

Application of evolutionary algorithms to optimise one- and two-dimensional gradient chromatographic separations

Huygens, Bram; Efthymiadis, Kyriakos; Nowe, Ann; Desmet, Gert

Published in:
Journal of Chromatography. A

DOI:
[10.1016/j.chroma.2020.461435](https://doi.org/10.1016/j.chroma.2020.461435)
[10.1016/j.chroma.2020.461435](https://doi.org/10.1016/j.chroma.2020.461435)

Publication date:
2020

License:
CC BY-NC-ND

Document Version:
Accepted author manuscript

[Link to publication](#)

Citation for published version (APA):
Huygens, B., Efthymiadis, K., Nowe, A., & Desmet, G. (2020). Application of evolutionary algorithms to optimise one- and two-dimensional gradient chromatographic separations. *Journal of Chromatography. A*, 1628, [461435]. <https://doi.org/10.1016/j.chroma.2020.461435>, <https://doi.org/10.1016/j.chroma.2020.461435>

Copyright

No part of this publication may be reproduced or transmitted in any form, without the prior written permission of the author(s) or other rights holders to whom publication rights have been transferred, unless permitted by a license attached to the publication (a Creative Commons license or other), or unless exceptions to copyright law apply.

Take down policy

If you believe that this document infringes your copyright or other rights, please contact openaccess@vub.be, with details of the nature of the infringement. We will investigate the claim and if justified, we will take the appropriate steps.

Application of evolutionary algorithms to optimise one- and two-dimensional gradient chromatographic separations

Bram Huygens⁽¹⁾, Kyriakos Efthymiadis⁽²⁾, Ann Nowé⁽²⁾, Gert Desmet^(1,*)

⁽¹⁾ *Vrije Universiteit Brussel, Department of Chemical Engineering, Pleinlaan 2, 1050 Brussels, Belgium*

⁽²⁾ *Vrije Universiteit Brussel, Artificial Intelligence Lab, Pleinlaan 2, 1050 Brussels, Belgium*

^(*) Corresponding author. E-mail address: gedesmet@vub.ac.be

Abstract

We report on the performance of three classes of evolutionary algorithms (genetic algorithms (GA), evolution strategies (ES) and covariance matrix adaptation evolution strategy (CMA-ES)) as a means to enhance searches in the method development spaces of 1D- and 2D-chromatography. After optimisation of the design parameters of the different algorithms, they were benchmarked against the performance of a plain grid search. It was found that all three classes significantly outperform the plain grid search, especially in terms of the number of search runs needed to achieve a given separation quality. As soon as more than 100 search runs are needed, the ES algorithm clearly outperforms the GA and CMA-ES algorithms, with the latter performing very well for short searches (<50 search runs) but being susceptible to convergence to local optima for longer searches. It was also found that the performance of the ES and GA algorithms, as well as the grid search, follow a hyperbolic law in the large search run number limit, such that the convergence rate parameter of this hyperbolic function can be used to quantify the difference in required number of search runs for these algorithms. In agreement with one's physical expectations, it was also found that the general advantage of the GA and ES algorithms over the grid search, as well as their mutual performance differences, grow with increasing difficulty of the separation problem.

Keywords: evolutionary algorithms; gradient elution; liquid chromatography; method development; multi-dimensional chromatography

1. Introduction

The evolution in column technology and instrumentation in liquid chromatography is such that ever more complex samples can be resolved, to meet the ever increasing demands in biomarker discovery, environmental and food analysis, pharmaceutical research and other life science applications. To resolve these samples, analysts have a vast toolbox at their disposal: the choice of the most selective mobile and stationary phase, the use of mobile phase additives, the optimisation of pH, temperature and gradient parameters, *et cetera*. Furthermore, instrumentation for multi-dimensional chromatography has become commercially available in recent years [1-2], thus allowing to exploit the fact that the peak capacity in multi-LC is, save effects of undersampling and non-orthogonality, equal to the product of the peak capacities of the individual dimensions [3].

Because of this wealth of possibilities, the analyst is faced with a daunting optimisation problem during method development. To alleviate this workload, several software packages have been developed [4-5]. However, these are typically applicable to optimisation problems with a small number of parameters, whereas in some cases, the number of parameters can be ten or more, with the case of multi-dimensional chromatography [6-9] being a notable example.

In the present contribution, we report on our study of the applicability of optimisation methods developed in the field of artificial intelligence – evolutionary algorithms, to be specific – to enhance method development. An evolutionary algorithm is an optimisation method which mimics biological evolution by iteratively applying selection and reproduction to a population of candidate solutions,

referred to as ‘individuals’ [10-11]. In the selection step, the fitness of the individuals is evaluated based on an objective function, allowing to separate the better from the worse. In the reproduction step, the properties of the fittest individuals are recombined and mutated, thus bringing forth new individuals. After a number of iterations, referred to as ‘generations’, the population of candidate solutions converges to an optimum of the objective function.

A major advantage of evolutionary algorithms is that they are better at coping with large numbers of parameters than most classical optimisation methods are [10-11]. Furthermore, they do not require any assumptions on the behaviour of the objective function (such as continuity, differentiability, *et cetera*) and their populous nature limits their sensitivity to local optima, which are a considerable issue in multi-parameter optimisation. These advantages have sparked interest in their applicability in the field of chromatography. They were introduced to the field in 1993 by Lopes Marques *et al.* [12-13] to fit a nine-parameter retention model to experimental data. Indeed, because of the complexity of fitting nine parameters, they found evolutionary algorithms to be more reliable than classical methods. In the early 2000’s, their range of application was extended from retention modelling to method development by Nikitas *et al.* [14-16]. They also compared the performance of several evolutionary algorithms with so-called Gaussian mutation (see Sections 2.6-2.7), which they considered to be an improvement upon other mutation methods. Since then, the number of applications of evolutionary algorithms to both retention modelling and method development has been increasing [17-24]. Within the broader scope of chemometrics, applications of evolutionary algorithms include molecular design, molecular modelling and analysis of spectroscopic data [25]. For example, in the latter case, they were found to be useful in feature selection in infrared spectra, reducing them to the most relevant subset of wavelengths.

In the context of method development, a chromatographic response function (CRF), evaluating the separation quality and separation time, can serve as the objective function of the optimisation problem. Numerous definitions of such a criterion can be found in literature, both for 1D and 2D chromatography [26-27].

In the present study, three classes of evolutionary algorithms, differing from one another in terms of recombination and mutation methodology (see Sections 2.5-2.7), were implemented and evaluated: genetic algorithms (GA), non-adaptive evolution strategies (ES) and a covariance matrix adaptation evolution strategy (CMA-ES). Firstly, they were applied to the optimisation of gradient parameters in 1D chromatography, with a single linear gradient (see Section 3.1). Secondly, this application was extended to 2D chromatography, with a linear gradient in both the first and second dimension (see Section 3.2). Logically, the application to 1D chromatography cannot fully reveal the advantages of the evolutionary algorithms, as only two parameters are optimised (*i.e.*, gradient offset and gradient slope). Nevertheless, this case is extensively discussed, since it allows to provide visual insight into the performance of the evolutionary algorithms, via heatmaps of the CRF as a function of the two parameters.

All experiments were performed *in silico*, based on simulated chromatographic separations of samples with randomly generated components. This allowed to repeat the experiments for a statistically relevant number of samples within a reasonable time frame, and to determine the optimal solution

for a sample by a brute force grid search (*i.e.*, by running the simulation for numerous sets of gradient parameters, applying a high-resolution grid to the optimisation space). Thus, this methodology facilitated a quantitative evaluation of the evolutionary algorithms' performance. The chromatography parameters were kept as simple as possible (Gaussian peaks, equal initial concentration of analytes, perfect orthogonality between 1D and 2D separation) in order to have the cleanest possible view on the possibilities of the different tested algorithms, devoid of any effects originating from the assumptions made on the non-ideal chromatographic behaviour. The performance of the algorithms was compared to the performance of a simplistic grid search subjected to the same quantitative evaluation as the evolutionary algorithms.

Concerning their applicability, it should be noted the evolutionary algorithms could be applied to either guide experimental, search-based method development as well as to speed up computational, model-based method development. The former is limited to simple separation problems, where suitable separation conditions can be found within a reasonable number of search runs (*e.g.* 100) and where an experimental search is preferred because, for example, retention modelling of the individual components is deemed too time consuming or too error-prone. The latter relies on accurate retention modelling of the individual components [28], after which search runs are performed *in silico* such that the affordable number of search runs can run in the millions or billions, as is needed to solve complex separation problems.

2. Computational methodology

The performance of evolutionary algorithms in the context of gradient optimisation was evaluated by meta-experiments, consisting of a large number ($\#S$) of sub-experiments ($\#S = 10,000$ in the case of 1D chromatography, $\#S = 1,000$ in the case of 2D chromatography). In each sub-experiment, an algorithm was applied to optimise the gradient parameters for a sample with randomly generated composition. To do so, the algorithm was allowed to perform a given number of search runs (each being a simulated chromatographic separation), after which it had to propose a final solution, which was evaluated by a chromatographic response function (CRF). After performing all sub-experiments, each meta-experiment was concluded by calculating the average of the achieved CRF values, a measure of the algorithms' performance which is referred to further on as its 'efficacy':

$$efficacy = \frac{1}{\#S} \sum_{i=1}^{\#S} CRF_i \quad (1)$$

Additionally, the standard errors on these efficacy values – originating from the variance between samples – were calculated based on the square root law.

Each part of the sub-experiments was executed by a MATLAB® script or function: the random generation of components, the simulated chromatographic separations, the CRF evaluation and the algorithms themselves. The corresponding MATLAB® code was developed in-house, and its functionality is described part-by-part in the following sections.

