University of Cyprus

DEPARTMENT OF COMPUTER SCIENCE

SCIENTIFIC WORKFLOW SYSTEMS AND MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS FOR LIFE SCIENCES INFORMATICS

DOCTOR OF PHILOSOPHY DISSERTATION

Christos C. Kannas

2017

University of Cyprus

DEPARTMENT OF COMPUTER SCIENCE

SCIENTIFIC WORKFLOW SYSTEMS AND MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS FOR LIFE SCIENCES INFORMATICS

Christos C. Kannas

A Dissertation Submitted to the University of Cyprus in Partial Fulfilment

of the Requirements for the Degree of Doctor of Philosophy

June, 2017

VALIDATION PAGE

Doctoral Candidate: Christos C. Kannas

Doctoral Thesis Title: SCIENTIFIC WORKFLOW SYSTEMS AND MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS FOR LIFE SCIENCES INFORMATICS

The present Doctoral Dissertation was submitted in partial fulfilment of the requirements for the Degree of Doctor of Philosophy at the ***Department of Computer Science*** *and was approved on the .... by the members of the* ***Examination Committee****.*

Examination Committee:

Research Supervisor: ______________________

Prof. Constantinos S. Pattichis

Committee Member: ______________________

Assoc. Prof. Christos Christodoulou

Committee Member: ______________________

Asst. Prof. Vasilis Promponas

Committee Member: ______________________

Dr. George M. Spyrou

Committee Member: ______________________

Prof. Val Gillet

DECLARATION OF DOCTORAL CANDIDATE

The present Doctoral Dissertation was submitted in partial fulfilment of the requirements for the Degree of Doctor of Philosophy of the University of Cyprus. It is a product of original work of my own, unless otherwise mentioned through references, notes, or any other statements.

______________________________ [Christos C. Kannas]

______________________________ [Signature]

ABSTRACT

Ο τομέας των Συστημάτων Διαχείρισης Επιστημονικών Ροών Εργασίας έχει λάβει μεγάλο ενδιαφέρον τα τελευταία χρόνια. Υπάρχουν τομείς όπως το ευρύτερο ερευνητικό πεδίο των βιοεπιστήμων, η επεξεργασία βίντεο και η πληροφορία δεδομένων, όπου η χρήση της ισχύος ενός Συστηματος Διαχείρισης Επιστημονικών Ροών Εργασίας έχει γίνει κανόνας. Ορισμένοι αναφέρονται στη διαδικασία σχεδιασμού μιας Επιστημονικής Ροής Εργασίας ως οπτικό προγραμματισμό. Έχουμε αναπτύξει και αξιολογήσει ένα Σύστημα Διαχείρισης Επιστημονικών Ροών Εργασίας, την πλατφόρμα **Li**fe **S**ciences **I**nformatics (LiSIs), η οποία είναι: (1) μια καινοτόμα ολοκληρωμένη διαδικτυακή πλατφόρμα Virtual Screening (VS), και (2) έχει χρησιμοποιηθεί με επιτυχία για την ταυτοποίηση καινοτόμων χημειοπροληπτικών παραγόντων του καρκίνου από μια εμπορική βάση δεδομένων με διαθέσιμα μόρια.

Η Αυτό-Προσαρμογή είναι ένας αποτελεσματικός τρόπος για τον έλεγχο των παραμέτρων αναζήτησης ενός Εξελικτικού Αλγόριθμου αυτόματα κατά τη διάρκεια τις βελτιστοποίισης. Βασίζεται στην εξελικτική αναζήτηση του χώρου των παραμέτρων αναζήτησης και έχει αποδειχθεί επίσης ως μέθοδος ελέγχου των παραμέτρων αναζήτησης σε πραγματικό χρόνο για μια ποικιλία παραμέτρων αναζήτησης. Ο προτεινόμενος Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) είναι ένας αλγόριθμος δύο επιπέδων. Το εξωτερικό επίπεδο είναι ο αλγόριθμος που είναι υπεύθυνος για τις αυτό-προσαρμοζόμενες τεχνικές και βασίζεται στο αλγοριθμηκό πλαισίο Multi-Objective Genetic Algorithm (MOGA). Το εσωτερικό επίπεδο είναι ο elite Multi-Objective Evolutionary Graph Algorithm (eMEGA). Τόσο ο εξωτερικός όσο και ο εσωτερικός αλγόριθμος είναι βασισμένοι στον προϊγούμενα προτεινόμενο αλγοριθμικό πλαισίο Multi-Objective Evolutionary Graph Algorithm (MEGA). Ο εξωτερικός αλγόριθμος λειτουργεί σε χρωμόσωματα στοιχείων, ενώ ο εσωτερικός αλγόριθμος λειτουργεί σε χρωμόσωματα μοριακού γραφήματος. Ο προτεινόμενος Self-Adaptive MOEA είναι: (1) μια μοναδική προσέγγιση πλαισίου βελτιστοποίησης πολλαπλών κριτηρίων, (2) χρησιμοποιεί ένα προσαρμοσμένο χρωμόσωμα για να

κωδικοποιήσει τις παραμέτρους αναζήτησης του eMEGA, και (3) έχει χρησιμοποιηθεί με επιτυχία για των σχεδιασμό νέων μορίων σε ευρύ φάσμα στόχων.

