Impact of Population Size, Selection and Multi-Parent Recombination within a Customized NSGA-II and a Landscape Analysis for Biochemical Optimization

Susanne Rosenthal, Markus Borschbach
University of Applied Sciences, FHDW
Faculty of Computer Science, Chair of Optimized Systems,
Hauptstr. 2, D-51465 Bergisch Gladbach, Germany
Email: {susanne.rosenthal, markus.borschbach}@fhdw.de

***Abstract*—The main task in the drug design process is the prediction of the peptide structure and the bioactivity with the focus on simultaneously optimization of molecular peptide features. The synthesis and laboratory screening are the conventional but cost-intensive steps for optimization. Multi-objective genetic algorithms provide a range of methods for an efficient design of drug peptides. A customized NSGA-II has been especially evolved for biochemical optimization with the focus on producing a great number of very different high quality peptides within a very low number of generations e.g., under 20, termed early convergence. The focus of this work are an insight into the impact of the interdependence between the selection procedure and the population size, the empirical verification of the early convergence behavior within a limited range of population size and the influence of multi-parent recombination on the algorithm performance. These purposes are exemplary investigated on two different dimensional biochemical optimization problems, which are concrete, but as generic as possible. A landscape analysis is performed to gain an insight into the characteristic features and difficulties of the multi-objective optimization problems. The performance is assessed on the basis of a convergence indicator especially evolved for our preference of comparing the convergence behavior of populations with different sizes.**

***Index Terms*—multi-objective biochemical optimization, population size, landscape analysis, multi-parent recombination.**

I. Introduction

A customized Non-dominated Sorting Genetic Algorithm (NSGA-II) has been evolved with a considerable low number of generations and population size, termed early convergence for the molecular optimization of peptide sequences [1] [2]. Small peptides are of special interest in the area of drug design as they have some favorable features like conformational restriction, membrane permeability, metabolic stability and oral bioavailability [3]. Nevertheless, for this purpose these peptides have to optimize several molecular features at the same time. As both the synthesis and the laboratory characterization of peptides is very cost-intensive [4], moGAs provide an economical and robust method for peptide identification.

The NSGA-II is customized with regard to the encoding and the components mutation, recombination and selection. Different mutation and recombination methods have been evolved for this purpose and are introduced in [5][6]. These components and their parameter are not only inter related, but are also responsible for the performance of a GA. So far, less work has been done to gain an insight in the influence of the population size on the performance and in the interdependence with the selection operator and its parameters in the case of moGAs. The population size is an important value in influencing the performance of evolutionary algorithms [7]. Small population sizes tend to result in poor convergence and large populations extend the computational complexity of a GA in finding high quality solutions [8]. Therefore, an adequate population size that results in good performance is challenging. Diverse results have been presented regarding the choice and the handling of the populations size for single-objective GA: Yu et. al [9] study the connection between selection pressure and population size and ratify the concept of interdependence of parameters and operators in GA. The concept of self-adaption is used to overcome the problem of determining the optimal population size. Two forms of self-adaption are used: First, Bäck et al. [10] uses self-adaption as a previous setup and configuration step for evolutionary strategies. The population size then remains the same over all iterations. Second, Arabas et al. [11] introduces a GA with varying population size. The self-adaption of the population size is used throughout the whole GA run and depends among others on different parameters like the reproduction ratio. Eiben et al. [12] provide empirical studies that self-adaption of selection pressure and population size is possible and further rewarding regarding algorithm performance. In this case study, the global parameters tournament size and population size are regulated.

Several works have been proposed studying the effect of different numbers of parents for recombination in a range of 2 up to 10 parents in Evolutionary Algorithms (EA) e.g., [13][14][15], among others. These studies show that the optimal number of parents for recombination depends on the optimization problem as well as on the recombination method. Eiben represents in [13] a very extensive series of tests - in total 23000 test runs. In most of these cases the largest algorithm performance improvement can be obtained when the number of parents is increased from 2 to 3 parents. The experiments support two kinds of conclusions: Firstly, increasing the number of parents improves the performance continuously, but the degree of improvement is decreasing. Secondly, the performance improves by increasing the parent number until a certain parent number and decreases or oscillates afterwards. Another effect of a larger parent number is that less information are inheritance of the same solution

and the generated offspring is more different from its parents.

The questions that we consider in this paper are:

1) Do large populations speed up the convergence behavior of the customized NSGA-II for a three-dimensional biochemical minimization problem?
2) Is there a predictable impact between population size and selection?
3) Is there a range of population size, which is able to perform well?
4) Do a variation of the parent number within the recombination procedure influence the algorithm performance?

The questions 1.-3. have been investigated and answered in [1] on a three-dimensional biochemical minimization problem and is further part of this work for comparison and completeness. The following question 5 is logical consequence and in the focus of this work:

5) Are the results of question 2.-4. transferable from the three-dimensional to the four- dimensional biochemical minimization problem?

These questions are answered in an empirical way: The performance of the customized NSGA-II is assessed regarding its early convergence and a high diversity within the solutions. Some metrics have been proposed to evaluate the convergence behavior of a moGA [16]. These metrics, generally, measure the distance of non-dominated solution sets to the true Pareto front [16]. This makes a comparison of generations with different sizes impossible. Therefore, a convergence indicator is introduced especially for the comparison of the generations with different sizes based on the hypervolume. The favorable features of this indicator are also discussed. A landscape analysis is performed to determine the characteristic properties of the biochemical landscape and to gain an insight into the difficulties of the optimization problems. Furthermore, we will discuss available open source Java tools that allow an easy implementation of the customized NSGA-II to solve multi-objective biochemical optimization problems.

The remainder of this paper is organized as follows: Section II describes the components of the customized NSGA-II. Section III provides a comparison of open source Java frameworks focused on a most simple implementation of the customized NSGA-II. Section IV presents a review on landscape analysis methods and the results of the landscape analysis performed on the biochemical objective functions. Section V introduces the new convergence metric and discusses the motivation for its evolution and the indicator features. Section VI provides the performance results of the configurations with different population sizes and multi-parent recombination assessed on the three- and four-dimensional optimization problem. Furthermore, this section responses the questions raised in this section. Section VII provides the conclusion of this work and gives an outlook on the future work.

II. The customized NSGA-II for peptide optimization

In this section, the customized NSGA-II is described as used in the presented experiments. In the previous work [2][5], we have assessed the performance and interaction of different recombination and mutation operators. In these experiments, we have determined the optimal onset of recombination and mutation method that is used within the following experiments. Additionally, we have customized the encoding and selection for the purpose of peptide optimization. The procedure of the customized NSGA-II corresponds to the procedure of the traditional NSGA-II [5] and is depicted in Fig. 1:

At first, the procedure initializes the start population with

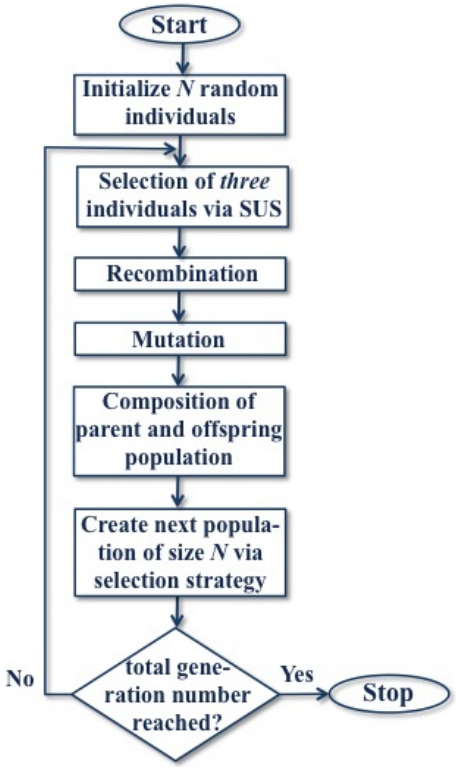

Figure 1. The procedure of the customized NSGA-II

a size of N. The main loop of the customized NSGA-II starts with the loop comprising the selection of the parent individuals for recombination and mutation via Stochastic Universal Sampling (SUS) based on roulette wheel selection: In each selection step, k individuals are selected via SUS to create k offsprings by recombination and mutation. This loop is repeated until the offspring population consists of N individuals. Then, the start and offspring population are shuffled together and the next population is created via a selection strategy. In the case, that the total number of generations is achieved, the main loop stops, otherwise it is repeated. The components of this procedure are motivated and described in detail below.

A. The encoding

The individuals are encoded as 20-character strings symbolizing the 20 canonical amino acids. The 20 characters are adopted of the single-letter code for amino acids. This encoding is firstly motivated by the idea of a most intuitive way of peptide encoding. Secondly, several tools predicting physiochemical or structural peptide properties make use of this encoding regarding the input data (e.g., see [17][18]). This avoids the data transformation before every fitness function evaluation.

B. The molecular fitness functions

Four molecular fitness functions are proposed to constitute the benchmark problems. These functions are selected under the aspect of predicting physiochemical properties, which are of importance for drug design [19]. Therefore, this combination of fitness functions allows conclusions on a range of important peptide properties [20]. These functions are associated to the three structural levels of peptides: The first two physiochemical functions refer to the primary structure. The third provides information about the secondary structure and the last one makes use of the primary structure of a peptide to provide information about a possible early tertiary structure disruption or an inadequate folding.

1) Molecular Weight (MW): The first fitness function is the calculation of the MW that is an important peptide feature for the purpose of drug design [3] and refers to the primary structure of a peptide. This fitness function is selected from the open source library BioJava, this library and a detailed description of this function are provided on the homepage [17].

2) Hydrophilcity (hydro): The second fitness function is the determination of the hydrophilicity (hydro) of a peptide. A hydrophilicity value is assigned to each peptide via the hydrophilicity scale of Hopp and Woods with a window size of the peptide length [21]. This fitness function refers also to the primary structure.

3) Needleman-Wunsch Algorithm (NMW): The third fitness function determines the optimal global similarity score provided by NMW [22] that is also part of the BioJava library [17]. The motivation for NMW is the identification of similarities between peptides regarding biochemical functionality and structure via a global sequence alignment to a pre-defined reference peptide. NMW makes use of a scoring model and in this case the BLOcks SUbstitution Matrix (BLOSUM) [18] in form of the percentage identity 100 (BLOSUM100) is used.

