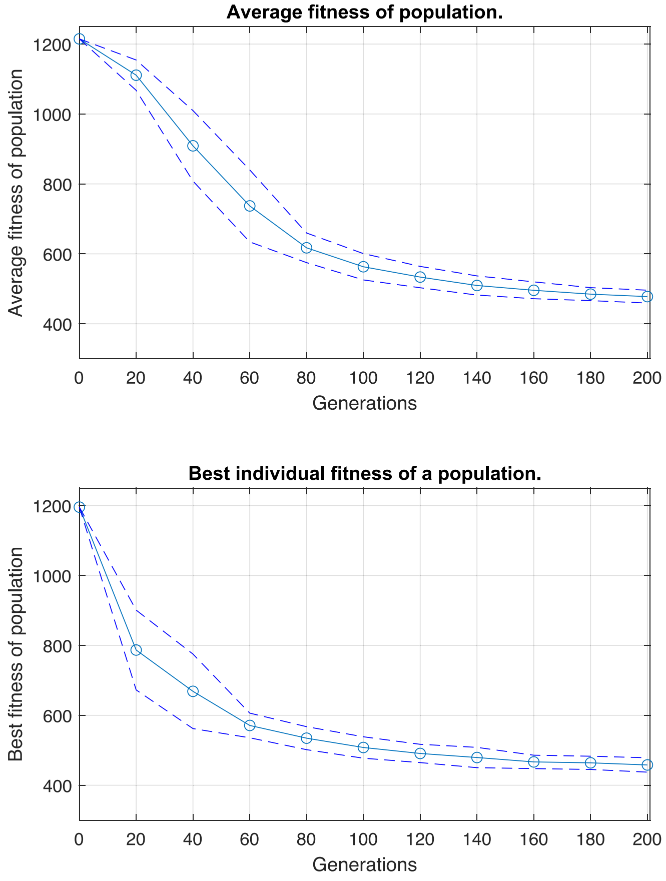


Fig. 5. Average and confidence levels (95%) results from 10 runs of GA with variable-length genome. (a) Evolution of average fitness of the population, (b) evolution of the best individual in the population.

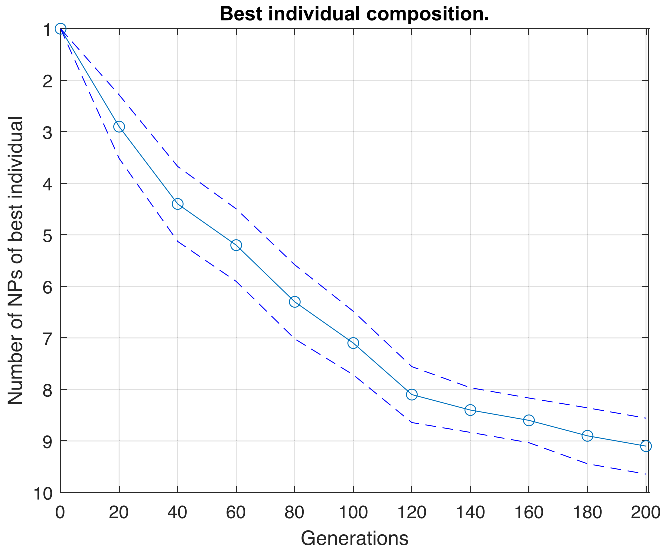
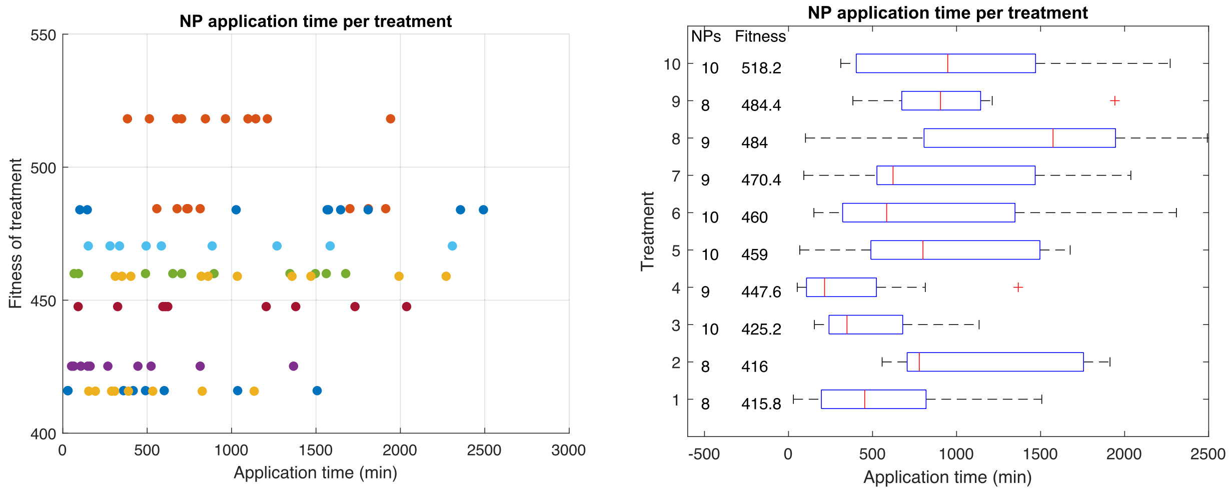