ABSTRACT

The field of Scientific Workflow Management Systems (SWMSs) has been receiving considerable interest in recent years. There are fields such as life sciences, video processing and data information, where utilising the power of a SWMS has become a norm. Some refer to the process of designing a Scientific Workflow (SW) as visual programming. We have developed and evaluated a SWMS specialised for Virtual Screening (VS), the **Li**fe **S**ciences **I**nformatic**s** (LiSIs) platform, which is: (1) a novel integrated web based VS framework, and (2) has been successfully used to identify novel cancer chemopreventive agents from a commercial database of available molecules.

Self-adaptation is an efficient way to control the search parameters of an Evolutionary Algorithm (EA) automatically during optimization. It is based on implicit evolutionary search in the space of search parameters, and has been proven to work well as on-line parameter control method for a variety of search parameters, from local to global ones. Our proposed Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) is a two level algorithm. The outer level is the algorithms that is responsible for the self adaptive techniques and is based on a Multi-Objective Genetic Algorithm (MOGA) implementation. The inner level is the actual elite Multi-Objective Evolutionary Graph Algorithm (eMEGA). Both the outer and inner algorithm are variations of our previously proposed Multi-Objective Evolutionary Graph Algorithm (MEGA) framework. The outer MOGA operates on chromosomes of elements, while the inner eMEGA operates on molecular graph chromosomes. The proposed Self-Adaptive MOEA is: (1) a unique approach Multi-Objective Optimization (MOO) framework, (2) uses a custom chromosome to encode the search parameters of eMEGA, and (3) has been successfully used to design novel molecules in a wide spectrum of targets.

ACKNOWLEDGEMENTS

This dissertation has been a task I have been working on for over 5 years. My initial plans, back in 2005, for entering a Ph.D program in Computer Science switched to an MSc. in Computer Science due to my desire to explore the world of applied innovation and commercial software development. That desire, although still firmly in place, never extinguished my drive to complete my studies and invest time and effort to thoroughly investigate the potential benefits of knowledge acquisition, use and transfer to everyday problems. To me, it was only a matter of time to formally enter the appropriate Ph.D. program when circumstances allowed, something which happened in 2011.

Still, the completion of what started out as a side-task required the support of numerous family members, friends and colleagues to whom I am greatly indebted and grateful. Above all I would like to thank my family starting from my son Constantinos for putting up with me during this time, especially the final year when I was more often absent than not; my parents who have instilled the urge for learning and improving myself and for their support; my parents in law for their understanding and support, and, last, but not least, my wife Christiana, without whose continued support, encouragement, understanding and love this work would have most probably remained a dream.

DEDICATION

My efforts would never have been successful had it been not for the mentor-ship and support of Prof. Costas Pattichis with whom I have worked for the past 8 years. It truly would have been a much more difficult task without his advice and guidance and, at times, patience.

This work and career path would have not been successful without the mentor-ship, support and friendship of Dr. Christodoulos Nicolaou, whom I had the privilege of working with. Working with Christodoulos at Noesis Cheminformatics Ltd. was the start of a career path I choose to follow. He is my mentor in anything related to chem[o]informatics and multi-objective evolutionary algorithms.

Thanks are also due to my collaborators in the GRANATUM project, Prof. Andreas Constantinou of the University of Cyprus, Dr. Christiana Neophytou, Dr. Christiana Savva for their expert guidance in Cancer Chemoprevention research area, Prof. Vasilis Prombonas of the University of Cyprus, Dr. Giannis Kirmitzoglou, Ioanna Kalvari and Zenonas Antoniou for being part of the development team of **Li**fe **S**ciences **I**nformatics (LiSIs) platform. This work had been partially supported through the EU-FP7 GRANATUM project, *"A Social Collaborative Working Space Semantically Interlinking Biomedical Researchers, Knowledge and data for the design and execution of In Silico Models and Experiments in Cancer Chemoprevention"*, contract number 270139. Special thanks to my collaborators for using LiSIs in at least one use case of EU-FP7 Linked2Safety project, *"A Next-Generation, Secure Linked Data Medical Information Space For Semantically-Interconnecting Electronic Health Records and Clinical Trials Systems Advancing Patients Safety In Clinical Research"*, contract number 288328.