2.1 Sample composition and retention properties

Artificial samples, consisting of a given number of components (s) with randomly generated retention properties, were generated *in silico*. These retention properties were based on the Neue-Kuß model, in which components are characterised by three retention parameters (k_W, S_1, S_2) and their retention factor (k) is expressed as a function of the volume fraction of organic modifier, commonly referred to as the mobile phase composition (φ) [29]:

$$k(\varphi) = k_W \cdot (1 + S_2\varphi)^2 \cdot \exp\left(-\frac{S_1\varphi}{1+S_2\varphi}\right) \quad (2)$$

For realistic ranges of the retention parameters, we adapted data from [29], where the Neue-Kuß model was used to fit retention time data for a number of small organic molecules. Although a straightforward approach would be to pick the retention parameters of a component independently from uniform distributions of k_W, S_1 and S_2 , this was only done for the latter. Instead, k_W was picked from a log-uniform distribution, because of its tendency to vary over multiple orders of magnitude. Furthermore, some calculations revealed that picking S_1 independently from the other two could result in components with unrealistic retention properties. Therefore, the extrapolated retention factor in pure organic modifier (k_M) was picked from a log-uniform distribution, after which S_1 was inferred from its relation to the other retention parameters. For this purpose, the following relation can be derived from Eq. 2 by setting the mobile phase composition equal to one, and subsequently isolating S_1 :

$$S_1 = (1 + S_2) \cdot \ln\left(\frac{k_W}{k_M} \cdot (1 + S_2)^2\right) \quad (3)$$

Thus, we arrived at realistic ranges for $\ln k_W$ (from 3.27 to 11.79), $\ln k_M$ (from -2.38 to -1.03) and S_2 (from -0.24 to 2.51). In the case of 2D chromatography, components were characterised by six retention parameters, three for each dimension. For the sake of simplicity, the two subsets of retention parameters were picked independently from one another. This implies that the two dimensions were assumed to be completely orthogonal. Although this limits the practical relevance of the results, we believe this approach provides the best starting point to gain a general insight in the possibilities of the tested algorithms, devoid of any interferences arising from the selected mechanism coupling the retention in the two dimensions.

2.2 Simulation of chromatographic separations

For each sample, a number of reversed-phase chromatographic separations with gradient elution was simulated by calculating the retention time (t_R) and peak spreading (σ_t) for each component. In the case of 1D chromatography, a linear gradient was employed, characterised by an offset (φ_0) and a slope (β) [30]:

$$\varphi(t) = \varphi_0 + \beta t \quad (4)$$

In the case of 2D chromatography, a linear gradient was employed in both the first and second dimension, characterised by a set of four gradient parameters ($\varphi_{0,1}, \beta_1, \varphi_{0,2}, \beta_2$). Within one chromatographic separation, the second dimension gradient was, for the sake of simplicity, repeated identically for each fraction, thus following the full-in-fraction method [31-32].

The retention times were calculated by substituting Eqs. (2) and (4) in the fundamental equation of gradient elution, taking into account the void time (t_0) [30]:

$$t_0 = \int_0^{t_R-t_0} \frac{dt}{k(\varphi(t))} \quad (5)$$

The calculation of peak spreading was based on the established models of peak broadening and peak compression, taking into account the column efficiency (N) [33]:

$$\sigma_t = \frac{\sqrt{1+p+\frac{p^2}{3}} (1+k(t=t_R)) t_0}{1+p \sqrt{N}} \quad (6)$$

$$\text{where } p = \beta t_0 S_1 \frac{k(t=0)}{1+k(t=0)}$$

Again for the sake of simplicity, peaks were assumed to be Gaussian and of equal height, thus neglecting the complications of asymmetry and shoulders. Examples of the idealised chromatograms are shown in Fig. 1. In the case of 2D chromatography, the same equations were applied in both the first and second dimension, thus calculating the four descriptors of each peak ($t_{R,1}$, $\sigma_{t,1}$, $t_{R,2}$, $\sigma_{t,2}$). The modulation time was assumed to be sufficiently small to not disturb the shape of the peaks.

Logically, the difficulty of method development is influenced by the sample complexity and the column efficiency. Therefore, several degrees of difficulty were considered throughout the meta-experiments, as given in Table 1. Since the void time does not influence the difficulty of method development, this can be considered as a dummy variable within the meta-experiments.

2.3 Chromatographic response function (CRF)

There are two classes of criteria which are typically incorporated in the definition of a CRF: criteria regarding the separation quality (resolution, peak-to-valley-ratio, *et cetera*) and criteria regarding the separation time [26-27]. Incorporating both, the CRF was defined as the sum of peak purities (P) – which is defined as the fraction of a peak that does not overlap with other peaks – counting only the peaks which eluted within a given time window (Δt):

$$CRF_{1D} = \sum_{i=1}^S (\mathbb{1}_{t_{R,i} < t_0 + \Delta t} \cdot P^i) \quad (7)$$

where $\Delta t = 20t_0$ in all considered cases

In the case of 2D chromatography, the CRF was defined analogously, with a given time window in both the first and second dimension:

$$CRF_{2D} = \sum_{i=1}^S (\mathbb{1}_{t_{R,1,i} < t_{0,1} + \Delta t_1} \cdot \mathbb{1}_{t_{R,2,i} < t_{0,2} + \Delta t_2} \cdot P^i) \quad (8)$$

where $\Delta t_1 = 20t_{0,1}$ and $\Delta t_2 = 20t_{0,2}$

As the peak purity is a value between zero and one, the CRF value is always smaller than or equal to the number of components. For example, the chromatogram shown in Fig. 1a has a CRF value of 18.1 out of 20.0, due to the minor overlap between peaks three and four and the major overlap between peaks five and six, and the chromatogram shown in Fig. 1b has a CRF value of 140.0 out of 150.0, due to various peak overlaps and co-elutions.

2.4 Evolutionary algorithms

The overall approach to method development followed by evolutionary algorithms is summarised in Fig. 2. Firstly, the population is initialised via a number of search runs with randomly chosen gradient parameters. Subsequently, the evolutionary algorithm cycles through selection and reproduction, performing additional search runs in each generation. Lastly, after having performed a given number of search runs, the evolutionary algorithm proposes the set of gradient parameters which achieved the highest CRF value as a solution to the optimisation problem.

In the selection step, the fitness of the individuals is evaluated based on the CRF. Several selection methods were implemented and evaluated: ‘elitist’, ‘probabilistic’ and ‘weighted’ [34-35]. In the elitist method, a limited number of individuals are chosen for reproduction, discarding individuals with low CRF values. Alternatively, in the probabilistic and weighted methods, each individual is given either some probability to reproduce or some weight in the reproduction method. The given probability or weight is proportional to the difference between its CRF value and the lowest CRF value within the population (this subtraction is done to avoid being too ‘indulgent’ towards individuals with low CRF values.)

In the reproduction step, new individuals are generated via recombination and mutation. These processes vary from one class of evolutionary algorithms to another, as described in Sections 2.5-2.7. A reoccurring design choice, however, is whether the old individuals should either be completely replaced by their offspring, or whether a selection of them should be conserved while adding their offspring to the population. Both methods, respectively referred to as ‘comma’ and ‘plus’ [34-35], were implemented and evaluated.

Another reoccurring design choice is which values should be allowed for the gradient parameters. As method development is typically a constrained optimisation problem, lower and upper bounds were chosen. Whereas the constraints on the gradient offset are straightforward (*i.e.*, from 0 to 1), the constraints on the gradient slope are given on a logarithmic scale (*i.e.*, from 10^{-3} to 10^0), such that the search runs of both the evolutionary algorithms and grid search can be more evenly spread over multiple orders of magnitude (see logarithmic βt_0 -axis of Fig. 3). In the case of 2D chromatography, the same constraints were applied in both the first and second dimension.

2.5 Genetic algorithms

In the implemented genetic algorithms (GA), sets of gradient parameters are encoded as genomes of binary numbers, with one gene per gradient parameter. More specifically, each gene is a binary number of eight bits (allowing an encoding accuracy of 1/256). Within each generation, reproduction occurs via crossover of, and point mutations within, genomes of selected individuals [35].

Firstly, for each individual to be generated, two parents are selected from the population via either the elitist or the probabilistic method. Secondly, crossover takes place at a random point along the parental genomes, with the child inheriting some bits from one and some bits from the other. Thirdly, randomised point mutations take place in the nascent genome, inverting zeroes to ones and *vice versa*.

Lastly, once a given number of individuals has been generated, the population is updated via either the comma or the plus method.

By combining the described design choices, four GA designs were implemented: comma-elitist, comma-probabilistic, plus-elitist and plus-probabilistic. Each of them has two design parameters to be optimised: the mutation rate (the probability of a given bit to be inverted) and the generation size (the number of individuals per generation). In addition, a so-called (1+1)-GA was implemented, a plus-elitist design with only one individual per generation. Although, in this case, the generation size during the evolution cycle was fixed, the initial population size was optimised instead.

2.6 Non-adaptive evolution strategies

In the implemented evolution strategies (ES), sets of gradient parameters are encoded as vectors of real numbers, with one component per gradient parameter. Within each generation, reproduction occurs via a random draw from a Gaussian distribution rather than pseudo-genetic operations [34].

The mean of the distribution represents the parent (\bar{p}), selected from the current generation. This can be the fittest individual or a fitness-weighted average of individuals, according to either the elitist or the weighted method. The standard deviation of the distribution represents the mutation strength (σ), a design parameter in ESs analogous to the mutation rate in GAs. A number of children (\bar{c}), which take their place in the next generation according to either the comma or the plus method, are sampled from the thusly defined mutation distribution:

$$\bar{c}_{g+1} \sim \mathcal{N}(\bar{p}_g, \sigma^2 \bar{I}) \quad (9)$$

Analogous to the four GA designs, four ES designs were implemented: comma-elitist, comma-weighted, plus-elitist and plus-weighted. Each of them has two design parameters to be optimised: the mutation strength and the generation size. In addition, and analogous to the previously described (1+1)-GA, a (1+1)-ES was implemented as well.