4) Instability Index (InstInd): The last fitness function is the InstInd and is used to analyze the primary structure of peptide sequences to predict a potential intracellular instability of peptides. This function is also provided by the BioJava library and a description is also given on the homepage [17]. These four fitness functions act comparatively: The fitness values of an individual are determined by the difference between the fitness function values of this peptide to the fitness function values of a predefined reference-peptide. Therefore, these four objective functions have to be minimized.

C. The recombination operator

In the previous work [5], different recombination operators are benchmarked on a three-dimensional molecular minimization problem. The linear $n-$point recombination operator achieved the best performance, where the number of recombination points n are determined by a linearly decreasing function:

$$x_R(t) = \frac{l}{2} - \frac{l/2}{T} \cdot t, \tag{1}$$

which depends on the peptide length l, the total number of the GA generations T and the index of the actual generation t. The motivation of this recombination operator is a preferred high explorative search behavior in the early generations and a high motifs-maintaining encouraging the local search in later generations. For this purpose, the number of recombination points in the first generation is $l/2$ and decreases linearly until one recombination point in the last generation. The recombination points themselves are determined randomly. One recombination point in the last generation guarantees a motif preservation of at least 50% of the peptide sequences. The recombination operator is usable as multi-parent recombination, where the default number of parent is three according to the results of Eiben [13]. The impact of the number of parents on the multi-dimensional biochemical minimization problem is challenging and in the focus of the following experiments.

D. The mutation operator

In combination with LiDeRP, an adaption of the deterministic dynamic operator of Bäck and Schütz revealed the best performance as reported in [5]. This mutation operator is motivated by the idea that a high number of mutations in the early generations provides a good exploration, whereas a low number of mutations in later generations leads to good exploitation. The mutation rates are determined via the decreasing function

$$p_{a,BS} = (a + \frac{l-2}{T-1}t)^{-1}, \tag{2}$$

with $a = 2$, l describes the peptide length, T the total generation number of the GA and t the index of the actual generation number. This decreasing function has been adapted to a lower start mutation rate as a high start mutation rate as well as a high start recombination rate results in a too high exploration in the early generations.

E. The Aggregate Selection

The flow diagram in Fig. 2 depicts the selection methods. The Aggregate Selection is tournament-based. From the tournament set individuals are chosen from the first front with a probability p_0 and with a probability $1-p_0$ the individuals are chosen via Stochastic Universal Sampling (SUS). The number N of pointers is the number of fronts and the segments are equal in size to the number of individuals in each front.

Therefore, the selection method has two parameters, the tournament size and the probability of choosing individuals

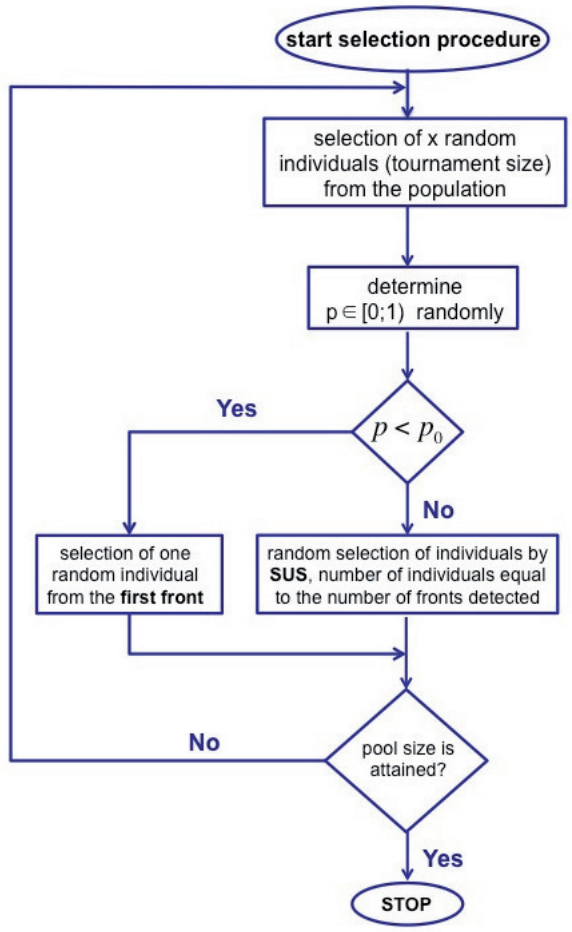

Figure 2. Aggregate selection strategy

from the first front. The tournaments size of 10 has proven to be an optimal choice. The parameter p_0 is challenging regarding the population size.

III. Open source Java frameworks

In this section, we summarize and describe different open source Java tools that provide Genetic Algorithm implementations. The summarization is focused on Java frameworks for a most simple implementation of BioJava, which provides several physiochemical properties via APIs. The main goal of this framework analysis is the selection of a tool that allows an easy implementation of the proposed customized NSGA-II.

The framework Java API for Genetic Algorithm (JAGA) in its current version 1.0 beta is a research tool developed and supported by the Computer Science Department of the University College London [23]. This tool does not include any moGAs, but it provides a protein string sequence encoding using 20 different characters symbolizing the 20 canonical amino acids. Among others, eight different physiochemical properties like hydrophobic, aliphatic, aromatic and polar are pre-defined for each canonical amino acid. In addition, it contains for each genotype a parameter-depending crossover and mutation method and allows a peptide or protein representation by the pre-defined amino acid patterns. The user interested in a moGA application has to extend this tool for this purpose, but the amino acid character encoding is a clear benefit. Other useful functions are the opportunity of creating a random initial population of protein sequences and the implementation of the Needleman-Wunsch or Smith-Waterman Algorithm.

The framework Metaheuristic Algorithms in Java (jMetal) in its current version 4.3 is an extensive and complex tool especially for moGA applications [24]. It contains beneath NSGA-II the moGA variants: Pareto Envelope-based Selection Algorithm (PESA), improved Strength Pareto Evolutionary Algorithm (SPEA2), improved PESA (PESA2), S-Metric Selection Evolutionary Multiobjective Evolutionary Algorithm (SMS-EMOA), Indicator-Based Evolutionary Algorithm (IBEA) and Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D). Further, different variation operators are implemented like single-, two- point, Simulated Binary Crossover (SBX) and polynomial, uniform and swap mutation. 'Ranking&crowding selection' is included as the traditional NSGA-II selection method as well as tournament and PESA2 selection. Additionally, jMetal provides several established metrics to evaluate the performance like the hypervolume, Inverse General Distance (IGD), General Distance (GD) and a measure for diversity. A definite advantage of jMetal is the intuitive and clear program construction, which allows an easy algorithmically extension. The disadvantage is a missing character or string encoding.

The framework Java-based Evolutionary Computation Research System (ECJ) in its current version 21 is comparable with jMetal in the issues functional complexity and potential extension. ECJ is developed at George Mason University's Evolutionary Computation Laboratory [25]. It includes the moGAs NSGA-II and SPEA2. Furthermore, different vector representations with corresponding variation operators are included as well as SUS and tournament selection, among others. Moreover, it proposes the potential to read populations from files. It does not provide an intuitive and clear program structure like jMetal.

The Multi-Objective Evolutionary Algorithm framework (MOEA framework) in the current version 2.1 is a Java framework for multi-objective optimization [26]. It provides a very wide range of MOEA variants as it includes the jMetal library in the version 4.3. Therefore, MOEA framework has the same features like jMetal regarding the benchmark problems, performance metrics and available variation operators. The MOEA provided by jMetal also only support binary, real-values and permutation encoding. On the other hand, the MOEA framework allows a new genotype implementation.

Evolutionary Algorithms workbench (EvA2) is a Java framework developed by the department of computer science at the Eberhard Karls University in Tübingen [27]. It is not only intended for research, but is also deployed for industrial applications and is available under LGPL license. Its specificity is its easy-to-use graphical user interface and provides a MATLAB interface to optimize functions in MATLAB

with standard algorithm implementations in EvA2. It also has a client-server structure and provides NSGA-II, PESA and SPEA2 as moGA implementations. A string or character encoding is not implemented and an implementation afterwards is challenging, because encoding affects all parts of the toolbox.

The framework Java Class library for Evolutionary Computation (JCLEC) in the current version 4 includes the evolutionary features NSGA-II and SPEA2. It proposes different encodings with various variation operators except string or character encoding, but provides an expendable program structure. Further, JCLEC includes fitness proportionate selection strategies like tournament and SUS.

The modular framework for metaheuristic optimization (OPT4J) contains a set of multi-objective optimization algorithms including SPEA2 and NSGA-II [28]. OPT4J has been evolved under two aspects: A simple evolutionary optimization of user-defined problems and the potential of an arbitrary optimization algorithm implementation. Further, the common benchmark problems ZDT, DTLZ, WFG and Knapsack problem are available as well as the genotype encoding binary, integer, real-values and permutation. Unfortunately, the module-based structure and the use of the GUI makes an extension for the purpose of implementation of the proposed customized NSGA-II more complicated. A special feature is the graphical visualization of the optimization process regarding the convergence and potential of a Pareto plot.

Fig. 2 gives an overview of the reviewed Java frameworks. These frameworks are compared under the aspects of: (i) configuration of a character or string encoding as an option, (ii) an implementation of NSGA-II, (iii) potential of a simple extension, and (iv) an intuitive program structure according to the moGA components.

TABLE I
OVERVIEW OF THE SPECIAL FRAMEWORK ASPECTS

	JAGA	jMetal	MOEA	ECJ	EvA2	JCLEC	OPT4J
(i)	x						
(ii)		x	x	x	x	x	x
(iii)		x	x	x		x	
(iv)		x	x				

Table I reveals that none of the open source Java frameworks attains all required aspects in an adequate level. As a consequence, the proposed customized NSGA-II is implemented within jMetal. MOEA framework is a possible alternative as it provides the jMetal library. Nevertheless, some programming effort is necessary regarding this implementation. Furthermore, the protein string sequence encoding of JAGA serves as a model to the targeted peptide encoding. The experiments of the four-dimensional optimization problem are performed with an implementation of the proposed customized NSGA-II in jMetal, the experiments on the three-dimensional optimization problem have been performed with an own implementation as presented in [1].

IV. MOLECULAR LANDSCAPE ANALYSIS

Fitness landscape analysis is a common and important methodology to gain an insight in the complexity and difficulty of an optimization problem with the aim of designing a search algorithm with optimized performance [29]. The components of a landscape are the configuration set X of all feasible solutions of the optimization problem. According to the organization of X, a notation on neighborhood, nearness distance or accessibility on X is required. The third and essential component are the fitness functions $f : X \to \mathbb{R}$, [30].