I would also like to thank Dr. Erika Loizidou for providing and helping me with the additional experiments on Proteasome B5 for further testing of Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2LB-MOPSO Two-Local-Best Multi-Objective Particle Swarm Optimization

ACO Ant Colony Optimization

ADME Absorption, Distribution, Metabolism, Excretion

AI Artificial Intelligence

BW Business Workflow

CE Cross Entropy

CEC Congress on Evolutionary Computation

cLogP calculated Octanol-Water partition coefficient

CMA-ES Covariance Matrix Adaptation Evolution Strategy

CoMODE Co-evolutionary Multi-Objective Differential Evolution

CPR Chemoprevention Research

DDP Drug Discovery Process

DE Differential Evolution

DND De Novo Design

DNMT DNA Methyltransferase

DT Decision Trees

EA Evolutionary Algorithm

EC Evolutionary Computation

EDA Estimation of Distribution Algorithm

eMEGA elite Multi-Objective Evolutionary Graph Algorithm

EMOO Evolutionary Multi-Objective Optimization

ES Evolutionary Strategies

EP Evolutionary Programming

ER-α Estrogen Receptor-α

ER-β Estrogen Receptor-β

GA Genetic Algorithm

GRASP Greedy Randomized Adaptive Search Procedure

HBA Hydrogen Bond Acceptors

HBD Hydrogen Bond Donors

HCS Hill Climber with Sidestep

IBEA Indicator-Based Evolutionary Algorithm

ID identity

IGD Inverted Generational Distance

I/O Input/Output

kNN k-Nearest Neighbours

LiSIs **Li**fe **S**ciences **I**nformatics

MADE Memetic Algorithm based on Differential Evolution

MCS Maximum Common Substructure

MEGA Multi-Objective Evolutionary Graph Algorithm

ML Machine Learning

mMOEA Memetic Multi-Objective Evolutionary Algorithm

MNFSSP Multi-Objective No-wait Flow-Shop Scheduling Problem

MO Multi-Objective

MOARF Multi-Objective Algorithm for Replacement of Fragments

MOCLPSO Multi-Objective Comprehensive Learning Particle Swarm Optimizer

MOEA Multi-Objective Evolutionary Algorithm

MaOEA Many-Objective Evolutionary Algorithm

MOEAD Multi-Objective Evolutionary Algorithm based on Decomposition

MOFit Multi-Objective Fitness

MOGA Multi-Objective Genetic Algorithm

MOGLS Multi-Objective Genetic Local Search

MOIS Multi-Objective Immune System

MOO Multi-Objective Optimization

MOOP Multi-Objective Optimization Problem

MaOOP Many-Objective Optimization Problem

MOTS Multi-Objective Tabu Search

MPGA Multi-Population Genetic Algorithm

MW Molecular Weight

NRGA Non-Dominated Ranked Genetic Algorithm

NPGA Niched Pareto Genetic Algorithm

NSGA Non-dominated Sorting Genetic Algorithm

OGD Optimal Graph Design

PA Pareto Archive

PAp Pareto Approximation

PEA Parallel Evolutionary Algorithm

PF Pareto Front

PMEGA Parallel Multi-Objective Evolutionary Graph Algorithm

PS Pareto Set

PSO Particle Swarm Optimization

QGA Quantum-inspired Genetic Algorithm

QSAR Quantitative Structure Activity Relationship

RASCAL RApid Similarity CALculation

RECAP Retro-synthetic Combinatorial Analysis Procedure

RM-MEDA Regularity Model-based Multi-Objective Estimation of Distribution Algorithm

RF Random Forests

Ro5 Rule of Five

RSD Ring System Decomposition

RCPSP Resource-Constrained Project Scheduling Problem

RIP Resource Investment Problem

RLP Resource Levelling Problem

SA Simulated Annealing

Self-Adaptive MOEA Self-Adaptive Multi-Objective Evolutionary Algorithm

SDF Structure Data File

SMILES Simplified Molecular Input Line Entry Specification

STIR STagnation Identification and Resolution

SOOP Single Objective Optimization Problem

SPEA2 Strength Pareto Evolutionary Algorithm 2

SVM Support Vector Machines

SW Scientific Workflow

SWMS Scientific Workflow Management System

TOPSIS Technique for Order of Preference by Similarity to Ideal Solution

TPSA Topological Surface Polar Area

TPSPGA two-phase Subpopulation Genetic Algorithm

VS Virtual Screening

WM Workflow Model

Chapter 1

Introduction

Over the last three decades computational approaches are being used to assist the research performed at the early steps of Drug Discovery Process (DDP) (Figure 1). The computational tools are used in the steps of: (a) **Virtual Screening (VS)**, where the selection of molecules with desired characteristics takes place, (b) **Lead Optimisation**, where experts use computational tools to make changes on previously selected molecules and validate them, (c) **Quantitative Structure Activity Relationship (QSAR)**, where computational tools are used to explore the activity of the optimised leads, and (d) **Optimised Synthesis**, where computational tools are used to help scientists to identify the best route for synthesising the molecules selected. Additionally computational approaches are being used in automatic and assisted design of new molecules.

