

# Optimising Cancer Chemotherapy Using Particle Swarm Optimisation and Genetic Algorithms

Andrei Petrovski<sup>†</sup>, Bhavani Sudha, John McCall

School of Computing  
The Robert Gordon University  
St. Andrew Street  
ABERDEEN, AB25 1HG, United Kingdom  
{ap, bs, jm}@comp.rgu.ac.uk

**Abstract.** Cancer chemotherapy is a complex treatment mode that requires balancing the benefits of treating tumours using anti-cancer drugs with the adverse toxic side-effects caused by these drugs. Some methods of computational optimisation, Genetic Algorithms in particular, have proven to be useful in helping to strike the right balance.

The purpose of this paper is to study how an alternative optimisation method - Particle Swarm Optimisation - can be used to facilitate finding optimal chemotherapeutic treatments, and to compare its performance with that of Genetic Algorithms.

## 1. Introduction

Many decision-making activities involve searching through a large space of possible solutions. In the chemotherapy problem we have studied, the size of the solution space increases exponentially with the number of decision variables, the values of which need to satisfy certain feasibility criteria.

The requirements imposed on decision variables often make the structure of a solution space quite intricate - regions of feasible solutions are scattered irregularly throughout the solution space, and only one of these regions contains the optimal solution. To find the optimal solution in such situations becomes a difficult task for conventional optimisation methods (gradient-based or simple heuristics). Similarly, the methods of mathematical programming cannot easily deal with multiplicity of feasible regions in the solution space.

It has been found [6], [7], [8], [10] that Genetic Algorithms show a good and robust performance on a class of non-linear, multi-constrained chemotherapy design problems. However, the field of evolutionary computation is growing, and alternative techniques of computational optimisation are being developed. One of such technique is the Particle Swarm Optimisation (PSO), introduced by Kennedy and Eberhart [3].

---

---

<sup>†</sup> corresponding author

The appeal of PSO is that it is also a population-based optimisation technique based on the 'social-psychological tendency of individual particles' within the swarm to 'emulate the success of other individuals' [3]. The search behaviour of a particle is influenced by the experience of its neighbours and represents a 'kind of symbiotic cooperative algorithm', aimed at efficient search through unpredictable solution spaces that have complex structures.

This property of PSO may make it particularly suitable to the optimisation problem of cancer chemotherapy, which exhibits such properties as multimodality and disjoint nature of feasible regions in the solution space. The purpose of this paper is to study the capabilities of PSO and to compare it with those of genetic algorithms.

In Section 2 we are going to explain the salient features of the chemotherapy optimisation problem. Section 3 describes the methodology of solving this optimization problem using two approaches – Genetic Algorithms and PSO. In Section 4 the details of experiments based on the identified approaches are given. Finally, Section 5 illustrates the experimental results and draws some conclusions, whereas in Section 6 we discuss the significance and our interpretation of the results obtained.

## **2. Problem Background**

Amongst the modalities of cancer treatment, chemotherapy is often considered as inherently the most complex [14]. As a consequence of this, it is extremely difficult to find effective chemotherapy treatments without a systematic approach. In order to realise such an approach, we need to take into account the medical aspects of cancer treatment.

### **2.1. Medical Aspects of Chemotherapy**

Drugs used in cancer chemotherapy all have narrow therapeutic indices. This means that the dose levels at which these drugs significantly affect a tumor are close to those levels at which unacceptable toxic side-effects occur. Therefore, more effective treatments result from balancing the beneficial and adverse effects of a combination of different drugs, administered at various dosages over a treatment period [7].

The beneficial effects of cancer chemotherapy correspond to treatment objectives which oncologists want to achieve by means of administering anti-cancer drugs. A cancer chemotherapy treatment may be either curative or palliative. Curative treatments attempt to eradicate the tumour; palliative treatments, on the other hand, are applied only when a tumour is deemed to be incurable with the objective to maintain a reasonable quality of life for as long as possible.