The aim of the landscape analysis is the investigation of the landscape structure and the determination of the landscape characteristics that have a strong influence on the search algorithm e.g., see [31]:

- **Modality:** The modality provides the tendency of the fitness landscape to produce local optima. Therefore, the number and distribution of local optima are indicators for the modality.
- **Correlation:** This is an indicator for the dependence between two solutions of X. In the area of multi-objective landscapes, the correlation is of particular interest as it provides information about the actually optimization problem dimension and therefore about the problem difficulty. In the single-objective case, the autocorrelation is commonly used as an indicator for fitness diversity.
- **Ruggedness:** This is a feature describing the fitness variation between the fitness values of a solution and its neighbors. The modality and the correlation provide hints for the level of ruggedness.
- **Plateaus:** These are areas referring to neutrality, constituted by a set of solutions with equal fitness values. The size of these areas and the probability of existence are used for description.

Different techniques have been proposed to analyze the characteristic features of a fitness landscape. These techniques are classified into two categories both based on solution sequences obtained by random walks: The statistical analysis and the information analysis. [31]

Statistic analysis techniques investigate the fitness landscapes in a qualitative manner. Therefore, several correlation metrics have been proposed to measure the ruggedness of a landscape. Weinberg proposed the random walk correlation function $r(s)$ that measures the autocorrelation between two sets of fitness points separated by s solutions [32]:

$$r(s) = \frac{\sum_{i=1}^{n-s}(f_i - \bar{f})(f_{i+s} - \bar{f})}{\sum_{i=1}^{n}(f_i - \bar{f})^2}, \qquad (3)$$

where f_i is the fitness function value of the random walk solutions $\{f_i\}_{i=1...n}$ and $\bar{f}$ is the average fitness value of all solutions. High autocorrelation values indicate that the fitness values are similar and the landscape is less rugged. Otherwise, a small autocorrelation value indicates that the fitness values are uncorrelated and the landscape is rugged. Another correlation metric based on the autocorrelation is the correlation length that measures the distance beyond which

two sets of fitness points becomes uncorrelated [31]:

$$l = -\frac{1}{ln|r(1)|}, \tag{4}$$

with $r(1) \neq 0$. The higher the correlation length the smoother is the landscape.

Jones proposed the Fitness Distance Correlation (FDC) [33]. This coefficient measures the relation of the fitness values and the distance of the solutions s_i to the nearest optimum x^* in the search space:

$$FDC = \frac{cov(f(s_i), d(s_i))}{\sqrt{var(f(s_i)) \cdot var(d(s_i))}}, \tag{5}$$

where d is the distance function to x^* and $cov(x, y)$ as well as $var(x)$ are the common statistical measures covariance and variance. The coefficient values are in the interval $[-1; 1]$. Jones further introduced a prediction of these values regarding the GA effectiveness in solving the optimization problem:

- $FDC \geq 0.15$: The fitness increases with the distance. The GA is potentially not effective or the problem is misleading.
- $-0.15 < FDC < 0.15$: There is virtually no correlation between fitness and distance. The problem is categorized as difficult.
- $FDC \leq -0.15$: The fitness increases as the optimum approaches. The GA is potentially effective or the problem is straightforward.

A great disadvantage of FDC is that the nearest optimum or at least the best-known solution has to be known in advance. Compared to the statistical analysis, the information analysis is a quantitative investigation and provides more detailed information about the landscape structure like a measurement of the optima density and plateaus as well as the degree of the random walk regularity [34]. Vassilev et al. [34] provides three information measures to determine the modality and the level of ruggedness. For each of these three measures, the random walk path $\{f_t\}_{t=0...n}$ is transformed into a string $S(\epsilon) = s_1 s_2 ... s_n$ with $s_i \in \{-1, 1, 0\}$, where

$$s_i = \begin{cases} -1 & \text{if } f_i - f_{i-1} < -\epsilon \\ 1 & \text{if } f_i - f_{i-1} > \epsilon \\ 0 & \text{if } |f_i - f_{i-1}| \leq \epsilon \end{cases} \tag{6}$$

and $\epsilon \in [0; l]$, where l is the maximal difference between two fitness values. The indicator is more sensitive to the steps of the random walk the smaller the value for ϵ. Then, the Information Content is defined via:

$$H(\epsilon) = -\sum_{p \neq p} P_{[pq]} log_6(P_{[pq]}), \tag{7}$$

where $p, q \in \{-1, 1, 0\}$, $P_{[pq]} = \frac{n_{[pq]}}{n}$ are the probabilities presenting frequencies of possible blocks pq and $n_{[pq]}$ is the number of occurrences of the blocks pq in $S(\epsilon)$. The base of the logarithm is chosen as 6. This is the number of all possible blocks pq. The information content depends on the parameter ϵ that is responsible for a more global or local view on the random walk according to the magnitude of ϵ. The Partial Information Content is a measure for the degree of ruggedness. The string $S(\epsilon)$ is transformed into a string $S(\epsilon')$ by deleting the elements 0 and blocks of equal elements are reduced to only one of these elements. The partial information content is defined by:

$$M(\epsilon) = \frac{v(\epsilon)}{n}, \tag{8}$$

where $v(\epsilon)$ is the length of $S'(\epsilon)$ and n the length of $S(\epsilon)$. Furthermore, $v(\epsilon)$ indicates the number of extrema along the landscape path. In the case $M(\epsilon) = 0$, the landscape path is nearly flat or monotonously increasing or decreasing. Otherwise, $M(\epsilon) = 1$ indicates that the landscape path is maximal rugged.

The Information Stability as the third indicator for information analysis proposed by Vassilev is an indicator for the highest difference between neighboring points in the landscape path. The information stability is defined as the smallest value of ϵ for which the landscape path becomes flat. In this case, the string $S(\epsilon)$ comprises only zeros.

Another information indicator was proposed by Leier et al. [35]. This indicator gives information about the density as well as length of flat areas. Therefore, it is an indicator for the ratio between flat and smooth parts of a landscape path and therefore an optimal measure for neutrality. It is defined as:

$$h(\epsilon) = -\sum_{p \in \{-1,1,0\}} P_{[pp]} log_3(P_{[pp]}), \tag{9}$$

where $P_{[pp]}$ is the frequency of blocks pp in $S(\epsilon)$.

A. Landscape analysis of the molecular fitness functions

The configuration set X of the molecular fitness landscapes are all feasible peptides of the length 20 consisting of the 20 canonical amino acids. Therefore, the landscape is of a high complexity 20^{20}. Furthermore, the search space is discrete as there are real-valued solutions that have no corresponding feasible peptides in the search space. The neighborhood of a solution is defined by all peptides differing to this solution in one amino acid in exactly one position [36]. Therefore, the move operator of the random walk is the mutation operator that changes one amino acid in a position in the solution. The mutation of an amino acid in the way that the same solution is achieved is excluded to prevent the random walk from stagnation.

The four fitness landscapes NWM, MW, hydro and InstInd are analyzed according to the important landscape properties: Modality, correlation and ruggedness. The basis of the landscape analysis is random walks of 100 steps that are repeated 30 times for statistical reasons. The starting solution of the random walk is initialized randomly.

In Fig. 3-6, six random walks are exemplarily depicted of the molecular fitness functions for a first global view on the landscapes. All four molecular fitness functions provide large variations of the fitness values over the 100 random walk steps and therefore indicate rugged landscapes. From

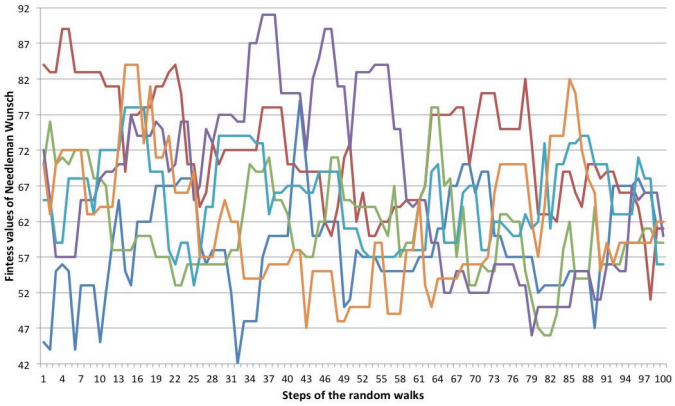

Figure 3. Exemplary presentation of six time series of the molecular fitness function NMW.

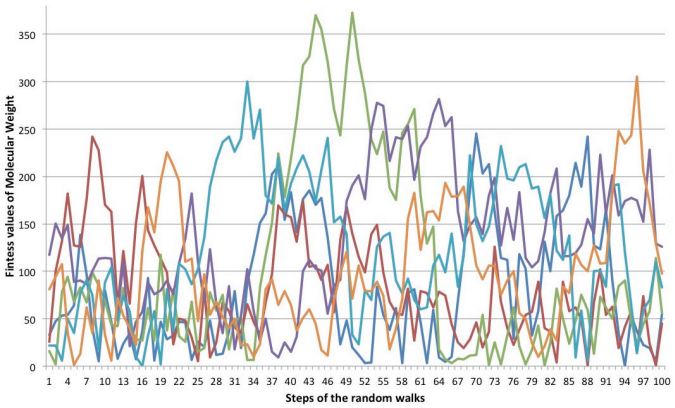

Figure 4. Exemplary presentation of six time series of the molecular fitness function MW.