2.7 Covariance matrix adaptation evolution strategy

The ESs described above are referred to as ‘non-adaptive’, since their mutation strength – and with it the covariance matrix of their mutation distribution – is kept constant throughout the evolution cycle. More complex ESs can increase or decrease their mutation strength from one generation to the next, depending on whether the entire optimisation space is to be explored or a given area is to be exploited. This scaling of the mutation distribution is controlled by a so-called self-adaptation method [34].

Adding another layer of complexity, a covariance matrix adaptation evolution strategy (CMA-ES), first described in [36], can also stretch and rotate the mutation distribution. To do so efficiently, the mutation strength and the normalised covariance matrix ($\bar{\bar{C}}$) are adapted independently from one another. Each generation, reproduction occurs via the adapted analogue of Eq. 9:

$$\bar{c}_{g+1} \sim \mathcal{N}(\bar{p}_g, \sigma_g^2 \bar{\bar{C}}_g) \quad (10)$$

Adaptation of the normalised covariance matrix strives to increase the probability of successful mutation steps (*i.e.*, mutation steps which increase fitness) towards future generations, by favouring the direction of successful mutation steps from past generations. In parallel, adaptation of the mutation strength strives to optimise the length of future mutation steps based on past mutation steps. A cumulative and derandomised method to do both, by integrating information on successful mutation steps via deterministic formulas, is described in [37].

However, based on the results discussed in Subsections 3.1.1-3.1.2, we chose to implement a (1+1)-CMA-ES, first described in [38], analogous to the (1+1)-GA and (1+1)-ES. In this design, mutation strength adaptation is based on the so-called ‘one-fifth success rule’, which states that a (1+1) evolutionary algorithm should strive to have about one ‘success’ (*i.e.*, an increase in fitness) per five generations [34]. When the success rate becomes either too low or too high, the mutation strength should be increased or decreased, respectively. Two recent improvements on the state-of-the-art (1+1)-CMA-ES were included: active covariance matrix adaptation [39], which allows it to learn from its failures as well as its successes, and constraint handling [40], which adapts the mutation distribution when it approaches the boundaries of the optimisation space.

The reader is referred to [40] for a detailed description of the implemented (1+1)-CMA-ES, including the advised values of the design parameters of its self-adaptation method (given in Table 1 of [40]). Two design parameters were optimised in this study: the initial population size and the initial mutation strength.

2.8 Grid search

In parallel to the evolutionary algorithms, a plain grid search algorithm implemented to serve as a benchmark. This simplistic algorithm projects an orthogonal grid onto the optimisation space and performs a search run at the centre of each grid cell. Thus, the number of search runs increases with the second power (in the case of 1D chromatography) or the fourth power (in the case of 2D chromatography) of the demanded grid resolution.

3. Results and discussion

The following sections, discussing the cases of 1D and 2D chromatography, each consist of four subsections. In the first three, the three implemented classes of evolutionary algorithms (GA, ES and CMA-ES) are studied one by one, after having optimised their design parameters. The last subsection is a comparative study between the evolutionary algorithms and the grid search, including a study on the influence of sample complexity and column efficiency.

Firstly, however, we introduce two plots which are used throughout this discussion: chromatographic response function (CRF) landscapes (Fig. 3) and efficacy curves (Fig. 4).

In a CRF landscape, the CRF value of a chromatographic separation is plotted for each possible set of gradient parameters, as determined by a brute force grid search (*i.e.*, using a 200x200 grid). In the case of 1D chromatography, the two-dimensional (φ_0, β) -space can be adequately visualised by a

heatmap, as shown in Fig. 3. On the contrary, the four-dimensional $(\varphi_{0,1}, \beta_1, \varphi_{0,2}, \beta_2)$ -space in the case of 2D chromatography cannot be adequately visualised.

As a side note, the optimal range for the gradient steepness ($S_1\beta t_0/2.3$) which becomes apparent when examining a sufficiently large number of samples can be compared with the optimal gradient steepness values found in practice. In our simulations, based on the Neue-Kuß model, the average S_1 -value is 23 and the optimal βt_0 -value lies between 0.01 and 0.05 for the majority of samples. Thus, the gradient steepness lies typically between 0.1 and 0.5, a range not so different from what is reported for the linear solvent strength model in literature, namely from 0.2 to 0.4 [30].

In a so-called efficacy curve, the sample-averaged CRF-value (cf. Eq. 1) achieved by a given algorithm is plotted as a function of the number of search runs (n) it performs. Logically, the longer an algorithm searches, the better the solutions it finds. Eventually, however, the algorithm runs into the limitations of the chromatographic separations, as the given column efficiency does not necessarily suffice to completely separate each of the samples in a meta-experiment. For example, in the case shown in Fig. 4, with a sample complexity of 20 and a column efficiency of 20,000, the efficacy limit is 19.16, as determined by a brute force grid search. Hence, the best a search algorithm can do is achieve this 19.16 efficacy limit, as an efficacy of 20 is impossible to achieve with the given column efficiency.

An interesting finding is that the efficacy curve of the grid search, shown in Fig. 4a, can be accurately fitted ($R^2 > 0.99$) with a hyperbolic trendline:

$$efficacy = efficacy_{max} - \frac{1}{a \cdot n + b} \quad (11)$$

Moreover, it was found that this holds for the GA and ES as well, though not for the CMA-ES. This is shown in Fig. 4b, where the efficacy curves are transformed such that the hyperbolic trendline becomes linear, allowing to emphasise the quality of the fit:

$$1/(efficacy_{max} - efficacy) = a \cdot n + b \quad (12)$$

3.1 Gradient optimisation in 1D chromatography

To evaluate the evolutionary algorithms, each of the studied designs is applied to a reference case: chromatographic separations with a sample complexity of 20 and a column efficiency of 20,000. Firstly, their design parameters are optimised based on a series of meta-experiments, focussing on their efficacy value after 50 search runs. This suffices to provide a good indication of their performance, since the differences between efficacy curves tend to decrease beyond this point anyhow (see Subsection 3.1.4 and Fig. 6). Secondly, the most suitable design from each class is subjected to further study, by means of CRF landscapes and efficacy curves.

Nota bene, the optimisation of the design parameters of the evolutionary algorithms typically involves comparing efficacy values which differ marginally. Some of these marginal differences – resulting from slightly nudging the design parameters – are in the same order as the standard errors on the efficacy values: 0.01 in the case of 1D chromatography and 0.1 in the case of 2D chromatography. Although these differences correspond to hundreds of samples where a given peak is or isn't resolved, the optimised design parameter values proposed here should be considered as indicative values, rather than constants 'set in stone'.

3.1.1 Genetic algorithms

For each of the GA designs, optimised design parameters and their resulting efficacy values are given in Table S1. The most notable differences are those between the elitist and probabilistic designs, with the former outperforming the latter. Surprisingly, optimisation of the plus-probabilistic design led to a generation size of one. This implies that two individuals are compared in each generation: a child, which has just been generated, and its parent, which was conserved by the plus method. Since the probabilistic method implemented here avoids selecting the least fit individual, it becomes a *de facto* elitist method, consistently selecting the most fit individual. This somewhat counterintuitive result led us to include the so-called (1+1) evolutionary algorithms, on which extensive literature exists [41].

However, the (1+1) evolutionary algorithms might not be as effective at exploring the entire optimisation space as they are at exploiting a given area. To compensate for this risk, their initial population size was considered as a design parameter instead of the generation size.

The resulting (1+1)-GA outperforms each of the other GA designs, albeit slightly. Hence, we chose to subject the (1+1)-GA to further study, especially considering it combines the four properties listed below:

(a) The plus method avoids discarding a suitable set of gradient parameters from one generation to the next.

(b) The elitist method avoids wasting search runs on ‘descendants’ of unsuitable sets of gradient parameters.

(c) Performing one search run per generation reduces the number of search runs that have to be performed throughout method development.

(d) Performing a number of search runs to initialise the population reduces the risk of beginning the evolution cycle in a non-promising area of the optimisation space.

To provide visual insight in the performance of the (1+1)-GA in the context of gradient optimisation, it was applied to the chromatographic separation which corresponds to the CRF landscape shown in Fig. 3. Plotting the coordinates of the search runs performed during this optimisation on this CRF landscape (Fig. 5a-b) visualises how the (1+1)-GA assigns its search runs ‘intelligently’ (*i.e.*, gradually evolving towards an optimum), allowing it to outperform the grid search, which assigns its search runs ‘arbitrarily’ (*i.e.*, following a grid which is fixed *a priori*).

During the first 50 search runs (Fig. 5a), the optimisation space is mainly explored via a series of scattered search runs. During the next 50 runs (Fig. 5b), the most promising area is exploited more thoroughly as the search runs become more clustered near the optima of the CRF landscape. Nonetheless, many search runs are still wasted in non-promising areas, as can be explained based on the employed mutation methodology. Due the pseudo-genetic operations implemented here, some large mutation steps (*e.g.*, from 0.25 to 0.75) are more likely to occur than some small mutation steps (*e.g.*, from 0.49 to 0.51). Furthermore, due to the probability of mutating one gradient parameter while not – or barely – mutating the other, many search runs are arbitrarily clustered on – or near –

horizontal or vertical lines. These properties indicate that GAs might not be the most suitable class of evolutionary algorithms in the context of gradient optimisation.