The adverse effects of cancer chemotherapy stem from the systemic nature of this treatment: drugs are delivered via the bloodstream and therefore affect

all body tissues. Since most anti-cancer drugs are highly toxic, they inevitably cause damage to sensitive tissues elsewhere in the body. In order to limit this damage, toxicity constraints need to be placed on the amount of drug applied at any time interval, on the cumulative drug dosage over the treatment period, and on the damage caused to various sensitive tissues [14]. In addition to toxicity constraints, the tumour size (i.e. the number of cancerous cells) must be maintained below a lethal level during the whole treatment period for obvious reasons.

The goal of cancer chemotherapy therefore is to achieve the beneficial effects of treatment objectives without violating any of the abovementioned constraints

## 2.2. Problem Formulation

In order to solve the optimisation problem of cancer chemotherapy, we need to find a set of treatment schedules, which satisfies toxicity and tumour size constraints yielding at the same time acceptable values of treatment objectives. This set will allow the oncologist to make a decision on which treatment schedule to use, given his/her preferences or certain priorities. In the remainder of this section we will define the decision vectors and the search space for the cancer chemotherapy optimisation problem, specify the constraints, and particularise the optimisation objectives.

Anti-cancer drugs are usually delivered according to a discrete dosage program in which there are  $n$  doses given at times  $t_1, t_2, \dots, t_n$  [5]. In the case of multi-drug chemotherapy, each dose is a cocktail of  $d$  drugs characterised by the concentration levels  $C_{ij}, i \in \overline{1, n}, j \in \overline{1, d}$  of anti-cancer drugs in the bloodplasma. Optimisation of chemotherapeutic treatment is achieved by modification of these variables. Therefore, the solution space  $\Omega$  of the chemotherapy optimisation problem is the set of control vectors  $\mathbf{c} = (C_{ij})$  representing the drug concentration profiles.

However, not all of these profiles will be feasible as chemotherapy treatment must be constrained in a number of ways. Although the constraint sets of chemotherapeutic treatment vary from drug to drug as well as with cancer type, they have the following general form.

1. Maximum instantaneous dose  $C_{\max}$  for each drug acting as a single agent:

$$g_1(\mathbf{c}) = \left\{ C_{\max j} - C_{ij} \geq 0 : \forall i \in \overline{1, n}, \forall j \in \overline{1, d} \right\} \quad (1)$$

2. Maximum cumulative  $C_{\text{cum}}$  dose for drug acting as a single agent:

$$g_2(\mathbf{c}) = \left\{ C_{\text{cum } j} - \sum_{i=1}^n C_{ij} \geq 0 : \forall j \in \overline{1, d} \right\} \quad (2)$$

3. Maximum permissible size  $N_{\max}$  of the tumour:

$$g_3(\mathbf{c}) = \{N_{\max} - N(t_i) \geq 0 : \forall i \in \overline{1, n}\} \quad (3)$$

4. Restriction on the toxic side-effects of multi-drug chemotherapy:

$$g_4(\mathbf{c}) = \left\{ C_{s\text{-eff } k} - \sum_{j=1}^d \eta_{kj} C_{ij} \geq 0 : \forall i \in \overline{1, n}, \forall k \in \overline{1, m} \right\} \quad (4)$$

The factors  $\eta_{kj}$  in the last constraint represent the risk of damaging the  $k^{\text{th}}$  organ or tissue (such as heart, bone marrow, lung etc.) by administering the  $j^{\text{th}}$  drug. Estimates of these factors for the drugs most commonly used in treatment of breast cancer, as well as the values of maximum instantaneous and cumulative doses, can be found in [1, 2].

Regarding the objectives of cancer chemotherapy, we focus our study on the primary objective of cancer treatment – tumour eradication. We define eradication to mean a reduction of the tumour from an initial size of around  $10^9$  cells (minimum detectable tumour size) to below  $10^3$  cells.