Fig. 6. Average and confidence levels (95%) results from 10 runs of GA with variable-length genome for the composition of the best solution.

In Fig. 7(a) a scatter plot is given, comparing the application time of each type of NPs comprising a treatment and the combinatorial treatment fitness. Each horizontal line of dots (with the same color), represent one treatment, thus, having the same fitness. The best individual/treatment found in each of the previous 10 runs were considered here. The association between the distribution of the application times of NPs and the fitness of the treatment has small variations, however, it seems that the better solutions (more efficient treatments) release their NPs earlier in the simulation, an expected outcome given the results from Fig. 4. To better illustrate a comparison between the different treatments found, Fig. 7(b) presents a boxplot of the application times per treatment. Also, the fitness of each treatment and the amount of different types of NPs are given for the best treatment found after each test. It can be concluded that treatments with small range of NP application times, skewed towards $t = 0$ are more efficient. Moreover, counter-intuitive conclusions can be derived from Fig. 7(b). Namely, comparing treatment 1 and 4, while treatment 1 has a median value of the application times on a later time slot, wider



(a) (b)

Fig. 7. (a) Scatter plot of application times for types of NPs and their collective fitness as a treatment (Each line of dots with the same color represents one combinatorial treatment - thus, the same fitness). (b) Boxplot of application times per best treatment found on each optimization test. Additional information in the figure are the fitness and amount of types of NPs per treatment.

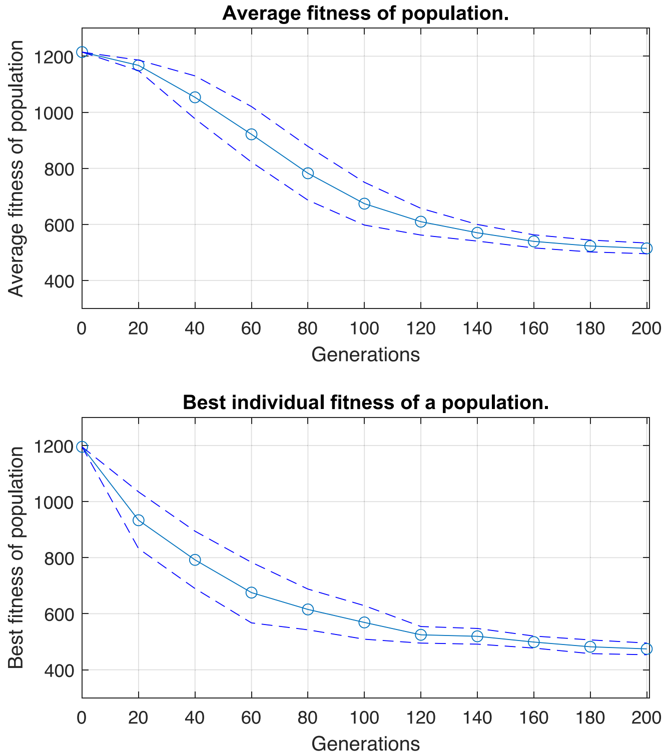


Fig. 8. Average and confidence levels (95%) results from 10 runs of GA with variable-length genome and parsimony pressure. (a) Evolution of average fitness of the population, (b) evolution of the best individual in the population.

range of application times and fewer types of NPs, it is fitter than treatment 4.