The last decade we moved from using individual tools to implement a VS process to using Scientific Workflow Management Systems (SWMSs), which are software suites providing all the relevant tools required to design and run VS processes. SWMSs are also used in QSAR analysis and for Optimised Synthesis with the aid of specialised software.

Using computational approaches for helping scientists design novel molecules dates back to 1990's [1]. The approaches used either come up with an approximate solution, e.g., by stochastic sampling, or restrict the search to a defined section of chemical space which can be screened exhaustively [2].

In this thesis, we document the research, development and use of SWMSs in Life Sciences, and more specifically about designing, developing and using such platforms to be used for designing and running VS workflows. Additionally we explore the application of Multi-Objective Evolutionary Algorithms (MOEAs) and Self-Adaptive Multi-Objective Evolutionary Algorithms (Self-Adaptive MOEAs) as software approaches for molecular De Novo Design (DND).

The following sections of this chapter provide overviews of Drug Discovery and Chem[o]informatics in Section 1.1, SWMSs in Section 1.2, and Self-Adaptive MOEAs in Section 1.3. Then in Section 1.4 we document the objectives of this thesis, and at the end in Section 1.5 we describe the original contributions of the research performed.

1.1 Drug Discovery and Chem[o]informatics

Modern Drug Discovery Process (DDP) is an interdisciplinary effort spanning the fields of medicinal practice, biology and chemistry. The process is long, laborious and complex. As such it produces an amazing wealth of information related to the very nature of life and its inner functions. Such wealth of information coupled with the complexity of the problem constitutes a fertile and challenging ground for the development of computational methods and applications. Figure 1 illustrates the steps and the time-line requirements of DDP with and without the use of *in silico* tools.

Chem[o]informatics combines the scientific working fields of Chemistry and Computer science [3]. The field has developed in line with increased pharmaceutical data generation and has been an active area of research for a number of years. Currently, chem[o]informatics permeates all aspects of drug discovery and continues to drive research and development of novel computational algorithms and applications [4], [5].

VS is the computational counterpart of biological screening performed in laboratories. Its goal is to decrease the number of compounds physically screened by identifying small subsets of large molecular databases that have an increased probability to be active against a specific biological target [6], [7]. In this respect the method is related to machine learning techniques, such as classification and regression, which prepare predictive models to estimate the behaviour of unknown records based

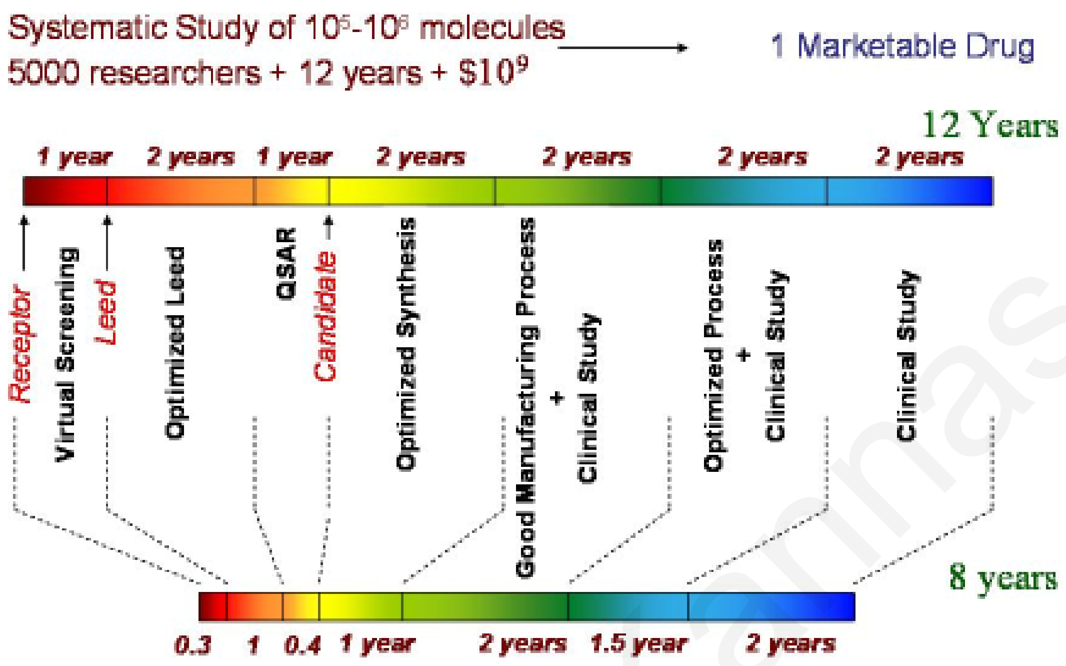


Figure 1: Time-line of Drug Discovery Process, retrieved via Google image search.