In order to simulate the response of a tumour to chemotherapy, a number of mathematical models can be used [5]. The most popular is the Gompertz growth model with a linear cell-loss effect [14]:

$$\frac{dN}{dt} = N(t) \cdot \left[ \lambda \ln\left(\frac{\Theta}{N(t)}\right) - \sum_{j=1}^d \kappa_j \sum_{i=1}^n C_{ij} \{H(t-t_i) - H(t-t_{i+1})\} \right] \quad (5)$$

where  $N(t)$  represents the number of tumour cells at time  $t$ ;  $\lambda, \Theta$  are the parameters of tumour growth,  $H(t)$  is the Heaviside step function;  $\kappa_j$  are the quantities representing the efficacy of anti-cancer drugs, and  $C_{ij}$  denote the concentration levels of these drugs. One advantage of the Gompertz model from the computational optimisation point of view is that the equation (5) yields an analytical solution after the substitution  $u(t) = \ln(\Theta/N(t))$  [5]. Since  $u(t)$  increases when  $N(t)$  decreases, the primary optimisation objective of tumour eradication can be formulated as follows [8]:

$$\underset{\mathbf{c}}{\text{minimise}} \quad F(\mathbf{c}) = \sum_{i=1}^n N(t_i) \quad (6)$$

subject to the state equation (5) and the constraints (1)-(4).

### 3. Methodology

In this section we are going to explain how the optimisation problem of cancer chemotherapy can be solved by two computational optimisation techniques – GA and PSO.

#### 3.1. Genetic Algorithms

The search process aims at finding chemotherapy schedules that satisfy treatment constraints and optimize the optimisation objective (6).

The search for such treatment schedules may be accomplished using the GA approach. Multi-drug chemotherapy schedules, represented by decision vectors  $\mathbf{c} = (C_{ij}), i \in \overline{1, n}, j \in \overline{1, d}$ , are encoded as binary strings. The representation space  $\mathbf{I}$  (a discretized version of  $\Omega$ ) can then be expressed as a Cartesian product

$$\mathbf{I} = A_1^1 \times A_1^2 \times \dots \times A_1^d \times A_2^1 \times A_2^2 \times \dots \times A_2^d \times \dots \times A_n^1 \times A_n^2 \times \dots \times A_n^d \quad (7)$$

of allele sets  $A_i^j$ . Each allele set uses a 4-bit representation scheme

$$A_i^j = \{a_1 a_2 a_3 a_4 : a_k \in \{0,1\} \forall k \in \overline{1,4}\} \quad (8)$$

so that each concentration level  $C_{ij}$  takes an integer value in the range of 0 to 15 concentration units [6], [7]. In general, with  $n$  treatment intervals and up to  $2^4$  concentration levels for  $d$  drugs, there are up to  $2^{npd}$  individual elements. Henceforth we assume that  $n=10$  and that the number of available drugs is restricted to ten [8]. The values  $n=10$  and  $d=10$  result in the individual (search) space of power  $|\mathbf{I}| = 2^{400}$  individuals, referred to as chromosomes.

Thus, a chromosome  $x \in \mathbf{I}$  can be expressed as

$$x = \{a_1 a_2 a_3 \dots a_{4nd} : a_k \in \{0,1\} \forall k \in \overline{1,4nd}\} \quad (9)$$

and the mapping function  $m: \mathbf{I} \rightarrow \mathbf{C}$  between the individual  $\mathbf{I}$  and the decision vector  $\mathbf{C}$  spaces can be defined as

$$C_{ij} = \Delta C_j \sum_{k=1}^4 2^{4-k} a_{4d(i-1)+4(j-1)+k}, \quad \forall i \in \overline{1, n}, j \in \overline{1, d} \quad (10)$$

where  $\Delta C_j$  represents the concentration unit for drug  $j$ . This function symbolizes the decoding algorithm to derive the decision vector  $\mathbf{c} = m(x)$  from a chromosome  $x$ . Applying the evaluation function  $F$  to  $\mathbf{c}$  yields the value of the optimisation objective.