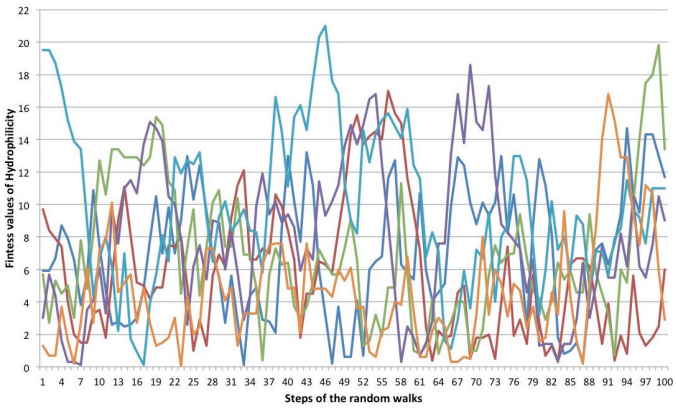

Figure 5. Exemplary presentation of six time series of the molecular fitness function hydro.

this global point of view, NMW is the only function revealing some plateaus or flat areas over two to five random walk steps (Fig. 3). The InstInd function also reveals some flat areas and plateaus, but in a lesser extent and averaging over a lower number of random walk steps (Fig. 6). The fitness values of the MW function are scaled by a factor of 10 and achieved some large jumps of the fitness values as well as some areas with

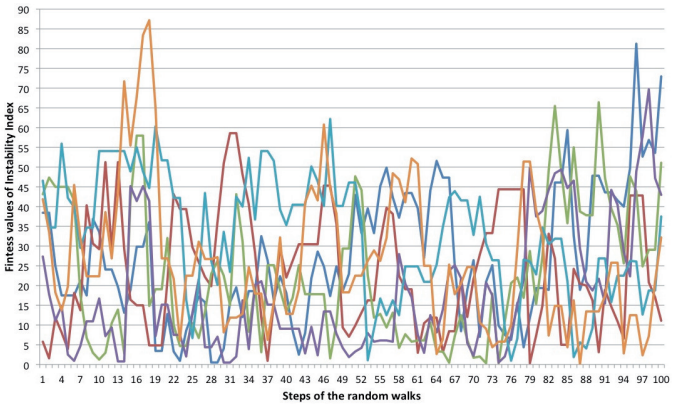

Figure 6. Exemplary presentation of six time series of the molecular fitness function InstInd.

oscillating parts (Fig. 4). The hydro fitness function appears similar to MW regarding the jumps and the oscillating parts (Fig. 5). Otherwise, it also reveals some isolated flat areas or plateaus. To quantify the rugged landscape properties of these four fitness functions, the autocorrelation of all solution (100 random walks repeated 30 times) is calculated after the model of Lee [36], which is an adaption of the autocorrelation function of Weinberg (eq. 3) by determining the average value and the standard deviation of all solutions and applying the average value on the starting point of the random walks:

$$p_s = \frac{\frac{1}{n+1}\sum_{i=0}^{n}(x_{i0}-\mu)(x_{is}-\mu)}{\sigma^2}, \tag{10}$$

where μ is the average value calculated by

$$\mu = \frac{1}{n}\sum_{i=1}^{n} f(x_i), \tag{11}$$

σ is the standard deviation determined by

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i)-\mu)^2} \tag{12}$$

and x_{i0} is the starting point and x_{is} is the $s-$step of the i-th random walk. The self-correlation value of the starting point p_0 has to be 1 or at least approximately 1. This adaption is motivated by the fact that the random walk length of 100 is relatively small compared to the search space complexity, which empirically leads to a time-varying volatility - meaning, the average value and the standard deviation are very different between the random walks [36]. The autocorrelation of the time series of the four molecular fitness functions depicted in Fig. 7 confirm the time-varying volatility as the values for p_0 are differing from the value 1. The high ruggedness of the four molecular landscapes is visible by the fast decrease of the autocorrelation values to 0 within the first 20 random walk steps. Furthermore, most of the autocorrelation values of all four molecular functions are in the range of -0.3 to $+0.3$, which indicate a weak correlation. The autocorrelation values

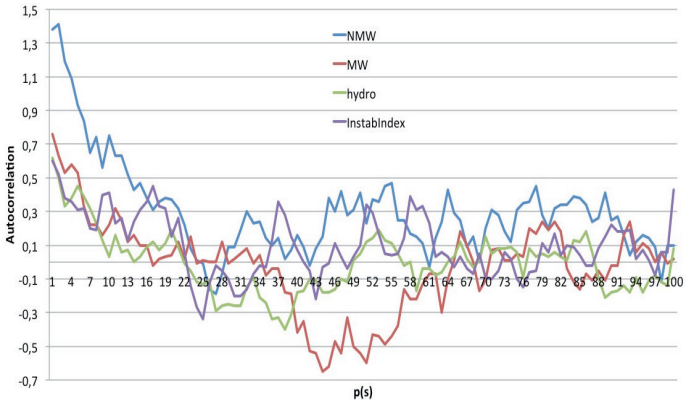

Figure 7. Autocorrelation functions of the random walks on the four molecular landscapes: NMW, MW, hydro, InstInd

of MW indicate a moderate autocorrelation between the random walk steps 40 to 57. Similarly, the autocorrelation values of NMW reveal some very moderate correlations between the random walk steps 45 to 88. Some outlier of very moderate correlations exists also for hydro and InstInd.

In addition to the autocorrelation, the correlation between the molecular fitness functions is also of great interest regarding the combination of these four molecular functions to a Multi-Objective Problem (MOP) as the high correlation between two time series of different fitness functions theoretically reduce the optimization problem dimension and makes the MOP less challenging. The correlation matrix indicates s potential linear relationship between different functions:

$$M_{corr} = \begin{pmatrix} 1 & corr(f_1, f_2) & \dots & corr(f_1, f_k) \\ corr(f_2, f_1) & 1 & \dots & corr(f_2, f_k) \\ \vdots & \vdots & \ddots & \vdots \\ corr(f_k, f_1) & corr(f_k, f_2) & \dots & 1 \end{pmatrix}, \quad (13)$$

where M_{corr} is symmetrical and consists of the Pearson correlation coefficients of the fitness function f_i and f_j:

$$corr(f_i, f_j) = \frac{\sum_{i=0}^{n}(f_i - \bar{f}) \cdot (f_j - \bar{f})}{\sigma_{f_i} \cdot \sigma_{f_j}} \quad (14)$$

Correlation values of $|corr(x, y)| < 0.3$ indicate a weak correlation between x and y, $0.3 \leq |corr(x, y)| \leq 0.8$ indicates a moderate correlation and $|corr(x, y)| > 0.8$ indicates a high linear correlation. The correlation matrix for the four molecular functions NMW (f_1), MW (f_2), hydro (f_3) and InstInd (f_4) according to eq. (13) is given by:

$$M_{corr} = \begin{pmatrix} 1 & 0.047 & 0.252 & 0.09 \\ \cdots & 1 & -0.014 & -0.032 \\ \cdots & \cdots & 1 & -0.266 \\ \cdots & \cdots & \cdots & 1 \end{pmatrix}. \quad (15)$$

This matrix is calculated of the 30 random walks consisting of 100 steps for each molecular function. The matrix entries reveal that there is no linear relationship between the time series of each two fitness functions: There is a weak relationship between NMW and MW (eq. (15): $corr(f_1, f_3) = 0.252$) as well as InstInd and hydro (eq. (15): $corr(f_3, f_4) = -0.266$) and no correlation between the other combinations.

Another important landscape property investigated in the following is the modality. The examination of the single molecular fitness functions according to the local optima is not advisable for the purpose of a MOP as the most of the local optima of the single functions are no optima in the sense of the MOP [13]. The optima in the multi-objective sense are the non-dominated solutions.

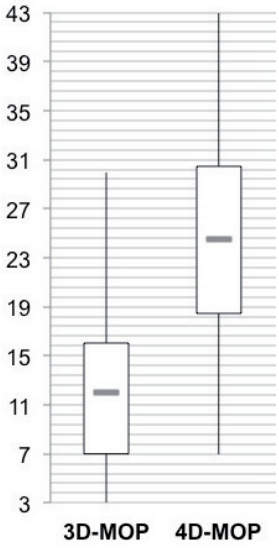

Figure 8. Number of the non-dominated solutions achieved by the random walks on the 3D and 4D molecular landscapes

Fig. (8) depicts the optima or non-dominated solution density of the 3D-MOP composed of NMW, MW and hydro as well as the 4D-MOP composed of NMW, MW, hydro and InstInd. In general, the 4D-MOP reveals nearly 50% more non-dominated solutions than the 3D-MOP within the 30 random walks of 100 steps. Further, the range of the non-dominated solutions achieved within the random walks is higher in the case of the 4D-MOP. As a consequence, the non-dominated solution density is higher in the case of the 4D-MOP and the corresponding landscape more rugged. Further, the 4D-MOP is more difficult for the customized NSGA-II then the 3D-MOP because of the MOML structures: The 3D-MOP has a higher front diversity and provides less solutions in the optimal front than the 4D-MOP.

Plateaus in this area of multi-objective real-valued landscapes are identified by consecutive equal fitness values for each molecular function. In the 30 random walks of the 3D-MOP, 20 plateaus have been identified: Two plateaus of each two consecutive equal fitness values have been identified in five random walks, a plateau of three consecutive equal fitness values have been found in one random walk and the remaining 9 plateaus have been identified in different random walks each consisting of two consecutive equal fitness values. In the case of the 4D-MOP, 8 plateaus have been identified: These plateaus consist only of two consecutive fitness values

and only one random walk achieved two of these plateaus. Consequently, the 3D-MOP reveals statistically more plateaus than the 4D-MOP and a local search is of a greater interest in this case.

Transferring these results of the molecular landscape analysis on design considerations of a metaheuristic allows the conclusion that the search process has to be guided in direction of at most optimal solutions, which are spread over the landscape. Then, the neighborhood has to be searched for further high quality solutions. Transferring these results on the explorative- exploitive balance of the metaheuristic associates that a high exploration of the search process in the early generations and a more exploitive search behavior in the later generations is beneficial for such rugged landscapes with statistically only a low number of flat regions.

V. Evaluation measures for convergence and diversity

Firstly, the convergence measure is introduced, which has been especially evolved to evaluate generations with different sizes. Subsequently, the features of this indicator are discussed followed by the presentation of measurement for diversity.

A. Introduction of the average cuboid volume

In the past, several metrics have been proposed to evaluate the convergence behavior of populations produced by a moGA. Usually, they act on the distance of the non-dominated solution set of a generation to the true Pareto front. The hypervolume or the S-metric measures the overlapped space of the non-dominated solution set to a predefined anti-optimal reference point [37]. The hypervolume is a very established convergence metric with its favorable mathematical properties as one reason. Another convergence metric is the D-metric [16]. The D-metric makes use of the hypervolume and calculates the coverage difference of two solution sets. A reference set is needed to assess the convergence to the true Pareto front. The C-metric is an appropriate measure to compare the dominance of two Pareto optimal sets [37]. The Error Ratio (ER) is a percentage measure for the number of solutions in a set that are to be found on the true Pareto front [16]. GD is a measure of the average distance between a Pareto optimal solution set to the true Pareto front [38]. It includes the minimal component-wise distance of a solution set to the nearest one on the true Pareto front. The convergence metric of Deb also measures the distance between a solution set and a reference set of the Pareto front [39]. It calculates the average normalized distance for all solutions in the solution set. A recently published convergence metric is the Averaged Hausdorff Distance Δ_p [40]. It is based on GD and the IGD [41].