4.3. Mutation with addition, reduction and selection with parsimony pressure evolution

As bloat was observed in the results of the previous section, appropriate changes were made to the evolution methodology to control it. Namely, the mutation operator was updated to remove, with the same probability to adding, a type of NP from the genome. Moreover, parsimony pressure was applied to the selection operator, requiring an improvement of at least 10% for solutions of longer length genomes (Genetic Algorithm with Mutation Additions and Parsimony Pressure *GA-MAPP*).

After replicating the evolution methodology for 10 tests, the collective results are outlined in Fig. 8. The average fitness of the population evolving throughout the generations is depicted in Fig. 8(a) with the average values indicating a gain of 58% (initial average at 1215 and final at 462). Moreover, the best fitness throughout the evolution are displayed in Fig. 8(b), where a total advance of 60% in fitness is realised (initial best at 1195 and final at 474).

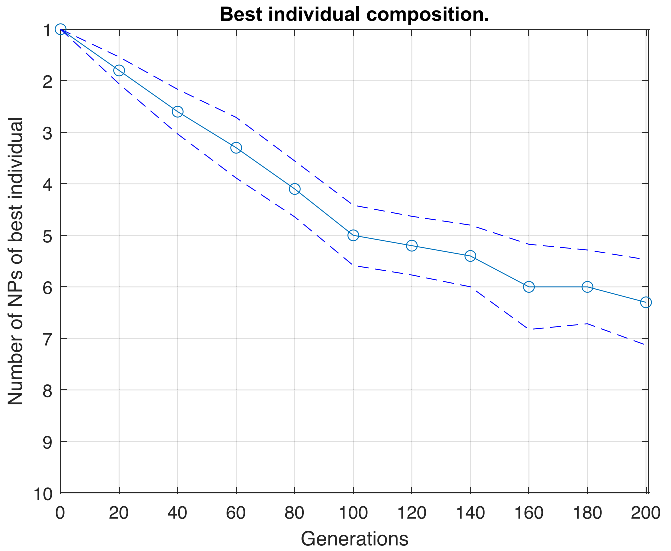


Fig. 9. Average and confidence levels (95%) results from 10 runs of GA with variable-length genome and parsimony pressure for the composition of the best solution.

Moreover, the composition of the best solutions found throughout the evolution generations are depicted in Fig. 9. Here, in contrast to the results from the previous methodology (Fig. 6), the increase in types of NPs is slower and seems to converge around 6 types of NPs, as a consequence of applying parsimony pressure.