### 3.2. Particle Swarm Optimisation

The PSO algorithm is initialised with a population of random candidate solutions, conceptualised as particles. These particles are flown through the hyperspace  $\Omega$  of solutions to the chemotherapy optimisation problem described in the previous section. The position of each particle  $\bar{c}_i^{k+1}$  at iteration  $k + 1$  corresponds to a treatment regimen of anti-cancer drugs and is determined by the following formula:

$$\bar{c}_i^{k+1} = \bar{c}_i^k + \bar{v}_i^k \quad (11)$$

where is  $\bar{v}_i^k$  a randomised velocity vector assigned to each particle in a swarm. The velocity vector drives the optimisation process and reflects the 'socially exchanged' information.

There exist different algorithms that regulate how this 'social' information is exchanged [3]. In the first algorithms – *individual best* – each particle compares its current position in the solution space  $\Omega$  to its own best position found so far; no information from other particles is used. In the second algorithm – *local best* – particles are influenced by the best position within their neighbourhood, as well as their own past experience. In the third algorithm – *global best* – the 'social' knowledge used to drive the movements of particles includes the position of the best particle from the entire swarm.

Therefore, each particle in the swarm is attracted towards the locations representing best chemotherapeutic treatments found by the particle itself, its neighbours, and/or the entire population. This is achieved by defining the velocity vector in (11) for each particle as:

$$\bar{v}_i^k = w \cdot \bar{v}_i^{k-1} + b_1 \cdot r_1 \cdot (\bar{c}_i^* - \bar{c}_i^{k-1}) + b_2 \cdot r_2 \cdot (\bar{c}_i^{**} - \bar{c}_i^{k-1}) \quad (12)$$

where:

$w$  is the inertia coefficient;

$b_1$  and  $b_2$  are empirical coefficients used to improve PSO performance;

$r_1$  and  $r_2$  are random numbers in the range  $[0,1]$ ;

$\bar{c}_i^*$  and  $\bar{c}_i^{**}$  are the best locations in  $\Omega$  found by the particle  $i$  and the entire population respectively;

$\bar{v}_i^{k-1}$  is the value of particle  $i$  velocity at previous iteration of the algorithm; the values  $\bar{v}_i^0$  are initialised at random.

The PSO algorithm works by finding a new position for each particle using (11) and (12), evaluating them and updating the personal and global best values.

## 4. Experiments

In our study we compared three algorithms - Genetic Algorithms, global best PSO, and local best PSO. The comparison has been done on the problem of multi-drug cancer chemotherapy optimisation addressed in [6], [7], [8]. The optimisation objective is to minimise the overall tumour burden  $F(\mathbf{c})$  defined by (6) with the aim to eradicate the tumour.

In our attempt to solve this problem, the following settings of algorithms' properties and parameters have been chosen.

### 4.1. Genetic Algorithms

The initial GA population of 50 individuals is chosen at random in accordance with the representation scheme (8)-(9). The selection procedure is based in the roulette-wheel selection, augmented by a linear fitness normalization technique [8] and an elitist strategy that reserves two best chromosomes in the population. Recombination is implemented as a 2-point crossover followed by a uniform mutation applied with the probabilities  $p_c = 0.5$  and  $p_m = 0.2$  respectively. (These settings are the same as in our previous studies [6], [7], [8].)

### 4.2. Particle Swarm Optimisation

Similar to the GA population, initial positions of 50 PSO particles are generated at random. Each particle in the swarm is assigned a random velocity value from the range  $[0,2]$ , i.e.  $\bar{v}_i^0 \in [0, 2]$ ,  $i \in [1, 50]$ ; this value changes at each iteration of the algorithm according to (12) with the following settings of parameters as recommended in [11]:

- $\omega$  is assigned a randomly generated value from the range  $[0.5,1]$ ;
- $b_1 = b_2 = 4$ ;
- $r_1, r_2$  are randomly generated from the range  $[0,1]$ ;
- the velocity bound values  $|v_{\max}|$  are set to 1 to help keep the swarm under control [9].

For the local best PSO algorithm, the neighbourhood size was chosen to be 20% of the population, i.e. each neighbourhood contains 10 particles and is formed on the basis of the numerical indexes assigned to these particles.