The reasons for the evolution of a new convergence metric in this paper and in the scientific community in general are multiple: The disadvantage of the metrics D-metric, ER, GD, Δ_p and the convergence metric of Deb is their dependency on the true Pareto front or at least a reference set of Pareto optimal solutions that are usually unknown in the case of real-world MOPs. Furthermore, these metrics are not useful indicators for an entire ranking between generations of different sizes. However, the populations in moGAs are generally limited in size. From a more global point of view, the evaluation and comparison of the global convergence behavior of whole populations - not only the non-dominated solution set of a generation - is required with respect to the influence of the population size or the selection pressure.

For this purpose, a new metric is presented that reflects the convergence behavior of a whole population and is a 'fair' indicator for comparison of generations of different sizes. This Average Cuboid Volume (ACV) is evolved according to the model of the hypervolume. The motivation for the exploitation of the hypervolume model is to profit from its preferable properties as mentioned above. The benefit of this new metric compared to the hypervolume is the low computational complexity as no point ordering is required.

In the following, we assume that the underlying optimization problem is to minimize. The metric calculates the average cuboid volume of the cuboids spanned by the solution points to a pre-defined reference point r:

$$ACV(X) = \frac{1}{n}\sum_{i=1}^{n}\left(\prod_{j=1}^{k}(x_{ij} - r_j)\right), \tag{16}$$

where n is the population size, k is the number of objectives, x_i are the solutions on the population X and x_{ij} is the $j-th$ component of a solution x_i. It holds $(x_{ij} - r_j) > 0$ as the predefined reference point is chosen as the theoretical minimal limit of the true Pareto front. The lower the indicator values the more positive is the global convergence behavior as the reference point is chosen as a theoretical optimal point.

Obviously, ACV is not a suitable indicator to compare the experimental results of different dimensional optimization problems.

B. Discussion of the average cuboid volume

The question regarding the suitability of a metric for evaluation depends on the intention of the investigation object and the preferences. ACV is intended to evaluate the global convergence behavior of a whole population with the ultimate aim of comparing solution sets of different sizes according to the proximity to the true Pareto front.

The first expectation that is important for the use of ACV is that the convergence quality shall not change if the number of equally solutions increases. ACV does not fulfill this averaging strategy: Let $x \in \mathbb{R}^k$ be a solution of the optimization problem and $X = \{x\}$. Further, $Y = \{x, ...x\}$ is a set of n equally copies of the solution x, then

$$\begin{aligned} ACV(Y) &= \frac{1}{n}\sum_{i=0}^{n}\left(\prod_{j=1}^{k}(x_j - r_j)\right) = \frac{1}{n}\cdot n\prod_{j=1}^{k}(x_j - r_j) \\ &= \prod_{j=1}^{k}(x_j - r_j) = ACV(X). \end{aligned} \tag{17}$$

The second expectation is described by the following observation: An intuitive indicator reflecting the quality of approximation sets of different Pareto front refinements requires 'better' indicator values for the finest approximation set. This effect is demonstrated for ACV by an example also used in [13]:

Example 1: The Pareto front is given by the bounded convex function $f(x) = 1/x^2$ between the points $y_1 = (0.1, 100)$ and $y_2 = (1.1, 0.826)$ meaning

$$PF_{true} = \{(x, y)|y = 1/x^2 \text{ with } x \in [0.1, 1.1]\}. \quad (18)$$

We consider the following three approximation sets of increasing refinement of the Pareto front

$$Y_1 = \{(0.1 + 0.2 \cdot i, 1/(0.1 + 0.2 \cdot i)^2)|i \in \{0, 1, ..., 5\}\},$$
$$Y_2 = \{(0.1 + 0.1 \cdot i, 1/(0.1 + 0.1 \cdot i)^2)|i \in \{0, 1, ..., 10\}\},$$
$$Y_3 = \{(0.1 + 0.01 \cdot i, 1/(0.1 + 0.01 \cdot i)^2)|i \in \{0, 1, ..., 100\}\}.$$

Table I depicts the indicator values of ACV for the three approximation sets with the reference point $(0, 0)$.

TABLE II
ACV VALUES FOR THE APPROXIMATION SETS $Y_1 - Y_3$ WITH THE REFERENCE POINT $(0, 0)$.

X	Y_1	Y_2	Y_3
ACV(X)	3.13	2,75	2.43

The third preferable expectation of this indicator is the averaging effect. It is trivial that a dominating solution x yields better indicator values than the dominated one y, because $ACV(\{x\}) = \prod_{i=1}^{k}(x_j - r_j) < \prod_{i=1}^{k}(y_j - r_j) = ACV(\{y\})$. From this observation it can be interpreted that if one dominated solution x_1 in the solution set $X = \{x_1, x_2, ..., x_n\}$ is replaced by a dominating one $\bar{x}_1$, then $ACV(\{x_1, x_2, ..., x_n\}) > ACV(\{\bar{x}_1, x_2, ..., x_n\})$. The averaging effect of ACV is illustrated by the example, which has also been used for Δ_p [40]:

Example 2: The true discrete Pareto front is described by $P = \{p_i|p_i = (0.1 \cdot (i-1); 1-(i-1) \cdot 0.1) \text{ with } i = 1, ..., 11\}$. Two solution sets are given by $X_1 = \{x_{1,1}, p_2, ..., p_{11}\}$ and $X_2 = \{x_{2,1}, x_{2,2}, ..., x_{2,11}\}$ with the elements $x_{1,1} = (\epsilon, 10)$ and $x_{2,i} = p_i + (\frac{\epsilon}{2}, 5)$ with $i = 1, ..., 11$. For the outlier $x_{1,1}$ the values $\epsilon = 0.001$ is used for numerical evaluations. X_1 is a better approximation of the true Pareto front, but it contains the outlier $x_{1,1}$. On the other side, X_2 is close to the true Pareto front and the difference of each element to the Pareto front is less than the one of the outlier. As we are interested in an averaging effect, the indicator values of X_1 has to be better than the one of X_2. This is true for ACV as $ACV(X_1) = 0.15$ and $ACV(X_2) = 2.65$.

The use of $ACV(X)$ as a convergence and as a diversity metric is not within our preferences. $ACV(X)$ is not a reliable indicator for diversity. A solution set with clustered solutions does not always achieve better indicator values demonstrated in the following example:

Example 3: Once more PF_{true} is described by eq. (18) and the solution set

$$Y_4 = \{(0.29, 11.89); (0.3, 11.11); (0.31, 10.4); (0.32, 9.77); (0.33, 9.18), (0.34, 8.65)\}$$

contains clustered solutions on the true Pareto front, then $ACV(Y_4) = 3.18 \approx ACV(Y_1)$. Though the solutions of Y_4 are much more clustered than those of Y_1, Y_4 receive nearly the same indicator values than Y_1.

C. The diversity measure

The measure for diversity calculates the average distance of all pairs of solutions (see [5]):

$$\Delta = \sum_{i,j=1, i<j, i \neq j} \frac{|d_{ij} - \bar{d}|}{N} \quad \text{with } N = \binom{n}{2}, \quad (19)$$

where $d_{i,j}$ symbolizes the Euclidean distance of two solutions x_i and x_j, $\bar{d}$ is the mean of all measured distances and n is the population size.

VI. RESULTS AND DISCUSSION

In this section the simulation onset for the test runs are described. The results are further depicted and discussed.

A. Simulation onsets

The test runs are performed for different configurations. The configurations are composed of a different population size (30, 50, 70, 100, 130, 150) and the selection parameters $p_0 = 0\%, 30\%, 50\%$. These parameters have been emphasized by previous experiments. The selection parameter $p_0 = 0\%$ stands for SUS exclusively. Each multi-objective configuration is repeated 20 times until the 18th generation - for statistical reasons. The test runs are evaluated via the convergence indicator ACV and the diversity measure as introduced in the last section. ACV uses the theoretical minimal limit $(0/0/0)$ of the Pareto front as an optimal reference point. Therefore, a good performance is achieved if the ACV value is as low as possible and the diversity value is as high as possible. Boxplots are created for each configuration and for each objective of evaluation as boxplots provide a good overview of the location parameter as well as the spread. The values of ACV and diversity are scaled under the same criterion for a better graphical presentation. The figures are ordered according to the population size. The standard population size within the customized NSGA-II is 100 [2][5] (Fig. 12, Fig. 23, Fig. 24). Therefore, the results are discussed regarding an increase and a decrease of this size.

B. Experiments on the 3D-MOP

The 3D-MOP comprises the molecular objective functions NMW, MW and hydro. In general, a decrease of the population size down to 70 and 50 results in an increase of the ACV values and a decrease of the diversity values (Fig. 10, Fig. 11). This means that the convergence and the spread within the solutions is reduced caused by decreasing the population size. The ACV values decrease for a population size of 30