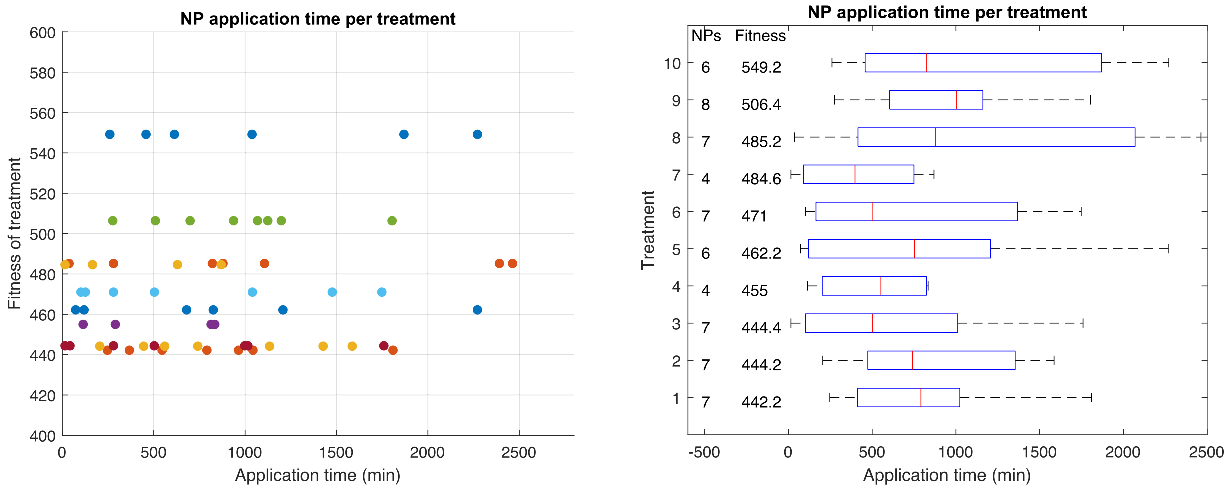
In Fig. 10(a) a scatter plot is given, comparing the application time of each type of NPs comprising a treatment and the combinatorial treatment fitness. Here the association between the distribution of the application times of NPs and the fitness of the treatment is not that apparent. In Fig. 10(b) a boxplot of the application times per treatment is presented. Correspondingly to previous results (presented in Fig. 7(b)), the majority of solutions with small range of application times and a median value closer to 0, perform better, as expected based on common sense. Nonetheless, based on the outcomes delivered from *GA-MAPP*, one can conclude that the improvements in fitness are not that significant to oppose possible unwanted side effects that may be caused by adding a higher amount of NPs, as described in the following section.

4.4. Comparison results

To better demonstrate the comparison of the results of the two evolutionary optimization methods, Fig. 11 is depicting the boxplot of fitnesses of the best individuals found by both methods at the final generation. Note here that the results from the *GA-CM* are not included in this figure, as they are significantly worse and

Table 2
Simulation parameters.

Parameter	NP#1	NP#2	NP#3	NP#4	NP#5	NP#6	NP#7	NP#8
attached worker migration bias	0.75710	0.71063	0.21146	0.35434	0.20843	0.65137	0.79332	0.98463
unattached worker migration bias	0.85409	0.94913	0.22137	0.27906	0.78920	0.34177	0.67675	0.19820
worker relative adhesion	5.22528	5.14334	8.47637	5.83786	5.72347	8.36366	0.83817	5.79598
worker relative repulsion	8.09247	3.53757	2.93546	2.80705	6.66245	5.40435	5.17048	5.04158
worker motility persistence time (min)	4.94834	4.00756	9.20594	0.77420	9.28788	1.68606	4.90029	9.60829
***GA-MA*application time (min)**	1134	154	289	533	826	307	193	389
Early application time (min)	150	152	154	156	158	160	162	164



(a) (b)

Fig. 10. (a) Scatter plot of application times for types of NPs and their collective fitness as a treatment. (b) Boxplot of application times per best treatment found on each optimization test. Additional information in the figure are the fitness and amount of types of NPs per treatment.

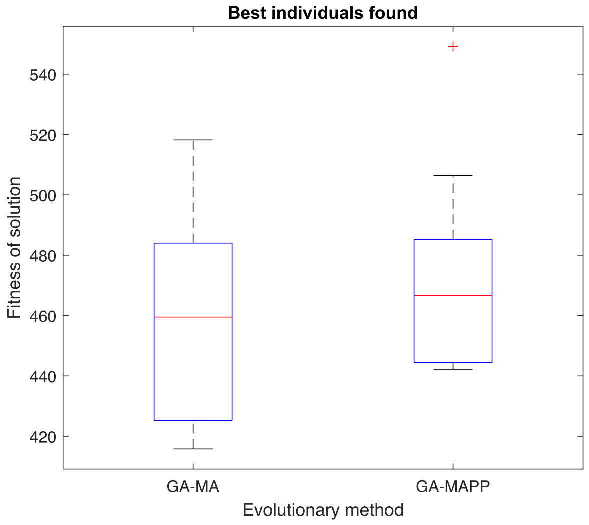


Fig. 11. Boxplot of fitness of best individuals found by two evolutionary methods (mutation operator adding one type of NPs *GA-MA* and mutation operator adding or removing one type of NPs with parsimony pressure *GA-MAPP*).