The programs implementing both GA and PSO algorithms are written in Java; these programs run until a predefined termination criterion is satisfied. The termination criterion was chosen to be 25,000 fitness function evaluations, because it has been empirically found that this number of evaluations

guarantees finding a feasible solution for all trial runs of at least one algorithm, which happened to be the global best PSO.

Because of the randomised nature of algorithms under investigation, the programs implementing GA, local and global best PSO were run 30 times each. In order to sharpen the comparison of performance, the same set of 30 random starting populations was used for each of the algorithms tested. This ensures that differences in performance between algorithms cannot be ascribed to a relatively poor set of random starts. In addition, repeated runs allowed us to gather empirical data, statistical analysis of which can lead to reliable results. These results are presented in the next section.

## 5. Results

During each trial run of the programs implementing GA, local and global PSO, the following outcomes were recorded:

- the number of algorithms' iterations (referred to as generations) required to find at least one feasible (i.e. satisfying all the constraints (1)-(4)) solution;
- the maximum value of the fitness function found at each iteration of the algorithms;
- the best solution found at the end of a trial run.

Figure 1 presents the comparative results based on the first measure - the mean number of generations required to find a feasible solution. The data are represented in the format adopted in [4].

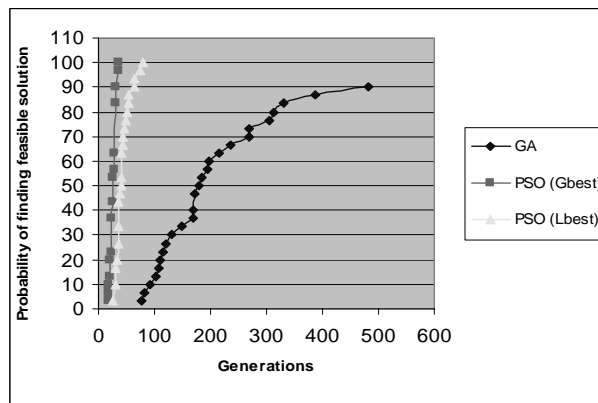


Fig. 1. Number of generations needed to find a feasible solution

From the above graph it is clear that the PSO algorithms find the feasible region in the solution space of chemotherapeutic treatments faster than Genetic Algorithms. This conclusion is confirmed by the statistical analysis based on interquartile ranges of experimental samples and on t-test comparison of means, summarised in Table 1.

**Table 1.** t-test comparison of the algorithms' speeds

Pair	Difference in Means	Std. Error Mean	t-test	p-value
GA vs. global PSO	170.9259	18.87354	9.056	.000
GA vs. local PSO	152.7778	19.00002	8.041	.000
Global vs. local PSO	-17.200	2.6920	-6.389	.000

As can be seen from the last column of Table 1, all p-values are  $\ll 0.05$ ; this indicates that the difference in algorithms' performance originates from their respective effectiveness rather than from random noise.

The second outcome of our comparison is that in addition to finding a feasible solution faster, the PSO algorithms yield the overall best solution (found at the end of each trial run) of the same quality as that of Genetic Algorithms as can be seen from Table 2.

**Table 2.** t-test comparison of best solutions found by each algorithm

Pair	Difference in Means	Std. Error Mean	t-test	p-value
GA vs. global PSO	-1.4063	.76898	-1.829	.078
GA vs. local PSO	-1.4040	.76872	-1.826	.078
Global vs. local PSO	.0023	.00446	.523	.605

This is an important observation implying that the PSO algorithms do not achieve efficiency at the expense of effectiveness.

## 6. Discussions

Our previous work [6], [7], [8] has shown that Genetic Algorithms can be useful in solving the multi-constrained and multi-dimensional problem of cancer chemotherapy optimisation. The present study has demonstrated that an alternative method - Particle Swarm Optimisation - is able to achieve the same optimisation objective in a new and faster way.