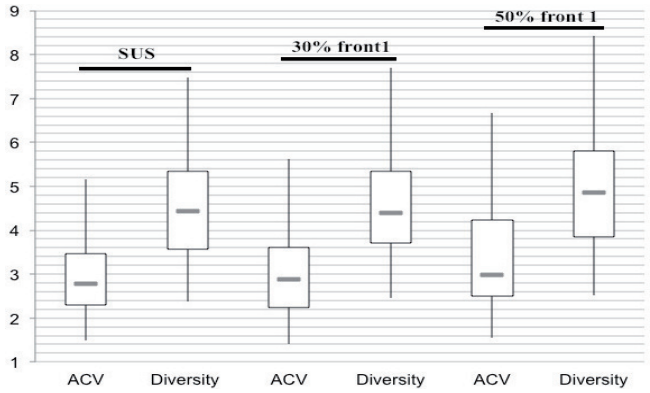

Figure 9. Population size 30

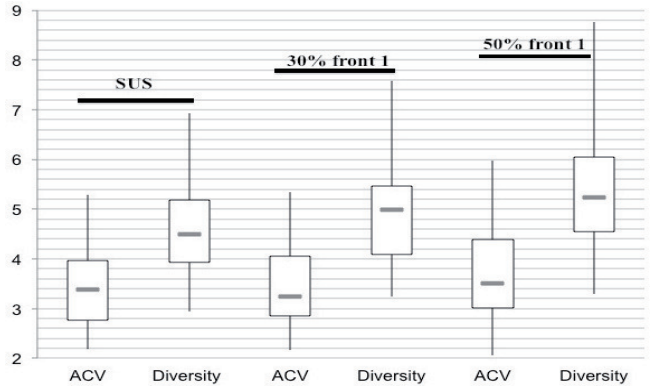

Figure 10. Population size 50

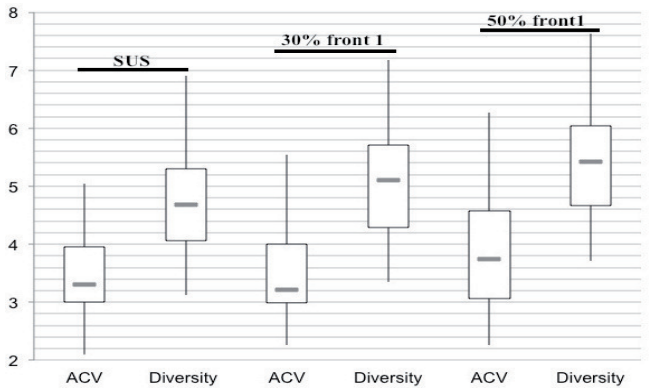

Figure 11. Population size 70

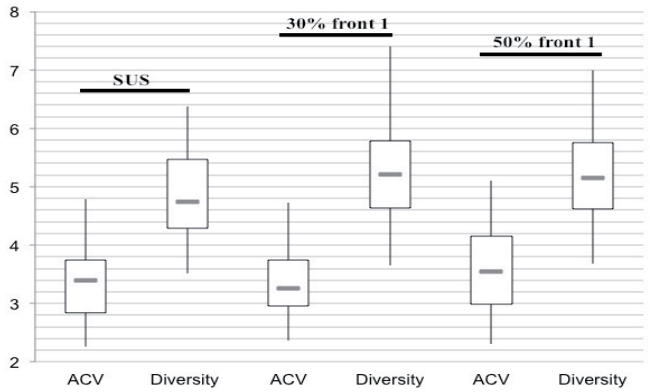

Figure 12. Population size 100

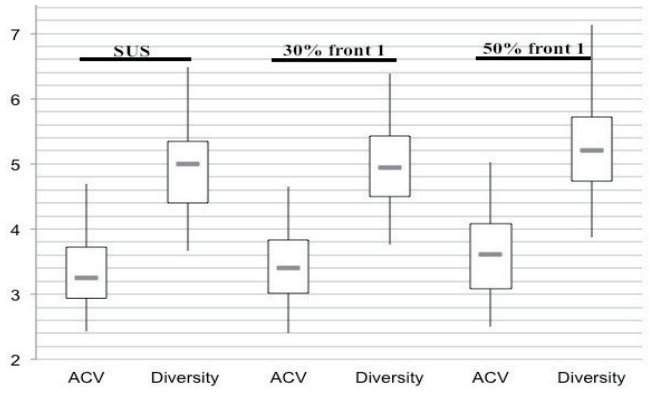

Figure 13. Population size 130

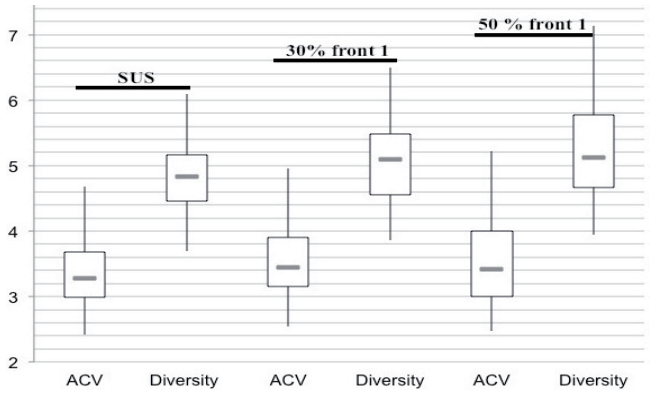

Figure 14. Population size 150

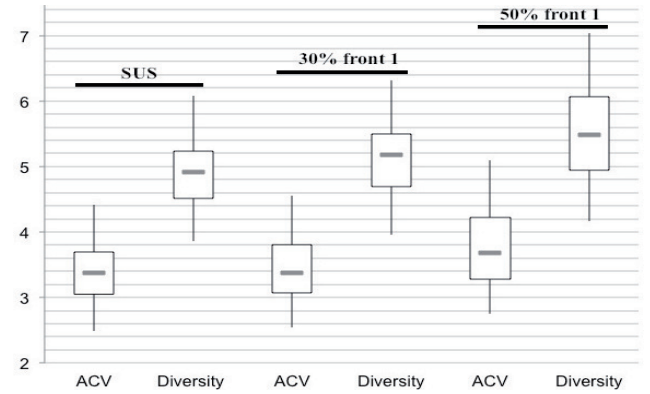

Figure 15. Population size 200

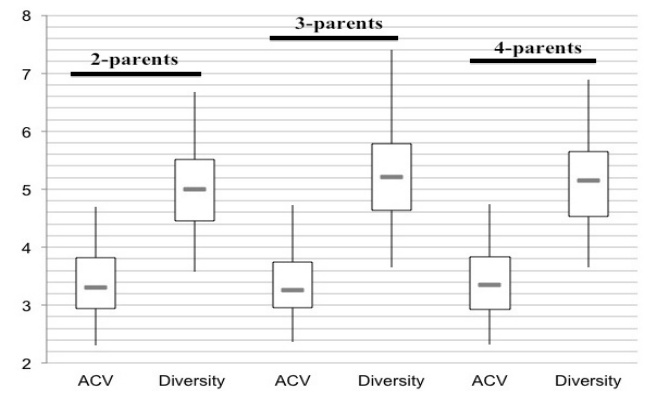

Figure 16. Multi-parent recombinations, population size 100, $p_0 = 30\%$

(Fig. 9) independent of the choice of the selection parameter. Moreover, the diversity also decreases and results in the lowest diversity among all configurations. An increase of the population size to 130 results in a decrease of ACV and in an increase of the diversity, once more independent of the selection parameter (Fig. 13). A further increase of the population size up to 150 and 200 results in a stagnation of the ACV and diversity values (Fig. 14, Fig. 15).

Further, the effect of the selection parameter is evaluated: Varying the population size from 50 to 100 (Fig. 10- Fig. 12), the ACV values are comparable for $p_0 = 0\%$ (denoted as 'SUS' in the figures) and $p_0 = 30\%$ (denoted as '30% front 1' in the figures), though the diversity improves evidently for $p_0 = 30\%$ compared to SUS. Independent of the population size, $p_0 = 50\%$ (denoted as '50% front 1' in the figures) results in a remarkable increase of the ACV values and only a slight improvement of diversity compared to SUS and $p_0 = 30\%$. For the population sizes from 130 to 200, the influence of the selection parameter is reduced (Fig. 13- Fig. 15): There is only a slight improvement to report in diversity for $p_0 = 30\%$ compared to SUS. The convergence is remarkable reduced for $p_0 = 50\%$, though the diversity is improved.

The best performance of the configurations is received with a population size from 70 to 100 and a selection parameter of 30% as the values for ACV are at most low, whereas the diversity values are at most high. At least, the performance of the configurations with a population size from 50 to 100 with $p_0 = 30\%$ are comparable in convergence and diversity with the performance of the configuration population size of 130 and SUS. Concluding, the best configuration is expectable with a population size in the range from 70 to 100 and a selection parameter of $p_0 = 30\%$.

The variation of the parent number within the recombination procedure reveals no effect on the ACV-values and therefore on the convergence (Fig. 16). A very slight increase of the diversity values is achieved by an increase of two parents to three parents. A further increase of the parent number results in a slight decrease of the diversity values. The variation effect is tested for the previously detected optimal algorithm settings of population size and selection parameters.

Regarding the questions presented in the introduction we conclude that an increase of the population size does not result in better performance. The customized NSGA-II provides good performance regarding convergence and diversity within a limited range of population size for the presented three-dimensional minimization problem. Empirically, there is no interdependence between population size and selection: The choice of $p_0 = 30\%$ usually results in the best performance independent of the population size. Therefore, it is not possible to speed up the convergence by increasing or decreasing of the population size and a suitable adaption of the selection parameter.

C. Experiments on the 4D-MOP

The 4D-MOP comprises the molecular objective functions NMW, MW, hydro and InstInd. The results are once more