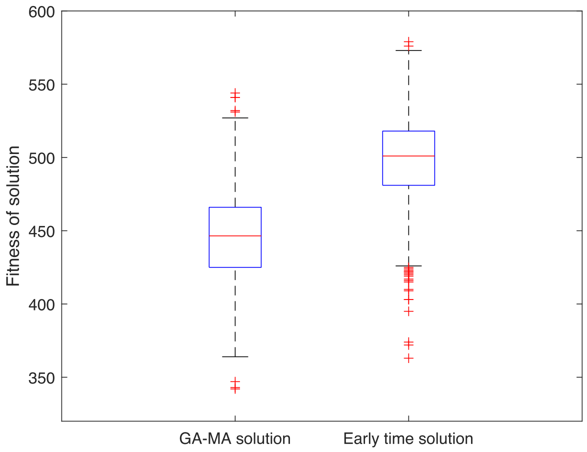


Fig. 12. Boxplot of fitness of 1000 simulations for solutions provided by evolutionary algorithm *GA-MA* and with altered early application times.

if it was included, the y-axis would leave the differences of the other two variations indistinct. The results are similarly distributed, and when the two distributions were compared under the two-sided Wilcoxon rank sum test, they were not proved to be significantly different (the null hypothesis that data in both distributions are samples from continuous distributions with equal medians is not rejected with $p = 0.3447$). Thus, despite the small difference of their sample median fitness values (460 for mutation operator adding on NP and 467 for parsimony pressure), the second method (namely *GA-MAPP*) is preferred, as it produces less complex treatments in terms of amounts of types of NPs.

Nonetheless, to investigate whether the diversity in application times found by the evolutionary methodologies is resulting in fitter solutions, one of the found solutions was tested by evaluating it on PhysiCell 1000 times. Then, the application times were arbitrarily altered into earlier time slots within a smaller range, as described in Table 2, whereas the rest of the parameters for the NPs were not changed. The results are presented in the boxplots of Fig. 12, where the solution found by the *GA-MA* is clearly outperforming the other one. The distribution of the *GA-MA* solution fitness has a median of 447 remaining cancer cell agents, while the early application time solution has a median of 501.

5. Conclusions

In order to match the complexity of a dynamic cancer tumor, a more elaborate drug delivery system is required to achieve a robust remedy. The possibility of releasing different kinds of NPs with the same therapeutic compound during different times was tested *in silico*. An evolutionary algorithm was implemented to optimize the design of the treatment (parameters of different types of NPs and release timetable), by using a metameric representation and appropriately modified mutation operator. In order to control bloat, namely the advance of the genome lengths to higher lengths without sufficient improvement in fitness, further alternations were applied to the evolutionary optimization methodology.

The results presented in this work, determine that the groups of NPs need to be released within a small range of time slots and fairly early in the simulation, to exhibit increased effectiveness. Moreover, by including parsimony pressure in the replacement operator, simpler treatments (in terms of amount of types of NPs) can be found, without worsening their relative fitness. The development of simpler treatments is important for keeping costs low during fabrication processes and alleviating the possibility of unpredicted and unwanted toxic behaviour of the NPs when combined *in vitro* and *in vivo*.

In order to further examine the relevance between application times and fitness, one of the solutions found by the evolutionary optimization methodology was compared with a solution where the application times were modified to earlier ones. The extensive evaluation of the two solutions within the simulator proved that the specific order of application time slots is highly important for the fitness of the treatment and can not be arbitrarily set to just earlier time slots. Note here that the rest of the parameters describing the behaviour of all NPs were the same.

A possible direction of future work would be to implement crossover operators that will be specialized for metameric representations [30]. Moreover, highly specialized [26] and innovative [25] evolutionary algorithms can be implemented to reach better results in a faster manner. Finally, due to the high computational cost of the simulator, a methodology of including machine learning techniques in the evolutionary optimization loop, like surrogate-assisted evolutionary algorithms [23], will be investigated to accelerate the extraction of results.

Funding information

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.

Declaration of Competing Interest

The authors report no conflict of interest.