3.1.2 Non-adaptive evolution strategies

The ES designs were optimised analogously to the GA designs, and the resulting efficacy values are given in Table S2. Since the standard errors on the efficacy values are in the order of 0.01, the differences between those in Table S2 are of borderline significance. Because it is analogous to the (1+1)-GA and combines the four properties listed above, we chose to subject the (1+1)-ES to further study.

An illustrative example (Fig. 5c-d) visualises the performance of the (1+1)-ES. As before, an initial exploration stage (Fig. 5c) and a subsequent exploitation stage (Fig. 5d) can be distinguished. However, compared to the (1+1)-GA, the (1+1)-ES appears to assign its search runs more intelligently. The Gaussian distribution implemented here favours small mutation steps ($< 2\sigma$) over large ones ($> 2\sigma$) and is isotropic, resulting in a coherent cluster of search runs. These properties, along with the efficacy values given in Tables S1 and S2, indicate that ESs might be more suitable than GAs in the context of gradient optimisation.

3.1.3 Covariance matrix adaptation evolution strategy

Continuing our study on (1+1) evolutionary algorithms, we chose to subject the (1+1)-CMA-ES to the same design optimisation and further study as was done for the GA and ES designs. With an initial mutation rate of 0.33 and an initial population size of 9, this state-of-the-art evolutionary algorithm achieves an efficacy value of 18.49 at 50 search runs, higher than both the (1+1)-GA (18.19) and (1+1)-ES (18.40). Since the (1+1)-CMA-ES can initially explore the optimisation space with a high mutation strength and subsequently exploit a given area with a low mutation strength, optimising its initial population size might not be as necessary as it is for the (1+1)-GA and (1+1)-ES. Nonetheless, the efficacy is slightly improved as compared to the (1+1)-CMA-ES with an initial population size of 1, which achieves an efficacy value of 18.44 at 50 search runs.

Analogous to the (1+1)-GA and (1+1)-ES, the performance of the (1+1)-CMA-ES was visualised (Fig. 5e-f). Because of the implemented self-adaptation method, the difference between the initial exploration stage (Fig. 5e) and the subsequent exploitation stage (Fig. 5f) is striking. Adaptation of the mutation strength – *i.e.*, scaling the mutation distribution – results in a concentrated cluster of search runs, while adaptation of the normalised covariance matrix – *i.e.*, stretching and rotating the mutation distribution – shapes this cluster into a diagonally oriented ellipsoid. Thus, in this particular example, the (1+1)-CMA-ES converges to the global optimum of the CRF landscape.

However, for other samples, the (1+1)-CMA-ES was found to converge to a local optimum instead of to the global optimum. Since their limited sensitivity to local optima is one of the reasons why evolutionary algorithms could be useful to solve method development problems, the risk of premature convergence of the (1+1)-CMA-ES should be considered a major drawback.

3.1.4 Comparative study

Fig. 6 mutually compares the efficacy curves of the different (1+1) evolutionary algorithms studied in the previous Subsections, using the performance of a plain grid search as a reference. Clearly, each of the evolutionary algorithms has an advantage over the grid search, though there are also differences between the evolutionary algorithms themselves.

These differences can be evaluated in two directions: vertically and horizontally. Evaluating them vertically answers the question ‘which algorithm proposes the solution with the highest CRF value after a given number of search runs?’, whereas evaluating them horizontally answers the question ‘which algorithm has to perform the least number of search runs before achieving a given CRF value?’. Since the latter directly addresses the required experimental or computational time, the latter can be considered a more relevant question than the former.

As most efficacy curves tend to be hyperbolic (*i.e.*, they can be accurately fitted with Eq. 11), their vertical differences are small and decrease after some number of search runs (*ca.* 50), as they come closer to the efficacy limit. On the other hand, their horizontal differences are large and increase proportionally to the number of search runs, as the convergence rate differs between algorithms. These differences in convergence rate can be quantified based on the α - and β -parameters of Eq. 11, which were estimated by non-linear least squares regression. In the large n limit of Eq. 11, the number of search runs at which an algorithm achieves a given efficacy value becomes inversely proportional to the α -value, as the β -value becomes negligible (*cf.* offset and slope of trendline in Fig. 4b, which is highly accurate for $n > 100$). Thus, the GA ($\alpha = 0.0153$) and ES ($\alpha = 0.0237$) can be said to respectively save up to 39% and 60% of search runs as compared to the grid search ($\alpha = 0.0094$). These values support the discussion of the illustrative examples in Sections 3.1.1 and 3.1.2, confirming the ES assigns its search runs more intelligently than the GA.

As an exception, the efficacy curve of the CMA-ES cannot be accurately fitted with Eq. 11, as it appears to converge exponentially rather than hyperbolically. Furthermore, it converges to an efficacy value (18.75) considerably below the efficacy limit (19.16), confirming the risk of premature convergence discussed in Section 3.1.3. The self-adaptation method causes the CMA-ES to propose suboptimal solutions for some samples. In practice, especially if the number of components in the sample is not known *a priori*, this might result in missing one component of the sample – or more, though the data indicate this is not as likely.

As a preliminary conclusion, the ES can be considered the most suitable evolutionary algorithm in the context of gradient optimisation. The GA is slower by a factor of about one-and-a-half and the CMA-ES, although it is intrinsically faster, might be too sensitive to local optima for this application. To verify whether this preliminary conclusion holds under other conditions, the influence of sample complexity and column efficiency was studied.

In a first comparison, shown in Fig. 7, the sample complexity was kept constant while the column efficiency was varied. Logically, this influences the peak capacity of the chromatographic separations. With a column efficiency of 10,000 (Fig. 7a) and 30,000 (Fig. 7b), the efficacy limit is respectively decreased to 17.99 and increased to 19.55. In parallel, the convergence rate of both the evolutionary algorithms and grid search is decreased and increased as well, in line with the difficulty of the method

development problem. This is related to the corresponding changes in the elevation and ruggedness of the CRF landscapes. As the column efficiency increases, low and narrow ‘hills’ coalesce into high and broad ‘plateaus’, corresponding to better separation conditions which are easier to find. This also reduces the risk of premature convergence of the CMA-ES, which gets slightly closer to the efficacy limit in the case of Fig. 7b (within 0.34) than it does in the case of Fig. 7a (within 0.45).

Again, the differences in convergence rate as compared to the grid search can be quantified based on the α -parameter, obtained by fitting Eq. 11. These show the GA and ES can respectively save up to 39% and 64% of search runs, in the case Fig. 7a, or up to 36% and 57% of search runs, in the case of Fig. 7b. Although there is but a weak influence of the column efficiency on these percentages, there is a strong influence on the number of search runs to which they correspond. The lower the column efficiency, the more difficult it is to find suitable separation conditions, such that saving a given percentage of search runs corresponds to a larger amount.

In a second comparison, shown in Fig. 8, the sample complexity was varied while the column efficiency was chosen such that the chromatographic separations are neither trivial nor unfeasible. For a sample complexity of 15 (Fig. 8a) and 25 (Fig 8b), a column efficiency of respectively 10,000 and 30,000 was chosen, resulting in efficacy limits of 14.26 and 23.88. Although both cases have a similar feasibility, there is a stark difference in terms of the difficulty of method development.

As before, this can be explained based on changes in the CRF landscapes. As the number of components increases, so does the number of ‘features’ (hills, plateaus, valleys, *et cetera*) in the CRF landscapes. Suitable separation conditions might be very well present, given an appropriate column efficiency is applied, but they are more difficult to find because of the increasing number of local optima. Logically, this also increases the risk of premature convergence of the CMA-ES, which strands slightly further from the efficacy limit in the case of Fig. 8b (within 0.59) than it does in the case of Fig. 8a (within 0.26).

Again quantifying the differences in convergence rate via the α -parameter of Eq. 11, the GA and ES can respectively save up to 28% and 55% of search runs, in the case Fig. 8a, or up to 43% and 65% of search runs, in the case of Fig. 8b. Not only does the sample complexity influence these percentages, it also influences the typical number of search runs that have to be performed. Thus, in more demanding applications in terms of sample complexity, evolutionary algorithms save a larger percentage of a larger number of search runs.

3.2 Gradient optimisation in 2D chromatography

Since method development in 2D chromatography is intrinsically more difficult than in 1D chromatography, it can be hypothesised that the evolutionary algorithms are especially advantageous in this context. Due to the increase in number of gradient parameters as well as in number of components, the number of search runs required to find suitable separation conditions is inevitably larger. To limit the computational workload, we restricted ourselves to the full-in-fraction method, in which the gradient programme is characterised by four parameters ($\varphi_{0,1}, \beta_1, \varphi_{0,2}, \beta_2$).

As a reference case, the studied evolutionary algorithms are applied to chromatographic separations with a sample complexity of 150 and a column efficiency of 20,000 (in the first dimension) and 1,000 (in the second dimension), focussing on their efficacy value at 100 search runs. Firstly, the performance of each of the studied designs is evaluated, using the optimised design parameter values given in Tables S1 and S2. Secondly, the design parameters of the (1+1) evolutionary algorithms are re-optimised. Finally, the optimised designs are subjected to a comparative study.

3.2.1 Genetic algorithms

Efficacy values for each of the GA designs are given in Table S3. Since the standard errors – originating from the variance between samples – on these efficacy values are in the order of 0.1, it would be irrelevant to discuss the differences between each of the GA designs in detail. It suffices to state that there is some similarity between Tables S1 and S3, with elitist designs outperforming probabilistic designs. Thus, there appears to be no incentive to swap the (1+1)-GA for another when moving from the case of 1D chromatography to the case of 2D chromatography.