on a set of records with known properties. Typically, VS processes involve substantial numbers of molecules and combine a variety of computational techniques [8], [9], [10], [11].

VS can be performed on libraries of real or virtual compounds and requires either measured activities for some known compounds or a structure of the biomolecular target [12]. When only measured activities of compounds are known VS may employ analog-based library design, classification/regression models or any combination of the above. The methods offer some advantages while they suffer from several shortcomings and so researchers typically design a VS experiment taking into account the specific requirements of each case. If high quality activity measurements about the ligands are available regression methods (in the form of e.g. QSAR models) can be used to extract rules capturing the essence of ligand similarity, and hopefully binding action, with high confidence. These rules can easily be used to filter untested compounds swiftly. Classification methods have fewer requirements than QSAR but also produce cruder results. Some methods rely on predefined sets of molecular descriptors and this makes them appropriate as general tools. However, such over-dependence on the descriptor set chosen restricts their potential pool of models and general findings. Reports in the literature [13], [14] describe the usage of descriptor sets in the 100's of thousands, a clear attempt to ensure that no significant ligand feature will be missed. Similar issues trouble the usage of 2D analog-based library design methods based on similarity searches. Approaches relying on the extraction and use of a detailed pharmacophore representation are plagued by a different set of problems. A major one is ensuring that the ligands under investigation bind in the same fashion to the target, i.e. share the same pharmacophore. This task is by no means trivial. In the event that the pharmacophore extraction process is applied on a set of compounds with distinct binding modes a result will be typically produced but it will be misleading. 3D pharmacophore extraction faces the additional concern of the inherent flexibility of the ligand molecules. In this case, common methods either force the selection of a single conformation for each ligand, and run the risk of picking conformations other than the bioactive one, or try to produce a pharmacophore representation general enough to accommodate some of the flexibility of the ligands. The latter approach is more complex and may produce pharmacophore representations that are way too general to be useful for the VS task.

When the structure of the target receptor is known the VS methods of choice typically rely heavily on docking and small molecule modelling. Initially they take advantage of the knowledge about the receptor site to model it and then perform docking of molecules from a database in a systematic manner. A number of conformations are usually sampled for each molecule [15] and a score for every possible docking attempt is kept [16], [17]. Due to the costly nature of numerous steps of the process Linux clusters are widely employed by the pharmaceutical industry [15], [18]. Additionally, databases of multiple conformers of compounds are prepared to avoid their reproduction for every VS run [19]. As a result, currently, databases with millions of compounds can be screened within a few hours [18].

The key success measurement of VS is the achievement of high enrichment, i.e. getting an experimental hit rate for the subset of compounds it recommends that is considerably increased over random compound sets [18]. A successful process with high enrichment results in considerable savings in resources and time, since fewer compounds need to be physically screened while most hits present in the original large database are retrieved. Often, to improve the results of VS several methods are used and their results are combined to produce a concise, high quality virtual hit list [15], [16]. Furthermore it is common to perform a pre-processing step where databases of molecules are cleaned by filtering out compounds with undesired properties such as large size, high flexibility and non-compliance to Lipinski's rule of 5 [20]. During this step compounds containing known unwanted substructures, e.g. known toxicophores, are also eliminated [18]. However, and despite drastic improvements of various algorithms and steps involved in the process, the accuracy of VS still varies depending on the pharmaceutical target, the virtual library and the docking and scoring methods used. Thus, a necessary last step to the process is evaluation of the virtual screening experiment results typically via visual inspection by a human expert [21]. Figure 2 illustrates the common steps of a VS experiment.