References

[1] I. Dagogo-Jack, A.T. Shaw, Tumour heterogeneity and resistance to cancer therapies, Nat. Rev. Clin. Oncol. 15 (2018) 81.
[2] C.A. Stewart, C.M. Gay, Y. Xi, S. Sivajothi, V. Sivakamasundari, J. Fujimoto, M. Bolisetty, P.M. Hartsfield, V. Balasubramaniyan, M.D. Chalishazar, et al., Single-cell analyses reveal increased intratumoral heterogeneity after the onset of therapy resistance in small-cell lung cancer, Nat. Cancer 1 (2020) 423–436.
[3] K. Eales, K. Hollinshead, D. Tennant, Hypoxia and metabolic adaptation of cancer cells, Oncogenesis 5 (2016) e190.
[4] L. Mele, V. del Vecchio, D. Liccardo, C. Prisco, M. Schwerdtfeger, N. Robinson, V. Desiderio, V. Tirino, G. Papaccio, M. La Noce, The role of autophagy in resistance to targeted therapies, Cancer Treat. Rev. (2020) 102043.
[5] E.J. Huijbers, J.R. van Beijnum, V.L. Thijssen, S. Sabrkhany, P. Nowak-Sliwinska, A.W. Griffioen, Role of the tumor stroma in resistance to anti-angiogenic therapy, Drug Resist. Updates 25 (2016) 26–37.
[6] Y. Gao, Y. Shi, L. Wang, S. Kong, J. Du, G. Lin, Y. Feng, Advances in mathematical models of the active targeting of tumor cells by functional nanoparticles, Comput. Methods Programs Biomed. 184 (2020) 105106.
[7] K. Sztandera, M. Gorzkiewicz, B. Klajnert-Maculewicz, Gold nanoparticles in cancer treatment, Mol. Pharm. 16 (2018) 1–23.
[8] M. Borkowska, M. Siek, D.V. Kolygina, Y.I. Sobolev, S. Lach, S. Kumar, Y.-K. Cho, K. Kandere-Grzybowska, B.A. Grzybowski, Targeted crystallization of mixed-charge nanoparticles in lysosomes induces selective death of cancer cells, Nat. Nanotechnol. 15 (2020) 331–341.
[9] P. Gener, S. Montero, H. Xandri-Monje, Z.V. Díaz-Riascos, D. Rafael, F. Andrade, F. Martínez-Trucharte, P. González, J. Seras-Franzoso, A. Manzano, et al., Zileuton^{TM} loaded in polymer micelles effectively reduce breast cancer circulating tumor cells and intratumoral cancer stem cells, Nanomedicine 24 (2020) 102106.
[10] V.P. Podduturi, I.B. Magaña, D.P. O'Neal, P.A. Derosa, Simulation of transport and extravasation of nanoparticles in tumors which exhibit enhanced permeability and retention effect, Comput. Methods Programs Biomed. 112 (2013) 58–68.
[11] C.F. Rodrigues, T.A. Jacinto, A.F. Moreira, E.C. Costa, S.P. Miguel, I.J. Correia, Functionalization of AuMSS nanorods towards more effective cancer therapies, Nano Res. 12 (2019) 719–732.
[12] L.S. Jabr-Milane, L.E. van Vlerken, S. Yadav, M.M. Amiji, Multi-functional nanocarriers to overcome tumor drug resistance, Cancer Treat. Rev. 34 (2008) 592–602.
[13] J.H. Ryu, H. Koo, I.-C. Sun, S.H. Yuk, K. Choi, K. Kim, I.C. Kwon, Tumor-targeting multi-functional nanoparticles for theragnosis: new paradigm for cancer therapy, Adv. Drug Deliv. Rev. 64 (2012) 1447–1458.
[14] X. An, A. Zhu, H. Luo, H. Ke, H. Chen, Y. Zhao, Rational design of multi-stimuli-responsive nanoparticles for precise cancer therapy, Acs Nano 10 (2016) 5947–5958.
[15] I. Bozic, C.J. Wu, Delineating the evolutionary dynamics of cancer from theory to reality, Nat. Cancer 1 (2020) 580–588.
[16] E. Blanco, T. Sangai, A. Hsiao, S. Ferrati, L. Bai, X. Liu, F. Meric-Bernstam, M. Ferrari, Multistage delivery of chemotherapeutic nanoparticles for breast cancer treatment, Cancer Lett. 334 (2013) 245–252.