Population size 30

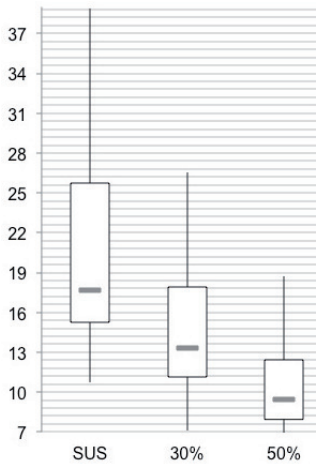

Figure 17. ACV

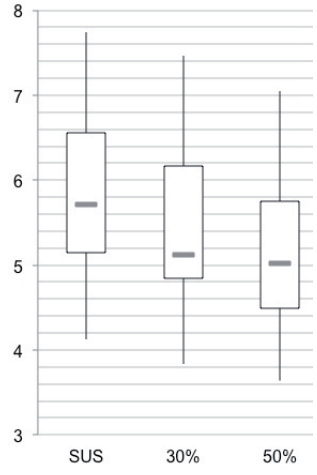

Figure 18. Diversity

Population size 50

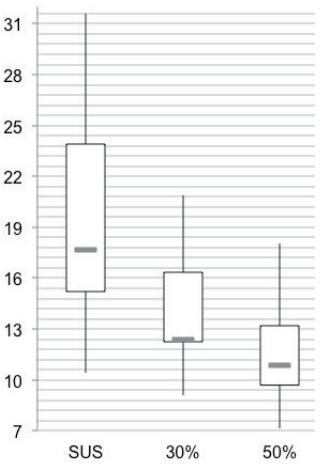

Figure 19. ACV

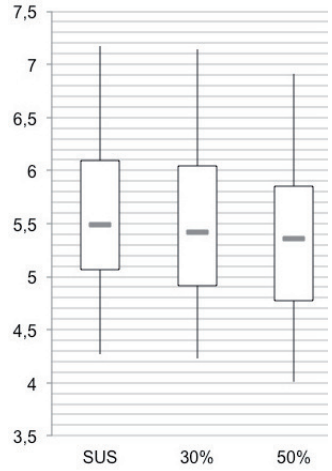

Figure 20. Diversity

Population size 70

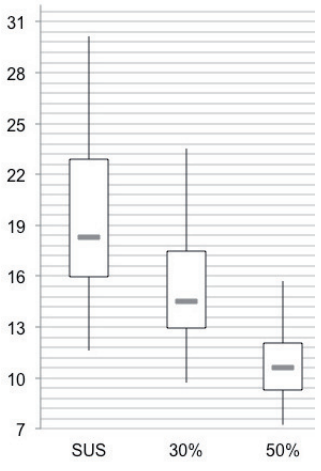

Figure 21. ACV

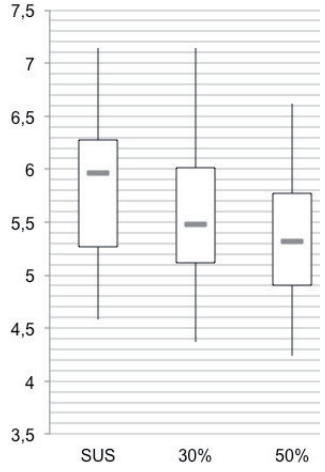

Figure 22. Diversity

Population size 100

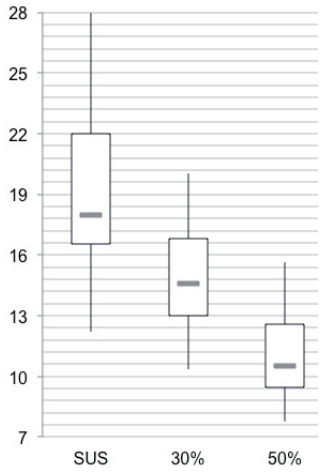

Figure 23. ACV

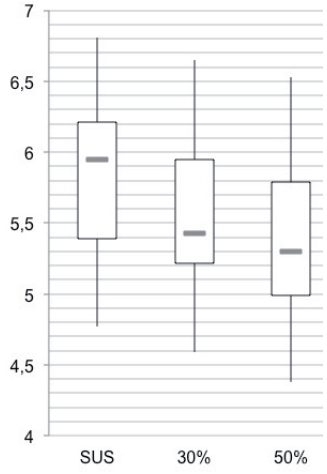

Figure 24. Diversity

Multi-parent recombination

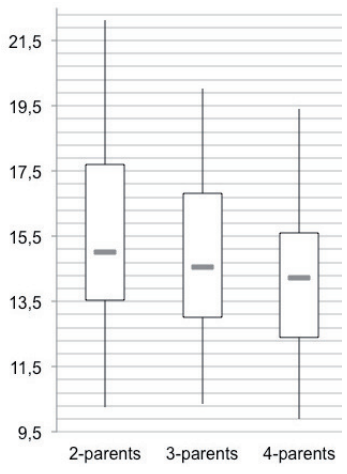

Figure 29. ACV

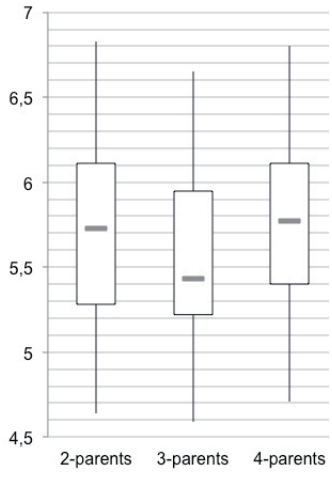

Figure 30. Diversity

Population size 130

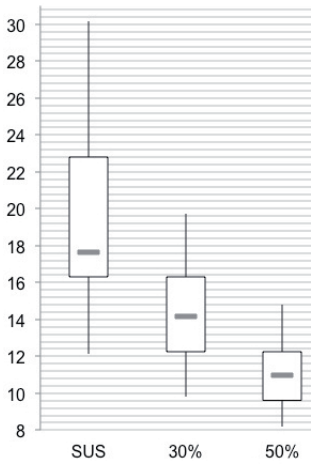

Figure 25. ACV

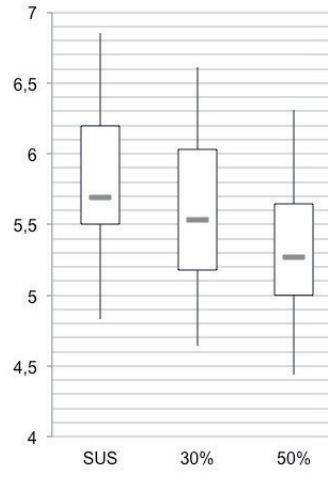

Figure 26. Diversity

Population size 150 and 200

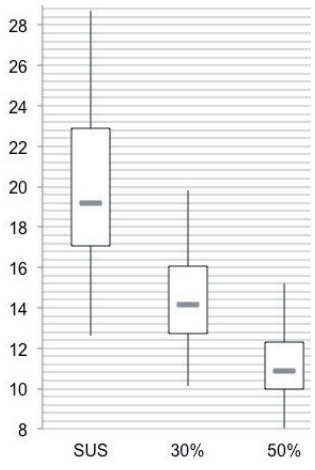

Figure 27. ACV

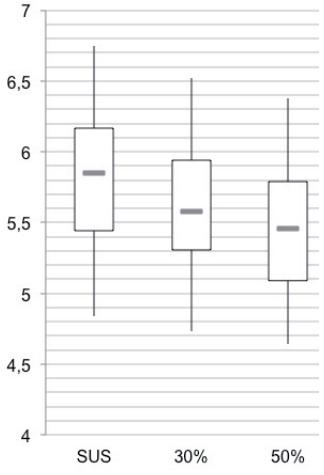

Figure 28. Diversity

discussed regarding an increase or decrease of the standard population size of 100 (Fig. 23, Fig. 24)). In general, a decrease of the population size from 100 to 70 (Fig. 21, Fig. 22), 50 (Fig. 19, Fig. 20) and 30 (Fig. 17, Fig. 18) results in an increase of the range for the ACV as well as the diversity values, which indicates an increasing spread of the indicator values. The tendency of the indicator spread lies in direction of higher values in the case of the convergence (Fig. 17, Fig. 19, Fig. 21). Otherwise, the tendency of the indicator spread for the diversity is more in direction of lower values (Fig. 18, Fig. 20, Fig. 22). This indicates a global convergence reduction with a decrease of the diversity at the same time. This effect is stronger for the selection parameter $p_0 = 0\%$ and is reduced with the increase of this parameter to 30% and 50%.

The increase of the selection parameter results clear decrease of the indicator values in general for all population sizes. This observation is quite different to the results of the 3D-MOP experiments. This is explained by the previously performed landscape analysis: The landscape of the 4D-MOP provides about 50% more optimal solutions and is therefore of a higher optima density than the 3D-MOP. Consequently, the distance between the optimal solutions is reduced. Hence, the higher the selection probability for selecting solutions from the first front the lower are the ACV values and the lower are the diversity values.

An increase of the population size from 100 up to 200 (Fig. 25, 26, 27, 28) results in a stagnation of the ACV values independent of the selection parameter. The same holds for the diversity values and the selection parameter $p_0 = 0\%$. An increase of the selection parameter results in a slight increase of the means as well as an increase of the indicator spread for a population size of 150 and 200 (Fig. 28). The results of population 150 and 200 are depicted in one figure, as there is no visible difference within the indicator values.

The optimal configuration regarding the convergence or ACV values is achieved with a population size of 100. The

optimal configuration for the diversity of the solutions is achieved with a population size of 70 followed directly by the configuration with a population size of 100. Concluding, the best configurations in general are expectable with a population size in range of 70 to 100 and a selection parameter of $p_0 = 30\%$.

The results of the multi-parent recombination is quite different in the case of the 4D-MOP: An increase of the parent number results in a continuous decrease of the ACV values and therefore in better performance regarding the convergence. In the case of the diversity values, the increase of the parent number from 2 to 3 reveals a clear decrease of the diversity values. A further increase of the parent number to 4 achieves slight better diversity values as the configuration with 2-parent recombination. Therefore, the best NSGA-II performance is expected with 4-parent recombination. This confirms the observations of Eiben as presented in the introduction that the optimal number of parents is problem depending.

Compared to the results of the 3D-MOP and the questions raised in the introduction, an increase of the population size does not results in an increase of the performance like in the case of 3D-MOP. The customized NSGA-II provides a good performance with regard to convergence and diversity within a limited range of population size for 3D-MOP as well as 4D-MOP. The selection parameter $p_0 = 30\%$ is a suitable choice for a good balance between at most low ACV values and at most high diversity values. Therefore, the optimal algorithm settings are equal for 3D-MOP as well as 4D-MOP.

D. Discussion of the results

The interdependence of the population size and the selection parameter in this customized NSGA-II as well as the influence of multi-parent recombination is exemplary examined on a generic three- and four-dimensional biochemical minimization problem and the results presented above are discussed according to the five questions raised in the introduction:

The first question is aimed at the influence of large populations on the convergence speed. Early convergence as a main goal of our moGA is defeated since an increase of the population size results in higher speed of convergence. The experiments show that the optimal population size regarding convergence and diversity is in a limited range from 70 to 100 for the three- as well as the four-dimensional optimization problem. An increase of the population size above 100 results in a stagnation of the convergence behavior and the diversity for the three-dimensional optimization problem. Furthermore, a population size lower than 50 does not provide a convincing diversity within the solutions. In the case of the four-dimensional optimization problem, an increase of the population size above 100 also results in a stagnation of the convergence behavior and no significant improvement of the diversity. A decrease of the population size below 70 results in worse convergence and diversity performance.

Our second question is focused on the impact of the population size and the selection parameter. A configuration rule for the selection parameter depending on the population size is necessary in the case of a large interdependence of both. However, the experiments of the three- dimensional optimization problem do not reveal an interdependence of the population size and the selection parameter. Though, the diversity of the configurations with a population size from 50 to 100 is remarkably improved with a selection parameter of 30% compared to $p_0 = 0\%$ (SUS). Further, higher values for p_0 are not advisable as the speed of convergence is reduced. In the case of the four-dimensional optimization problem, in increase of the selection probability above 30% is not advisable as this results in a significant decrease of the diversity independent of the population size.