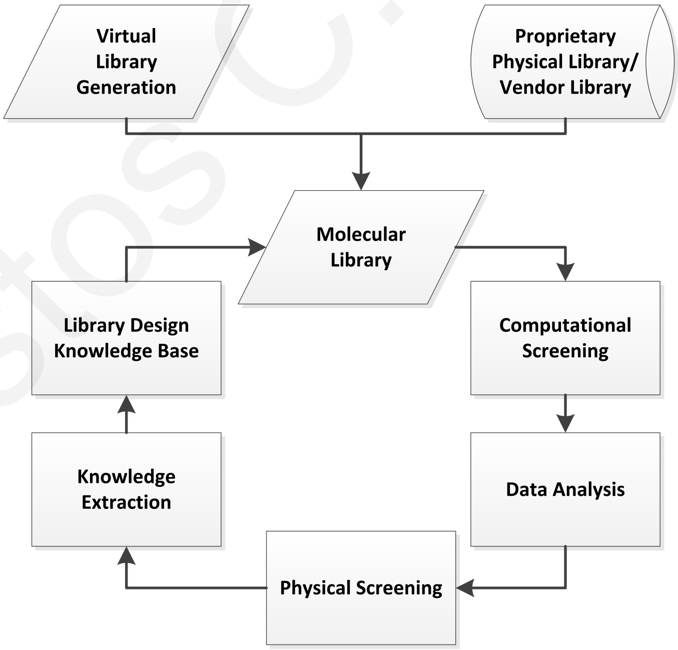

Virtual Library Generation
Proprietary Physical Library/ Vendor Library
Molecular Library
Library Design Knowledge Base
Computational Screening
Knowledge Extraction
Data Analysis
Physical Screening

Figure 2: A typical set-up for a Virtual Screening (VS) experiment.

1.2 Scientific Workflow Management Systems

Scientific Workflows (SWs) enable scientists to plug together problem solving computational components and implement complex *in silico* experiments [22], such as the analysis of datasets of multi-Terabyte magnitude that arise from sensors or computer simulations, the design and execution of complicated algorithms requiring multiple computationally intensive steps. SWMSs accelerate scientific discovery by incorporating data management, analysis, simulation, and visualization tools into a common platform. They provide an interactive visual interface that facilitates the design and execution of workflows.

More over SWMSs enable remote access as well as data and services sharing, making possible collaborations among geographically distributed researchers. Traditionally, many scientists have been using batch files, shell scripts, and programs written in general-purpose scripting languages (e.g., Perl, Python) to automate their tool-integration tasks [23]. Visual representation of the task flow and visual channelling of data are two of the advantages that derive from the properties of a workflow as opposed to lines of code directing the flow. Provenance information, which is very important for the reproducibility of the experiments as well as for the tracking of errors, is also a useful characteristic of workflows commonly not present in scripting tools [24].

Re-usability and transparency is achieved easily by the reuse of a workflow or the use of an existing workflow inside a new workflow. Finally complex implementation details such as parallelism and pipelining can be handled transparently by the SWMSs in order to achieve maximum efficiency for execution time [25]. Essentially, SW technology is a tool that automates the execution of an experiment, which can offer multiple benefits for all the phases of an experiment's life-cycle. The recent popularity of SWMSs is partially owed to the emergence of the computational science paradigm, which promotes collaboration between scientists both within and across disciplines. Through the use of SWs, such interdisciplinary teams can collaborate closely, share workflows and computational components, and jointly undertake research initiatives requiring end-to-end scientific data management and computational analysis [26]. Advances in grid technologies allow workflows to exploit parallel executions enabling large-scale data processing. In this case, workflows are used as a parallel programming model for data-parallel applications. Web services allow ease of access to local and distributed data sources as well as data aggregation from highly heterogeneous environments [25].

As in the case of many other tools, SWMSs quickly found application in a great number of diverse scientific domains, although they were originally developed with a specialized domain application in mind. Figure 3 illustrates some of the main application domains of SWMSs [27]. This domain independence is mainly owed to the abstraction that characterizes the workflow paradigm.