However, there does appear to be an incentive to re-optimize the design parameters, since this significantly increases the efficacy of the (1+1)-GA (*cf.* last row in Table S3). Because of the intrinsic vastness of the optimisation space, which is now four-dimensional instead of two-dimensional, the optimal number of initial search runs increases (from 12 to 17). Furthermore, a decrease in mutation rate (from 0.25 to 0.17) somewhat compensates the increase in genome size (*i.e.*, number of gradient parameters), such that the overall occurrence of mutations does not become excessive.

3.1.2 Non-adaptive evolution strategies

The results for the ES designs, given in Table S4, are analogous to those of the GA designs. Whereas Tables S1 and S3 suggest the superiority of elitist over probabilistic designs, Tables S2 and S4 suggest the superiority of elitist over weighted designs. In both cases, there is no decisive argument to deviate from the previously established preference for (1+1) evolutionary algorithms. Analogous to the re-optimisation of the (1+1)-GA, re-optimisation of the (1+1)-ES leads to an increase in initial population size (from 8 to 18) and a decrease in mutation strength (from 0.17 to 0.10).

3.2.3 Covariance matrix adaptation evolution strategy

Analogous to the GA and ES designs, the performance of the (1+1)-CMA-ES was first evaluated using the optimised values of the initial mutation strength (0.33) and initial population size (9) from Section 3.1.3. This yields an efficacy value of 143.1 at 100 search runs, higher than the efficacy values in Tables S3 and S4. However, the risk of premature convergence is expected to turn up eventually. Surprisingly, re-optimisation of the initial mutation strength and initial population size of the (1+1)-CMA-ES did not lead to a decrease or increase of these two design parameter values.

3.2.4 Comparative study

Fig. 9 shows the efficacy curves of the studied (1+1) evolutionary algorithms for the case of a sample complexity of 150. These are compared to the efficacy values achieved by 3x3x3x3 and 5x5x5x5 grid searches, that is, grid searches which evaluates all combinations, either 81 or 625, that arise from considering either three or five values for each of the four gradient parameters ($\varphi_{0,1}, \beta_1, \varphi_{0,2}, \beta_2$).

On the one hand, there are some similarities with the results from 1D chromatography, concerning how the evolutionary algorithms relate to one another in terms of performance. The ES outperforms the GA, converging faster by a factor of about one-and-a-half, whereas the CMA-ES converges intrinsically faster, but does so to local as well as global optima.

On the other hand, there are some differences with the results from 1D chromatography, in terms of the saved number of search runs. Because of the intrinsic vastness of the optimisation space, it takes more search runs for both the grid search and evolutionary algorithms to find suitable separation conditions (note the difference in scale between Fig. 6-8 and Fig. 9-10). However, the influence of this intrinsic difficulty is much stronger for the grid search than it is for the evolutionary algorithms, such that the latter can achieve with fewer than 100 search runs what the former achieves with 625 search runs. Whereas in the case of 1D chromatography, evolutionary algorithms could speed up solving gradient optimisation problems by some moderate percentage (see Subsection 3.1.4), they could do so by a factor of six or more in the case of 2D chromatography.

These results are supported by the additional efficacy curves shown in Fig. 10, generated for a sample complexity of 100 (Fig. 10a) and 200 (Fig. 10b). The results are in line with those in Subsection 3.1.4, as an increase in sample complexity – and thus the difficulty of the method development problem – leads to a decrease in convergence rate. In both cases, however, the GA surpasses the efficacy value of the 5x5x5x5 grid search with fewer than 100 search runs, whereas the ES and CMA-ES do so in fewer than 70 search runs. Apparently, the difficulty arising from the sample complexity is subordinate to the difficulty arising from the intrinsic vastness of the optimisation space.

4. Conclusions

After optimisation of their design parameters, evolutionary algorithms such as genetic algorithms (GA), evolution strategies (ES) and the covariance matrix adaptation evolution strategy (CMA-ES) can significantly speed up searches in the parameter spaces for method development in 1D- and 2D-LC. Having found that, at least for $n > 100$, the evolution of the objective function value (CRF) as a function of the number of search runs (n) follows a simple hyperbolic law in case of the GA and the ES, as well as for the plain grid search, the convergence rate parameter (a) of this hyperbolic function can be used to quantify the difference in search performance between these algorithms. The presently studied CMA-ES does not follow this law, as it converges intrinsically faster, but tends to end up in local optima. This makes this search method unsuitable in the context of gradient optimisation, except for cases where only a limited number of searches ($n < 50$) is required or permitted, as is the case in algorithm-guided experimental searches. To solve difficult separation problems ($n > 50$), interpretation of the convergence rate parameter (a) in Eq. 11 shows that the evolutionary algorithms can speed up searches by a moderate percentage (*ca.* 40% for GA and 60% for ES) in the case of 1D-LC, whereas they can do so by a considerable factor (*ca.* factor of 6 for GA and factor 9 for ES) in the case

of 2D-LC. As can be expected, the advantage of the ES over the GA and especially the plain grid search grows with increasing difficulty of the separation problem. This further allows to infer that the advantage of the evolutionary algorithms will be considerably larger for search problems involving more parameters (*e.g.*, including pH, temperature, mobile phase composition, stationary phase type, or more elaborate gradient programmes) than considered in the present study.

Whereas the present study was conducted by greatly simplifying the chromatographic reality (perfect orthogonality, equal concentration of analytes in samples, Gaussian peaks) to obtain the purest possible view on the possibilities of the different search algorithms, it is obvious that a follow-up study will have to focus on the effect of these non-idealities, using the present results as a yardstick against which these effects can be measured.

To further improve the algorithms, a significant gain can be expected from a more sophisticated choice for the initial search runs (which were randomised in the present study). This could be done by including prior know-how on the most promising area of the method development parameter space, either available through the analyst's experience or through crude optimisation methods such as the predictive elution window stretching and shifting (PEWS²) method [42]. A second modification concerns the (1+1)-CMA-ES. Although it is somewhat criticised in Sections 3.1 and 3.2 because of its risk of premature convergence, it is intrinsically the fastest of the studied evolutionary algorithms. Tempering its self-adaptation method, for example by setting a lower bound to the mutation strength [43], might solve this convergence problem. In some sense, this approach can be considered as a compromise between the CMA-ES with and the ES without self-adaptation method.

Acknowledgements

BH and GD gratefully acknowledge the FWO-FRNS grant EOS nr. 30897864.

References

- [1] D.R. Stoll, K. Shoykhet, P. Petersson, S. Buckenmaier, Active solvent modulation: A valve-based approach to improve separation compatibility in two-dimensional liquid chromatography, *Analytical Chemistry* 89 (2017) 9260-9267, <https://doi.org/10.1021/acs.analchem.7b02046>.
- [2] M. Pursch, A. Wegener, S. Buckenmaier, Evaluation of active solvent modulation to enhance two-dimensional liquid chromatography for target analysis in polymeric matrices, *Journal of Chromatography A* 1562 (2018) 78-86, <https://doi.org/10.1016/j.chroma.2018.05.059>.
- [3] D.R. Stoll, P.W. Carr, Two-dimensional liquid chromatography: a state of the art tutorial, *Analytical Chemistry* 89 (2017) 519-531, <https://doi.org/10.1021/acs.analchem.6b03506>.
- [4] J.W. Dolan, D.C. Lommen, L.R. Snyder, Drylab[®] computer simulation for high-performance liquid chromatographic method development : II. Gradient Elution, *Journal of Chromatography A* 485 (1989) 91-112, [https://doi.org/10.1016/S0021-9673\(01\)89134-2](https://doi.org/10.1016/S0021-9673(01)89134-2).
- [5] S.V. Galushko, A.A. Kamenchuk, G.L. Pit, Calculation of retention in reversed-phase liquid chromatography: IV. ChromDream software for the selection of initial conditions and for simulating

614 chromatographic behaviour, *Journal of Chromatography A* 660 (1994) 47-59,
 615 [https://doi.org/10.1016/0021-9673\(94\)85098-4](https://doi.org/10.1016/0021-9673(94)85098-4).

616 [6] G. Vivó-Truyols, S. van der Wal, P.J. Schoenmakers, Comprehensive study on the optimization of
 617 online two-dimensional liquid chromatographic systems considering losses in theoretical peak
 618 capacity in first- and second-dimensions: a Pareto-optimality approach, *Analytical Chemistry*, 82
 619 (2010) 8525-8536, <https://doi.org/10.1021/ac101420f>.

620 [7] B.W.J. Pirok, S. Pous-Torres, C. Ortiz-Bolsico, G. Vivó-Truyols, P.J. Schoenmakers, Program for the
 621 interpretive optimization of two-dimensional resolution, *Journal of Chromatography A* 1450 (2016)
 622 29-37, <https://doi.org/10.1016/j.chroma.2016.04.061>.

623 [8] B.W.J. Pirok, A.F. Gargano, P.J. Schoenmakers, Optimizing separations in online comprehensive
 624 two-dimensional liquid chromatography, *Journal of Separation Science* 41 (2018) 68-98,
 625 <https://doi.org/10.1002/jssc.201700863>.

626 [9] B.W.J. Pirok, D.R. Stoll, P.J. Schoenmakers, Recent developments in two-dimensional liquid
 627 chromatography: Fundamental improvements for practical applications, *Analytical Chemistry* 91
 628 (2019) 240-263, <https://doi.org/10.1021/acs.analchem.8b04841>.

629 [10] J.H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with*
 630 *Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor,
 631 1975.

632 [11] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary*
 633 *Programming, Genetic Algorithms*, Oxford University Press, Oxford, 1996.

634 [12] R.M. Lopes Marques, P.J. Schoenmakers, C.B. Lucasius, L. Buydens, Modelling chromatographic
 635 behaviour as a function of pH and solvent composition in RPLC, *Chromatographia* 36 (1993) 83-95.