[17] J. Sun, C. Luo, Y. Wang, Z. He, The holistic 3m modality of drug delivery nanosystems for cancer therapy, Nanoscale 5 (2013) 845–859.
[18] S. Ruan, L. Zhang, J. Chen, T. Cao, Y. Yang, Y. Liu, Q. He, F. Gao, H. Gao, Targeting delivery and deep penetration using multistage nanoparticles for triple-negative breast cancer, RSC Adv. 5 (2015) 64303–64317.
[19] A. Ghaffarizadeh, R. Heiland, S.H. Friedman, S.M. Mumenthaler, P. Macklin, PhysiCell: An open source physics-based cell simulator for 3-d multicellular systems, PLoS Comput. Biol. 14 (2018) e1005991.
[20] P. Dogra, J.D. Butner, Y.-l. Chuang, S. Caserta, S. Goel, C.J. Brinker, V. Cristini, Z. Wang, Mathematical modeling in cancer nanomedicine: a review, Biomed. Microdevices 21 (2019) 40.
[21] N.R. Stillman, M. Kovacevic, I. Balaz, S. Hauert, In silico modelling of cancer nanomedicine, across scales and transport barriers, npj Comput. Mater. 6 (2020), doi:10.1038/s41524-020-00366-8.
[22] Y. Gao, Y. Shi, M. Fu, Y. Feng, G. Lin, D. Kong, B. Jiang, Simulation study of the effects of interstitial fluid pressure and blood flow velocity on transvascular transport of nanoparticles in tumor microenvironment, Comput. Methods Programs Biomed. (2020) 105493.
[23] R.J. Preen, L. Bull, A. Adamatzky, Towards an evolvable cancer treatment simulator, BioSystems 182 (2019) 1–7.
[24] J. Ozik, N. Collier, R. Heiland, G. An, P. Macklin, Learning-accelerated discovery of immune-tumour interactions, Mol. Syst. Des. Eng. (2019).
[25] M.-A. Tsompanas, L. Bull, A. Adamatzky, I. Balaz, Novelty search employed into the development of cancer treatment simulations, Inf. Med. Unlocked 19 (2020) 100347, doi:10.1016/j.imu.2020.100347.
[26] M.-A. Tsompanas, L. Bull, A. Adamatzky, I. Balaz, Haploid-diploid evolution: nature's memetic algorithm, arXiv preprint arXiv:1911.07302 (2019).
[27] M.-A. Tsompanas, L. Bull, A. Adamatzky, I. Balaz, Utilizing differential evolution into optimizing targeted cancer treatments, 2020a, arXiv:2003.11623.
[28] M.-A. Tsompanas, L. Bull, A. Adamatzky, I. Balaz, Evolving nano particle cancer treatments with multiple particle types, 2020b, arXiv:2011.04975.
[29] M.L. Ryerkerk, R.C. Averill, K. Deb, E.D. Goodman, Solving metameric variable-length optimization problems using genetic algorithms, Genet. Program. Evolvable Mach. 18 (2017) 247–277.
[30] M.L. Ryerkerk, R.C. Averill, K. Deb, E.D. Goodman, A survey of evolutionary algorithms using metameric representations, Genet. Program. Evolvable Mach. 20 (2019) 441–478.
[31] J.C. Mora, J.M.C. Barón, J.M.R. Santos, M.B. Payán, An evolutive algorithm for wind farm optimal design, Neurocomputing 70 (2007) 2651–2658.
[32] N. Weicker, G. Szabo, K. Weicker, P. Widmayer, Evolutionary multiobjective optimization for base station transmitter placement with frequency assignment, IEEE Trans. Evol. Comput. 7 (2003) 189–203.
[33] R. Poli, N.F. McPhee, Parsimony pressure made easy: solving the problem of bloat in GP, in: Theory and Principled Methods for the Design of Metaheuristics, Springer, 2014, pp. 181–204.
[34] K.O. Stanley, R. Miikkulainen, Evolving neural networks through augmenting topologies, Evol. Comput. 10 (2002) 99–127.