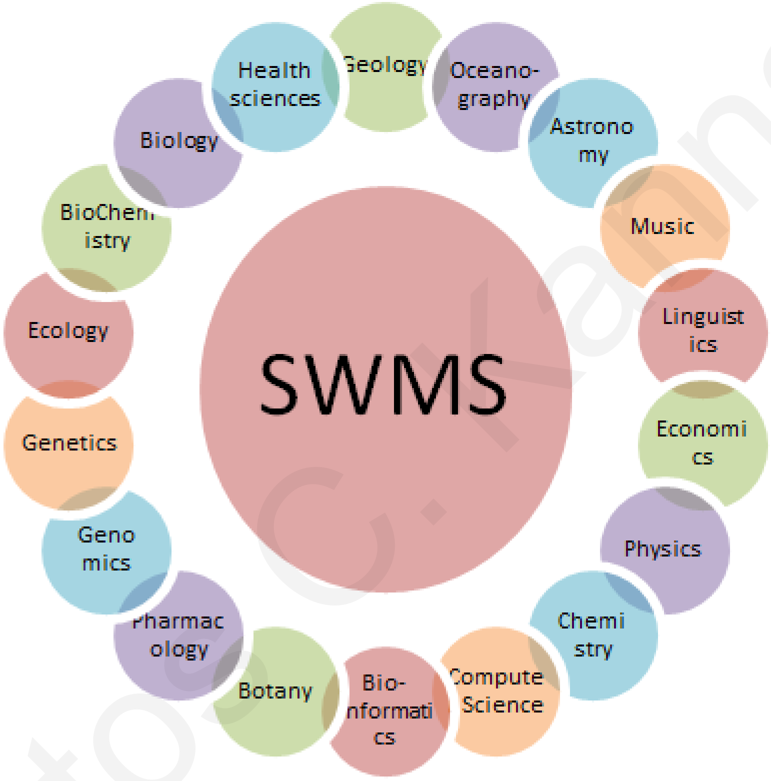


Figure 3: Application domains of Scientific Workflow Management Systems (SWMSs), provided by Achilleos *et al.* [27].

1.3 Self-Adaptive Multi-Objective Evolutionary Algorithms

Self-adaptation is an efficient way to control the strategy parameters of an Evolutionary Algorithm (EA) automatically during optimization. It is based on implicit evolutionary search in the space of strategy parameters, and has been proven to perform well as an online parameter control method for a variety of strategy parameters, from local to global ones [???].

Research on self-adaptation dates since the late 1960s, with Reed *et al.* in [28] proposing parameter adaptation mechanisms for an EA that learns to play poker. A few years later in 1970, Rosenberg in [29] proposed a technique to adapt the probability for applying crossover. At the same time frame Weinberg and Berkus in [30] introduced the meta-evolutionary approaches. Then in 1973, Rechenberg in [31] introduced the 1/5th rule, an adaptation mechanism for step size control of Evolutionary Strategies (ES). Five years later (1978), Mercer and Sampson in [32] reintroduced the meta-evolutionary approaches. And in 1986, Grefenstette in [33] proposed a meta-level adaptive system for tuning the primary optimisation algorithms parameters.

From there on researchers propose new algorithms and approaches based on those initial research articles. Eiben *et al.* [34] proposed a taxonomy regarding parameter setting for EAs. Spears and Jong in 1991 [35] proposed to use a uniform crossover operator as an adaptive operator technique. Saravanan *et al.* in 1995 [36] compared a Gaussian perturbation self-adaptive mechanism with Lognormal perturbation self-adaptive mechanism. Recently, Batista *et al.* in 2010 [37] proposed a Chaotic differential mutation function as self-adaptive mechanism. Jain and Deb in 2013 [38], proposed an improved version of Non-dominated Sorting Genetic Algorithm (NSGA) for Many-Objective Optimization Problem (MaOOP) named ”Improved Adaptive Approach for Elitist Nondominated Sorting Genetic Algorithm for Many-Objective Optimization” (A^2-NSGA-III). Also in 2013, Oliver *et al.* [39], proposed a modified Multi-Objective Genetic Algorithm (MOGA) based on NSGA-II with self-adaptive strategies in mutation (mutation probability per gene) and crossover (crossover probability per chromosome and uniform crossover). In 2015, Shahsavar *et al.* [40] proposed three self-adaptive Genetic Algorithms (GAs) for a triple-objective project scheduling problem.

1.4 Objectives

VS is a computational process that involves numerous steps depending on the task at hand. A VS process can be represented as a directional graph, where the computational steps are the nodes and the data transfer between steps are the edges of the graph. As such VS can be represented by a SW.

SWMSs are used in Life Sciences as a tool to design SWs for VS processes for many years. Most of these platforms are commercial and proprietary that have been built for this specific purpose. Though there are a few that are free to use, these serve a more general purpose and have been developed as desktop tools.

There was the need to provide a SWMS platform specialised for VS that would be accessed via a web interface. We took up this challenge, to design and implement a web accessed SWMS specialised for VS process. As a result we have built **Li**fe **S**ciences **I**nformatic**s** (LiSIs), a web accessed SWMS specialised for VS process based on Galaxy[1] SWMS.

Using Artificial Intelligence (AI) and Machine Learning (ML) algorithms to solve problems requires to model the problem in an appropriate form that the selected algorithm can read and then use a set of input parameters for initialising and running the algorithm. These two things are the most important information we have to provide in order to have meaningful results.

Most of the time we focus too much in one of the two, and the results we get are not what we expect. Figuring out how to model the problem for a selected algorithm requires knowledge of the problem we want to find solutions for, and knowledge about the algorithm's inputs and the process used to solve a problem. On the other hand providing the set input parameters for the algorithm comes to performing a number of different executions of the algorithm starting from a reference point of input parameters and adjusting them according to the results we obtain and their comparison between the different executions.