636 [13] P.J. Schoenmakers, R. Tijssen, Modelling retention of ionogenic solutes in liquid chromatography
 637 as a function of pH for optimization purposes, *Journal of Chromatography A*, 656 (1993) 577-590.

638 [14] P. Nikitas, A. Pappa-Lousi, A. Papageorgiou, A. Zitrou, On the use of genetic algorithms for
 639 response surface modeling in high-performance liquid chromatography and their combination with
 640 the Microsoft Solver, *Journal of Chromatography A* 942 (2002) 93-105,
 641 [https://doi.org/10.1016/S0021-9673\(01\)01353-X](https://doi.org/10.1016/S0021-9673(01)01353-X).

642 [15] P. Nikitas, A. Pappa-Lousi, New approach to linear gradient elution used for optimisation in
 643 reversed-phase liquid chromatography, *Journal of Chromatography A* 1068 (2005) 279-287,
 644 <https://doi.org/10.1016/j.chroma.2005.02.004>.

645 [16] P. Nikitas, A. Pappa-Lousi, P. Agrafiotou, Multilinear gradient elution optimisation in reversed-
 646 phase liquid chromatography using genetic algorithms, *Journal of Chromatography A* 1120 (2006) 299-
 647 307, <https://doi.org/10.1016/j.chroma.2006.01.005>.

648 [17] Z. Zhang, M. Mazzotti, M. Morbidelli, Multiobjective optimization of simulated moving bed and
 649 Varicol processes using a genetic algorithm, *Journal of Chromatography A* 989 (2003) 95-108,
 650 [https://doi.org/10.1016/S0021-9673\(02\)01802-2](https://doi.org/10.1016/S0021-9673(02)01802-2).

651 [18] R. Leardi, Genetic algorithms in chemistry, *Journal of Chromatography A* 1158 (2007) 226-233,
652 <https://doi.org/10.1016/j.chroma.2007.04.025> .

653 [19] C. Ortiz-Bolsico, J. R. Torres-Lapasió, M. C. García-Alvarez-Coque, Optimisation of gradient elution
654 with serially-coupled columns. Part I: Single linear gradients, *Journal of Chromatography A* 1350 (2014)
655 51-60, <https://doi.org/10.1016/j.chroma.2014.05.017>.

656 [20] C. Ortiz-Bolsico, J. R. Torres-Lapasió, M. C. García-Alvarez-Coque, Optimisation of gradient elution
657 with serially-coupled columns Part II: Multi-linear gradients, *Journal of Chromatography A* 1373 (2014)
658 51-60, <https://doi.org/10.1016/j.chroma.2014.10.100>.

659 [21] M. Talebi, G. Schuster, R. A. Shellie, R. Szucs, P. R. Haddad, Performance comparison of partial
660 least squares-related variable selection methods for quantitative structure retention relationship
661 modelling of retention times in reversed-phase liquid chromatography, *Journal of Chromatography A*
662 1424 (2015) 69-76, <https://doi.org/10.1016/j.chroma.2015.10.099>.

663 [22] M. Taraji, P. R. Haddad, R. I. J. Amos, M. Talebi, R. Szucs, J. W. Dolan, C. A. Pohl, Prediction of
664 retention in hydrophilic interaction liquid chromatography using solute molecular descriptors based
665 on chemical structures, *Journal of Chromatography A* 1486 (2017) 59-67,
666 <https://doi.org/10.1016/j.chroma.2016.12.025>.

667 [23] S. Mun, Optimization of production rate, productivity, and product concentration for a simulated
668 moving bed process aimed at fucose separation using standing-wave-design and genetic algorithm,
669 *Journal of Chromatography A* 1575 (2018) 113-121, <https://doi.org/10.1016/j.chroma.2018.09.025>.

670 [24] T. Alvarez-Segura, S. López-Ureña, J. R. Torres-Lapasió, M. C. García-Alvarez-Coque, Multi-scale
671 optimisation vs. genetic algorithms in the gradient separation of diuretics by reversed-phase liquid
672 chromatography, *Journal of Chromatography A* 1609 (2020) 460427,
673 <https://doi.org/10.1016/j.chroma.2019.460427>.

674 [25] A. Niazi, R. Leardi, Genetic algorithms in chemometrics, *Journal of Chemometrics* 26 (2012) 345-
675 351, <https://doi.org/10.1002/cem.2426>.

676 [26] R.M.B.O. Duarte, J.T.V. Matos, A.C. Duarte, A new chromatographic response function for
677 assessing the separation quality in comprehensive two-dimensional liquid chromatography, *Journal of*
678 *Chromatography A* 1225 (2012) 121-131, <https://doi.org/10.1016/j.chroma.2011.12.082>.

679 [27] J.T.V. Matos, R.M.B.O. Duarte, A.C. Duarte, Chromatographic response functions in 1D and 2D
680 chromatography as tools for assessing chemical complexity, *Trends in Analytical Chemistry* 45 (2013)
681 14-23, <https://doi.org/10.1016/j.trac.2012.12.013>.

682 [28] G. Vivó-Truyols, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, Enhanced calculation of optimal
683 gradient programs in reversed-phase liquid chromatography, *Journal of Chromatography A* 1018
684 (2003) 183-196, <https://doi.org/10.1016/j.chroma.2003.08.043>.

685 [29] U.D. Neue, H.-J. Kuß, Improved reversed-phase gradient retention modeling, *Journal of*
686 *Chromatography A* 1217 (2010) 3794-3803, <https://doi.org/10.1016/j.chroma.2010.04.023>.

687 [30] L.R. Snyder, J.W. Dolan, High-Performance Gradient Elution: The Practical Application of the
688 Linear-Solvent-Strength Model, John Wiley & Sons, Hoboken, 2007.

- [31] D.R. Stoll, P.W. Carr, Fast, comprehensive two-dimensional HPLC separation of tryptic peptides based on high-temperature HPLC, *Journal of the American Chemical Society* 127 (2005) 5034-5035, <https://doi.org/10.1021/ja050145b>.
- [32] P. Jandera, T. Hájek, P. Česla, Comparison of various second-dimension gradient types in comprehensive two-dimensional liquid chromatography, *Journal of Separation Science* 33 (2010) 1382-1397, <https://doi.org/10.1002/jssc.200900808>.
- [33] U.D. Neue, D.H. Marchand, L.R. Snyder, Peak compression in reversed-phase gradient elution, *Journal of Chromatography A* 1111 (2006) 32-39, <https://doi.org/10.1016/j.chroma.2006.01.104>.
- [34] H.-G. Beyer, H.-P. Schwefel, Evolution strategies. A comprehensive introduction, *Natural Computing* 1 (2002) 3-52, <https://doi.org/10.1023/A:1015059928466>.
- [35] D. Shiffman, *The Nature of Code*, The Magic Book Project, San Fransisco, 2012.
- [36] N. Hansen, A. Ostermeier, Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, *Proceedings of IEEE International Conference on Evolutionary Computation* (1996) 312-317, <https://doi.org/10.1109/ICEC.1996.542381>.
- [37] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, *Evolutionary Computation* 9(2) (2001) 159-195, <https://doi.org/10.1162/106365601750190398>.
- [38] C. Igel, T. Suttorp, N. Hansen, A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies, *Genetic and Evolutionary Computation Conference* (2006) 453-460, <https://doi.org/10.1145/1143997.1144082>.
- [39] D.V. Arnold, N. Hansen, Active covariance matrix adaptation for the (1 + 1)-CMA-ES, *Genetic and Evolutionary Computation Conference* (2010) 385-392, <https://doi.org/10.1145/1830483.1830556>.
- [40] D.V. Arnold, N. Hansen, A (1+1)-CMA-ES for constrained optimisation, *Genetic and Evolutionary Computation Conference* 2012 (2012) 297-304, <https://doi.org/10.1145/2330163.2330207>.
- [41] S. Droste, T. Jansen, I. Wegener, On the analysis of the (1+1) evolutionary algorithm, *Theoretical Computer Science* 276 (2002) 51-81, [https://doi.org/10.1016/S0304-3975\(01\)00182-7](https://doi.org/10.1016/S0304-3975(01)00182-7).
- [42] E. Tyteca, A. Liekens, D. Clicq, A. Fanigliulo, B. Debrus, S. Rudaw, D. Guillarme, G. Desmet, Predictive elution window stretching and shifting as a generic search strategy for automated method development for liquid chromatography, *Analytical Chemistry* 84 (2012) 7823-7830, <https://doi.org/10.1021/ac301331g>.
- [43] S. Meyer-Nieberg, H.-G. Beyer, Self-adaptation in evolutionary algorithms, *Studies in Computational Intelligence* 54 (2007) 47-75, https://doi.org/10.1007/978-3-540-69432-8_3.

Figure captions

Fig. 1. Examples of the idealised chromatograms of simulated reversed-phase chromatographic separations with gradient elution. **(a)** 1D-LC chromatogram ($s = 20$, $N = 20,000$ and $\varphi = 0.30 + 0.03 \cdot (t/t_0)$). **(b)** 2D-LC chromatogram ($s = 150$, $N_1 = 20,000$, $N_2 = 1,000$, $\varphi_1 = 0.30 + 0.03 \cdot (t_1/t_{0,1})$ and $\varphi_2 = 0.30 + 0.03 \cdot (t_2/t_{0,2})$).

Fig. 2. General schematic of an evolutionary algorithm applied in the context of method development in liquid chromatography.

Fig. 3. Example of a CRF landscape ($s = 20$ and $N = 20,000$). The black dots represent the search runs of a 10x10 grid search.