The third question asks for a range of the population size providing the best performance: This range is fixed to a population size from 70 to 100 based on the evaluation of the experiments. More precisely, the optimal performance for the three- and the four-dimensional optimization problem is achieved within the same range of population size and the same parameter settings. This allows the hypothesis that the optimal algorithm settings are independent of at least three and four dimensions and the customized NSGA-II is an effective and robust tool for biochemical optimization.

The fourth question refers to the influence of the variation of the number of parents within the recombination on the algorithm performance. Three different numbers of parents for recombination are tested. The experiments on the three-dimensional optimization problem reveal no effect on the convergence behavior and the diversity for 2-, 3- and 4- parent recombination, whereas an increase of the parent number results in an improvement of the convergence behavior for the four-dimensional optimization problem. The highest diversity tendency is achieved for four parents. Therefore, an increase of the standard setting of 3 parents to 4 is advisable. A reason for these observations is challenging and the optimal number has to be empirically verified for each optimization problem.

The fifth question refers to the generalization of the results and algorithm settings on higher dimensional optimization problems. As mentioned above, the best performance is achieved for the same algorithm settings in the case of the three-dimensional and four-dimensional optimization problem. This confirms the hypothesis that the performance results are transferable on high-dimensional optimization problems. Only the number of parents within the recombination procedure is challenging for each optimization problem.

VII. Conclusion and Future Work

The presented customized NSGA-II provides a reliable good performance according to the convergence and diversity for a three- and four-dimensional biochemical minimization problem. This good performance is achieved with the same optimal settings, though the three-dimensional problem investigated here is more challenging than the investigated four-dimensional one due to the higher front diversity. This allows the hypothesis that this customized NSGA-II is an efficient and robust genetic algorithm, which potentially provides a high

performance for a wider range of similar molecular problem classes with the property of early convergence.

For future work, we currently work on a selection strategy based on ACV indicator for ongoing improvements. Furthermore, an analysis concept is challenging to gain a deeper insight into molecular MOP or more general real-valued MOP and is in the focus of our research.

REFERENCES

[1] S. Rosenthal and M. Borschbach, "Impact of Population Size and Selection within a Customized NSGA-II for Biochemical Optimization Assessed on the Basis of the Average Cuboid Volume Indicator," BIOTECHNO 2014: The Sixth International Conference on Bioinformatics, Biocomputational Systems and Biotechnologies, IARIA 2014, pp. 1-7.

[2] S. Rosenthal, N. El-Sourani, and M. Borschbach, "Introduction of a Mutation Specific Fast Non-dominated Sorting GA Evolved for Biochemical Optimization," SEAL 2012, LNCS 7673, 2012, pp. 158-167.

[3] D. J. Craik, D. P. Fairlie, S. Liras, and D. Price, "The Future of Peptide-based Drugs," Chemical Biology & Drug Design, 81(1), 2013, pp. 136-147.

[4] N. Röckendorf, M. Borschbach, and A. Frey, "Molecular Evolution of Peptide Ligands with Custom-tailored Characteristics," PLOS Comput Biol 8(12), 2012

[5] S. Rosenthal, N. El-Sourani, and M. Borschbach, "Impact of Different Recombination Methods in a Mutation-Specific MOEA for a Biochemical Application," L. Vanneschi, W. S. Bush, and M. Giacobini (Eds.): EvoBIO 2013, LNCS 7833, 2013, pp. 188-199.

[6] S. Rosenthal and M. Borschbach, "A Benchmark on the Interaction of Basic Variation Operators in Multi-Objective Peptide Design evaluated by a Three Dimensional Diversity Metric and a Minimized Hypervolume," M. Emmerich et al. (eds.): EVOLVE - A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation IV, 2013, pp. 139-153.

[7] J. T. Alander, "On Optimal Population Size of Genetic Algorithms," in Proceedings of the IEEE Computer Systems and Software Engineering, 1992, pp. 65-69.

[8] V. K. Koumousis and C. P. Katsaras, "A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance," IEEE Transactions on Evolutionary Computation, vol. 10, no. 1, 2006, pp. 19-28.

[9] T.-L. Yu, K. Sastry, D. E. Goldberg, and M. Pelikan, "Population sizing for entropy-based model building in genetic algorithms," Illinois Genetic Algorithms Laboratory, University of Illinois, Tech. Rep., 2006.

[10] T. Bäck, A. Eiben, and V. der. Vaart, "An empirical study on GAs without parameters," in Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, 2000, pp. 315-324.

[11] Z. M. Jaroslaw Arabas and J. Mulawka, "GAVaPSa genetic algorithm with varying population size," in Proceedings of the IEEE International Conference on Evolutionary Computation, 1995, pp. 73-78.

[12] A. E. Eiben, M. C. Schut, and A. R. Wilde, "Is Self-Adaption of Selection Pressure and Population Size Possible? a Case Study," in Parallel Problem Solving from Nature - PPSN IX, vol. 4193, 2006, pp. 900-909.

[13] A.E. Eiben, P.-E. Raue, and Zs. Ruttkay, "Genetic Algorithms with Multi-parent Recombination," Proceedings of the Parallel Problem Solving from Nature III, 1994, pp.78-87.

[14] I. Ono and S. Kobayashi, "A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover," Proceedings of the Seventh International Conference on Genetic Algorithm, 1997, pp. 246-253.

[15] S. Tsutsui and A. Ghosh, "A Study on the Effect of Multi-parent Recombination in Real Coded Genetic Algorithms," Proceedings of the 1998 IEEE ICEC, 1998, pp. 828-833.

[16] G. Grosan, M. Oltean, and D. Dumitrescu, "Performance Metrics for Multiobjective Evolutionary Algorithms," Proceedings of Conference on Applied and Industrial Mathematics (CAIM), 2003

[17] BioJava: CookBook, URL: http://www.biojava.org/wiki/BioJava/ [retrieved: December, 2013].

[18] S. Henikoff and J.G. Henikoff, "Amino Acid Substitution Matrices from Protein Blocks," Proc. Natl. Acad. Sci. USA, vol. 89(22), 1992, pp. 10915-10919.

[19] T. Sovany, K. Papos, P.Jr. Kasa, I. Ilic, S. Srcic, and K. Pintye-Hodi, "Application of Physiochemical Properties and Process Parameters in the Development of a Neural Network Model for Prediction of Tablet Characteristics," AAPS PharmSciTech, vol. 14(2), 2013, pp. 511-516.

[20] D. Heider, J. Appelmann, T. Bayro, W. Dreckmann, A. Held, and J.. Winkler, "A Computational Approach for the Identification of Small GTPases based on Preprocessed Amino Acid Sequences," in Technol. Cancer Res. Treat 8, 2009, pp. 333-341.

[21] T. P. Hopp and K. R. Woods, "A computer program for predicting protein antigenic determinants," Mol Immunol, 20(4), 1983, pp. 483-489.

[22] S. Needleman and C. Wunsch, "A General Method Application to the Research for Similarities in the Amino Acid Sequence of Two Proteins," Journal of Molecular Biology, vol. 48(3), 1970, pp. 443-453.

[23] Java API for Genetic Algorithm (JAGA), URL: www.jaga.org/ [retrieved: January, 2014].

[24] Metaheuristic Algorithms in Java (jMetal), URL: www.jmetal.sourceforge.net/ [retrieved: January, 2014].

[25] Java-based Evolutionary Computation Research System (ECJ), URL: www.cs.gmu.edu/∼edab/projects/ecj/ [retrieved: January, 2014]

[26] MOEA framework, URL: www.moeaframework.org/ [retrieved: August, 2014].

[27] Evolutionary Algorithms workbench (EvA2), URL: www.ra.cs.uni-tuebingen.de/software/EvA2/introduction.html/ [retrieved: January, 2014]

[28] Modular framework for Meta-Heuristic Optimization (OPT4J), URL: http://opt4j.sourceforge.net].

[29] P. Merz and B. Freisleben, "Fitness Landscape Analysis and MEmetic Algorithms for the Quadratic Assignment Problem," IEEE Transactions on Evolutionary Computation, vol. 4 (4), 2000, pp.337-352.

[30] P. Stadler, "Fitness Landscape," In Lecture Notes in Physics, vol. 585, 2002.

[31] G. Merkuryeva and V. Bolshakovs, "Benchmark Fitness Lands-cape Analysis," International Journal of Simulation Systems, Science and Technology 12, 2, 201, pp. 3845.

[32] E. Weinberg, "Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference," Biological Cybernetics 63, 1990, pp. 325336.

[33] T. Jones and S. Forrest, "Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithm," In Proceedings of the 6th International Conference on Genetic Algorithms,1995, pp. 184192.

[34] V.K. Vassilev, T.C. Fogarty, and J.F. Miller, "Information Characteristics and the Structure of Landscapes," Evolutionary Computation, vol. 8(1), 2000, pp.31-60.

[35] A. Leier and W. Banzhaf, "Exploring the Search Space of Quantum Programs," In Proceedings of the 2003 Congress on Evolutionary Computation IEEE Press, vol. 1, 2003, pp.170-177.

[36] B.V. Lee, "Analysing Molecular Landscapes using Random Walk and Information Theory," Masterthesis, LIACS, University of Leiden, 2009.

[37] E. Zitzler and L. Thiele, "Multiobjective Optimization using Evolutionary Algorithms - a Comparative Case Study," in A. E. Eiben, T. Bäck, M. Schoenauer, and H. P. Schwefel (EDS.), Fifth International Conference on Parallel Problem Solving form Nature (PPSN-V), 1998, pp. 292-301.

[38] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective Evolutionary Algorithm Test," in Proceedings of the 1999 ACM Symposium on Applied Computing, San Antonio, Texas, 1999, pp. 351-357.

[39] K. Deb and S. Jain, "Running performance metrics for Evolutionary Multiobjective Optimization," Kan GAL Report No. 2002004, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, 2002.

[40] O. Schütze, X. Esquivel, A. Lara, and C. A. Coello Coello, "Using the Averaged Hausdorff Distance as a performance measure in evolutionary multiobjective optimization," IEEE Transactions on Evolutionary Computation, vol. 16(4), 2012, pp. 504-522.

[41] C. A. Coello Coello and N. Cruz Cortis, "Solving Multiobjective Optimization Problems using an Aritifical Immune System. Genetic," Programming Evolvable Mach., vol. 6 (2), 2005, pp. 163-190.