The ability of the PSO algorithms to explore the solution space faster than GA has been reported in [9] and [11]. Our experimental results support these findings and show that the PSO algorithms, the global best PSO in particular, optimise cancer chemotherapy treatments in a more robust manner - all trial runs of the PSO programs led to finding a feasible solution. On the contrary, some GA runs did not result in finding a feasible region in the solution space having evaluated 25,000 treatments. We hypothesize that the nature of the solution space is such that optimal treatments lie on the boundaries of feasible regions. Recombination operators of Genetic Algorithms may cause these boundaries to be crossed, leading to infeasible solutions. The PSO algorithms, on the other hand, tend to keep particles within feasible regions by pulling them toward remembered locations in the solution space that

proved their trustworthiness. In this respect, a PSO algorithm can rely on its memory - the advantage that Genetic Algorithms do not have. Presumably, historical information on the best solutions found by each particle and the population on the whole is a valuable asset in the context of cancer chemotherapy optimisation, where multiple constraints and a very large solution space lead to a disjoint and sparse nature of the feasible region.

Although there is only a soft real-time constraint on finding a good treatment schedule, the cost of evaluating a chemotherapeutic treatment is likely to increase with the introduction of new anti-cancer drugs and imposing patient-specific constraints. Therefore, finding a solution to the problem of cancer chemotherapy optimisation faster without compromising its quality is an important and useful goal for oncologists. The results of this study show that PSO algorithms can be a viable, and even better, alternative to Genetic Algorithms in achieving this goal.

## References

1. Cassidy, J., McLeod, H.: Is it possible to design a logical development plan for an anti-cancer drug. *Pharmaceutical Medicine*, (1995), **9**, 95-103.
2. Dearnaley, D., et al.: *Handbook of adult cancer chemotherapy schedules*. The Medicine Group (Education) Ltd., Oxfordshire, (1995).
3. Eberhart, R.: *Computational Intelligence PC Tools*, Academic Press Professionals (APP), (1996) 185-196.
4. Hoos, H., Stutzle, T.: *Local Search Algorithms for SAT: An Empirical Evaluation*. *J. Automated Reasoning*, special Issue "SAT 2000", 1999.
5. Martin, R., Teo, K.: *Optimal Control of Drug Administration in Cancer Chemotherapy*. World Scientific, Singapore New Jersey London Hong Kong (1994).
6. McCall, J., Petrovski, A.: *A Decision Support System for Cancer Chemotherapy Using Genetic Algorithms*. *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*, Vol. 1. IOS Press (1999) 65-70.
7. Petrovski, A., McCall, J. A. W. *Multi-objective optimisation of cancer chemotherapy using evolutionary algorithms*. *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimisation*, Zurich, Switzerland (2001).
8. Petrovski, A.: *An Application of Genetic Algorithms to Chemotherapy Treatment*. PhD thesis, The Robert Gordon University, Aberdeen, U.K., (1999).
9. Robinson, J., Sinton, S., Rahmat-Samii, Y.: *Particle Swarm, Genetic Algorithm, and their Hybrids: Optimisation of a Profiled Corrugated Horn Antenna*. *IEEE International Symposium on Antennas & Propagation*. San Antonio, Texas, (2002).
10. Kay Chen Tan, Khor, E. F., Cai, J., Heng, C. M., Lee, T. H.: *Automating the drug scheduling of cancer chemotherapy via evolutionary computation*. *Artificial Intelligence in Medicine* 25(2): 169-185 (2002).
11. Trelea, I.: *The particle swarm optimization: convergence analysis and parameter selection*. *Information Processing Letters* (2003), **85**, 317-25.
12. Ujjin, S., Bentley, P.: *Particle Swarm Optimization Recommender System*. In *Proceedings of the IEEE Swarm Intelligence Symposium*, Indianapolis, 2003.
13. Venter, G., Haftka, R., Sobieszczanski-Sobieski, J.: *Robust Design Using Particle Swarm and Genetic Algorithm Optimisation*. *5<sup>th</sup> World Congress of Structural and Multidisciplinary Optimization*, Lido di Jesolo, Italy, May 19-23, 2003.
14. Wheldon, T.: *Mathematical models in cancer research*. Adam Hilger, Bristol Philadelphia (1988).