Fig. 4. Examples of efficacy curves ($s = 20$ and $N = 20,000$). **(a)** Calculated efficacy data of the grid search (dots) fitted with a hyperbolic trendline (solid line). The dashed line represents the efficacy limit, *i.e.*, the average of the maximally achievable CRF-values for each of the samples in the meta-experiment. **(b)** Transformed efficacy data (*cf.* Eq. 12) of the grid search (dots), GA (green line) and ES (blue line), as well as the best fit for each algorithm (dashed lines), based on non-linear least squares regression (*cf.* Eq. 11).

Fig. 5. Example of the distribution of search runs 1 to 50 (left) and 51 to 100 (right) performed during the optimisation of a chromatographic separation, plotted as black dots on the corresponding CRF landscape ($s = 20$ and $N = 20,000$). **(a-b)** Applying the (1+1)-GA. **(c-d)** Applying the (1+1)-ES. **(e-f)** Applying the (1+1)-CMA-ES.

Fig. 6. Efficacy curves of the four studied algorithms ($s = 20$ and $N = 20,000$): GA (green), ES (blue), CMA-ES (red) and grid search (black dots). The dashed line represents the efficacy limit.

Fig. 7. Efficacy curves of the four studied algorithms: GA (green), ES (blue), CMA-ES (red) and grid search (black dots). The dashed line represents the efficacy limit. **(a)** With $s = 20$ and $N = 10,000$. **(b)** With $s = 20$ and $N = 30,000$.

Fig. 8. Efficacy curves of the four studied algorithms: GA (green), ES (blue), CMA-ES (red) and grid search (black dots). The dashed line represents the efficacy limit. **(a)** With $s = 15$ and $N = 10,000$. **(b)** With $s = 25$ and $N = 30,000$.

Fig. 9. Efficacy curves of the four studied algorithms ($s = 150$, $N_1 = 20,000$ and $N_2 = 1,000$): GA (green), ES (blue), CMA-ES (red) and grid search (crosses). The dashed line represents the efficacy limit.

Fig. 10. Efficacy curves of the four studied algorithms: GA (green), ES (blue), CMA-ES (red) and grid search (crosses). The dashed line represents the efficacy limit. **(a)** With $s = 100$, $N_1 = 20,000$ and $N_2 = 1,000$. **(b)** With $s = 200$, $N_1 = 20,000$ and $N_2 = 1,000$.

Figure 1

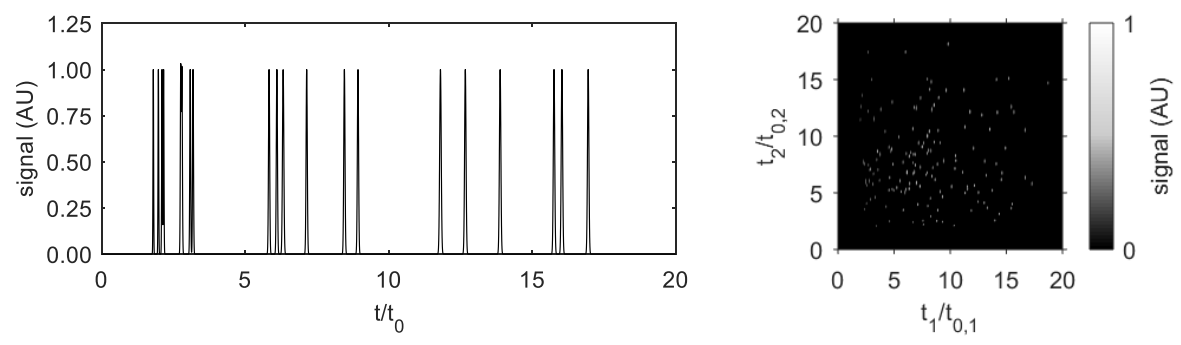


Figure 2



Figure 3

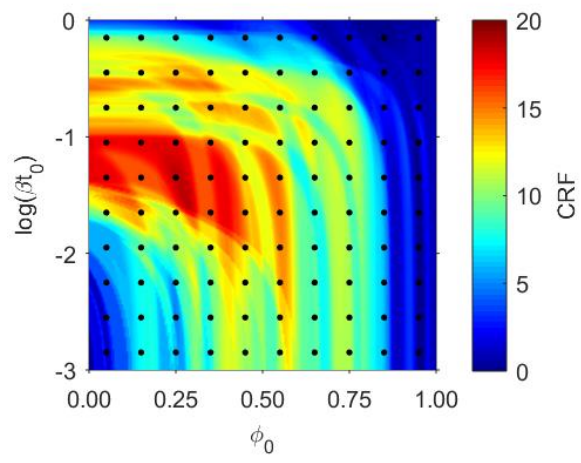


Figure 4

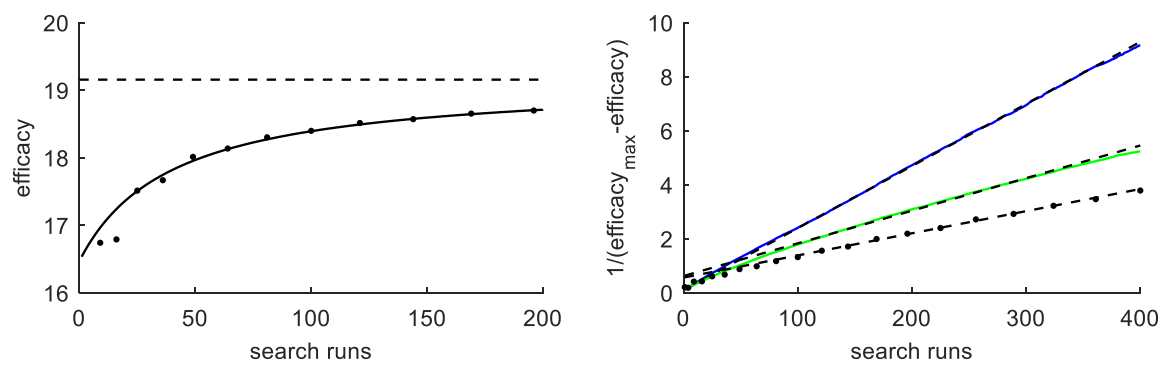


Figure 5

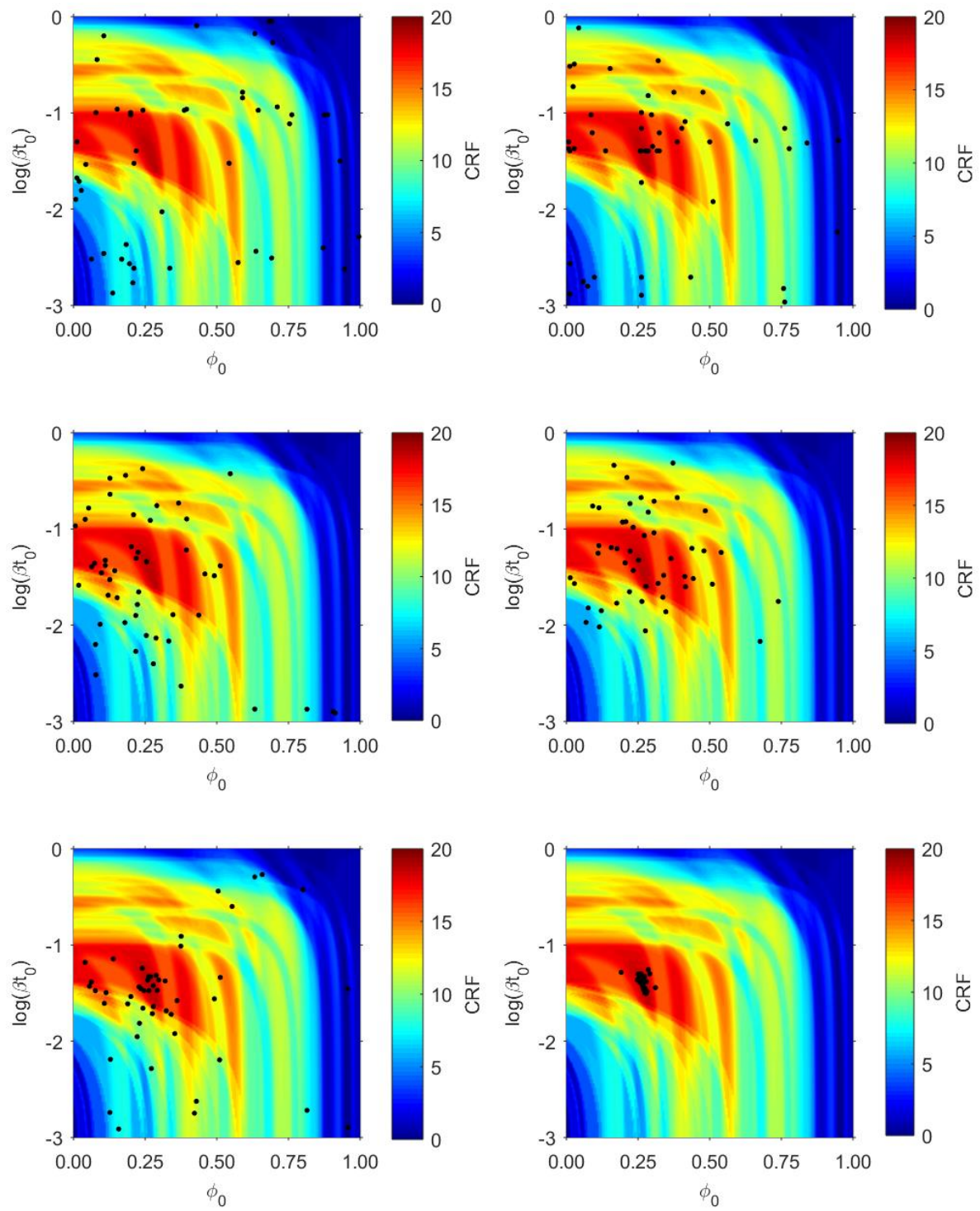


Figure 6

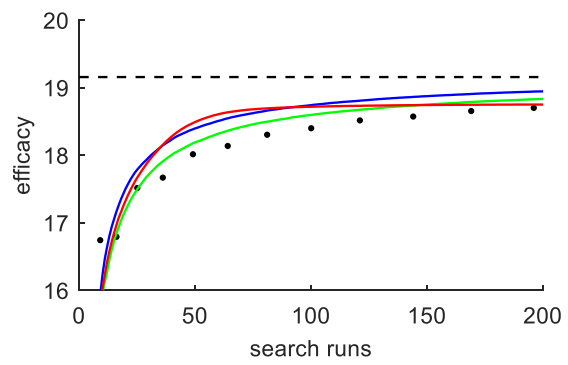


Figure 7

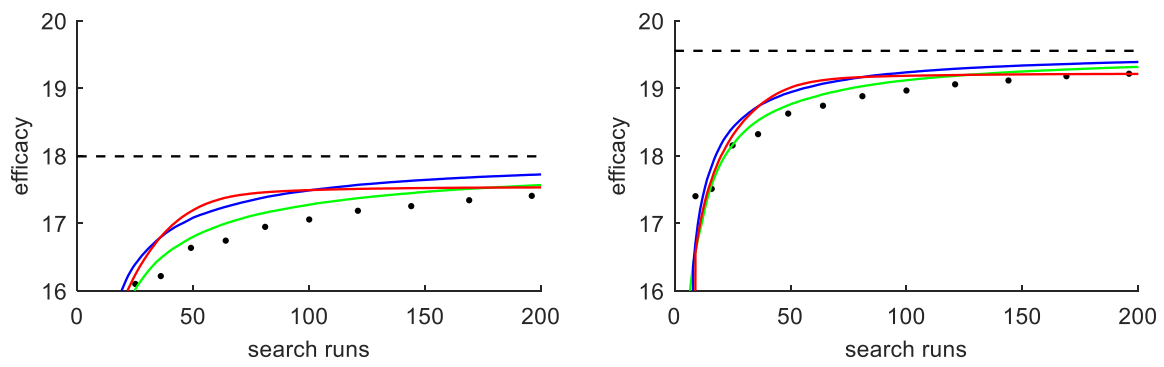


Figure 8

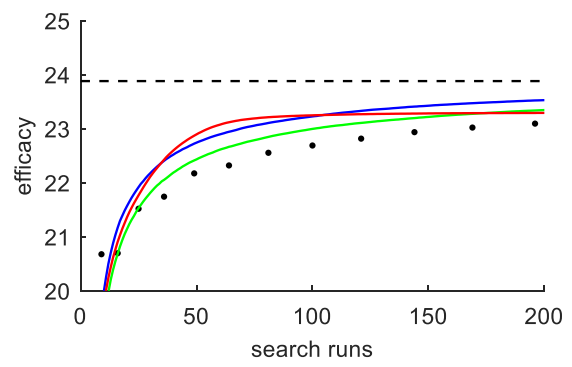
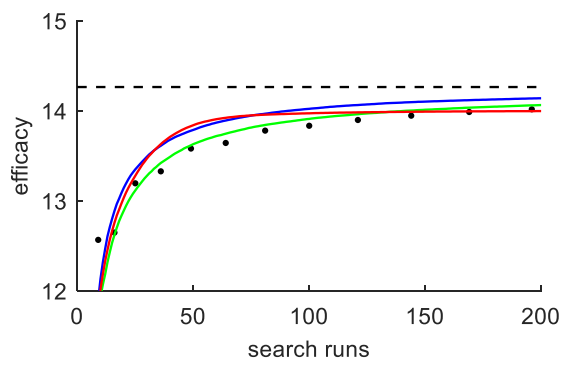


Figure 9

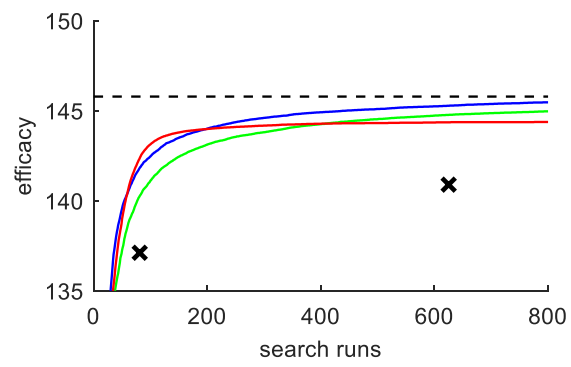


Figure 10

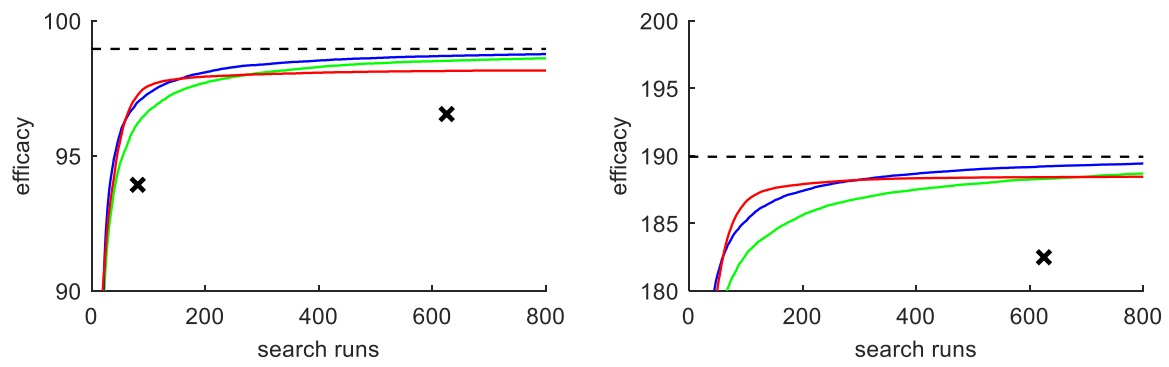


Table 1. Parameters influencing the difficulty of method development: sample complexity and column efficiency.

	1D chromatography	2D chromatography
<i>s</i>	15 – 20 – 25	100 – 150 – 200
<i>N</i>	10,000 - 20,000 - 30,000	20,000 x 1,000

Supporting Material

Application of evolutionary algorithms to optimise one- and two-dimensional gradient chromatographic separations

Bram Huygens⁽¹⁾, Kyriakos Efthymiadis⁽²⁾, Ann Nowé⁽²⁾, Gert Desmet^(1,*)

⁽¹⁾ *Vrije Universiteit Brussel, Department of Chemical Engineering, Pleinlaan 2, 1050 Brussels, Belgium*

⁽²⁾ *Vrije Universiteit Brussel, Artificial Intelligence Lab, Pleinlaan 2, 1050 Brussels, Belgium*

^(*) Corresponding author. E-mail address: gedesmet@vub.ac.be

Abstract

This supporting material provides additional Tables summarizing the most important technical detail data discussed in the text.

Table of contents

Table S1: Optimised design parameters of the GA designs (1D chromatography).

Table S2: Optimised design parameters of the ES designs (1D chromatography).

Table S3: Efficacy values of the GA designs at 100 search runs (2D chromatography).

Table S4: Efficacy values of the ES designs at 100 search runs (2D chromatography).

Table S1. Optimised design parameters of the GA designs, and their efficacy value at 50 search runs ($s = 20$ and $N = 20,000$). As a reference, a 7x7 grid search yields an efficacy value of 18.01 and the efficacy limit is 19.16.

design	mutation rate	generation size	efficacy
comma-elitist	0.20	16	18.14
comma-probabilistic	0.25	5	17.95
plus-elitist	0.25	6	18.14
plus-probabilistic	0.25	1	18.10
(1+1)-GA	0.25	12 ^a	18.19

^a For the (1+1)-GA, the initial population size was optimised instead of the generation size.

Table S2. Optimised design parameters of the ES designs, and their efficacy value at 50 search runs ($s = 20$ and $N = 20,000$). As a reference, a 7x7 grid search yields an efficacy value of 18.01 and the efficacy limit is 19.16.

design	mutation strength	generation size	efficacy
comma-elitist	0.14	6	18.42
comma-weighted	0.14	5	18.36
plus-elitist	0.17	5	18.37
plus-weighted	0.20	2	18.37
(1+1)-ES	0.17	8 ^a	18.40

^a For the (1+1)-ES, the initial population size was optimised instead of the generation size.

Table S3. Efficacy values of the GA designs at 100 search runs ($s = 150$, $N_1 = 20,000$ and $N_2 = 1,000$). As a reference, a 3x3x3x3 grid search yields an efficacy value of 137.1 and the efficacy limit is 145.8.

design	efficacy
comma-elitist	140.2
comma-probabilistic	137.1
plus-elitist	140.0
plus-probabilistic	139.7
(1+1)-GA	140.0
(1+1)-GA ^a	141.1

^a With re-optimised design parameters: a mutation rate of 0.17 instead of 0.25 and an initial population size of 17 instead of 12.

Table S4. Efficacy values of the ES designs at 100 search runs ($s = 150$, $N_1 = 20,000$ and $N_2 = 1,000$). As a reference, a 3x3x3x3 grid search yields an efficacy value of 137.1 and the efficacy limit is 145.8.

design	Efficacy
comma-elitist	141.7
comma-weighted	141.0
plus-elitist	141.6
plus-weighted	141.0
(1+1)-ES	141.7
(1+1)-ES ^a	142.5

^a With re-optimised design parameters: a mutation strength of 0.10 instead of 0.17 and an initial population size of 18 instead of 8.