As we can understand from this, we can provide a better model of the problem only through the good knowledge of the problem domain and the selected algorithm process. But finding the most suitable set of input parameters is an iterative process of using different input arguments and comparing the results with previous executions. Which raises the question, **”What if we had an automated way of finding the most suitable set of input parameters and getting the optimal or near optimal solutions for a given problem and algorithm?”**.

[1] `https://galaxyproject.org/`

In our research we are using MOEAs for molecule DND. The algorithms that we use represent their solutions as graphs, which is a suitable representation of a molecule, and have graph inspired mutation and crossover operators; thus we consider that the modelling of the problem is good enough for our needs. But when we are looking for solutions then we have to run our algorithmic implementations multiple times using different input arguments i.e. population size, iterations, mutation and crossover probability, selection mechanism, and new generation mechanism. This makes it difficult to find the most suitable combination of input arguments given a specific problem. Looking for a way to simplify and automate (to a certain degree) this iterative, time consuming and dull process is a motivating and exiting experience that will be explored to develop a Self-Adaptive MOEA which will be using our previously proposed framework Multi-Objective Evolutionary Graph Algorithm (MEGA) for molecule DND.

1.5 Original Contributions

- **Integrated web based Virtual Screening Framework:** Computational tools for VS exist as standalone tools or in commercial software suites. Also there are approaches that provide integration of such tools as desktop applications in a client server approach. Our approach, **Li**fe **S**ciences **I**nformatics (LiSIs) is a VS integrated platform based on SW modelling for Life Sciences. LiSIs aims to provide a set of tools to create, update, store and share SWs for the discovery of active compounds for biomedical researchers. To the best of our knowledge this was one of the first web based SWMS for Life Sciences. The system is available via a web interface through a password protected, tiered login process. LiSIs is comprised of five (5) major layers of functionalities: (a) Input, (b) Pre-Processing, (c) Processing, (d) Post-Processing, and (e) Output. Each layer hosts a collection of components categories essentially implementing a variety of functionalities. A component category may implement different variations of the same functionality.

- **Application of Integrated web based Virtual Screening Framework:** LiSIs was used for the implementation of a VS experiment in order to identify molecules able to bind to Estrogen Receptor-α and/or Estrogen Receptor-β. A SW was designed and used for this specific experiment. A selection of molecules highly ranked were hand-picked and further investigated in *in-vitro* experiments to provide feedback for the calibration of the tools available on LiSIs platform and also were used to select a small set for further research. LiSIs platform aimed to fill the current void in the application of advanced chem[o]informatics and computational chemistry technology in determining efficacy and predicting possible mechanism(s) of action or identifying a possible receptor(s) for a chemopreventive agent in life sciences.

- **Self-Adaptive Multi-Objective Evolutionary Algorithm:** A Self-Adaptive MOEA based on our previously published Multi-Objective Evolutionary Graph Algorithm (MEGA) framework, is proposed. Self-Adaptive MOEA is a two level algorithm. The first/outer level is the algorithm that is responsible for the self adaptive techniques and is a Multi-Objective Genetic Algorithm (MOGA) implementation. The second/inner level is the actual elite Multi-Objective Evolutionary Graph Algorithm (eMEGA).

- **Original Components of Self-Adaptive Multi-Objective Evolutionary Algorithm:** Two original contributions are documented, *chromosome encoding* and *population evaluation*. These features make our Self-Adaptive MOEA, a problem agnostic algorithm. Self-Adaptive MOEA has been designed in a way that is easily adaptable, expandable and scalable (utilising multi-core parallelism). Our proposed Self-Adaptive MOEA operates on, *chromosomes* of fixed length that are generated from an alphabet where each gene has different type and value ranges. In regards to the chromosome details, the gene in position 0 represents the mutation probability, the gene in position 1 represents the crossover probability, the gene in position 2 represents the selection type of eMEGA, and finally the gene in position 3 represents the diversity type of eMEGA, which can get one of ”phenotype” and ”genotype”.

Population evaluation in Self-Adaptive MOEA is based on the following objective fitness functions: (a) **Non Dominated Solutions Percentage:** where for each individual it calculates the percentage of non dominated solutions over the total number of solutions. (b) **Unique Solutions Percentage:** where for each individual it calculates the percentage of unique solutions over the total number of solutions. (c) **Pareto Front Hypervolume:** where for each individual it calculates the hypervolume [41] of its Pareto Front (PF)s. Hypervolume measures the space covered by each PF from a reference point, this might be the target if it is known or a starting point from the initial population. For example, if the reference point is a starting point then the PF with the larger hypervolume value yields better results.

- **Case Studies for Self-Adaptive Multi-Objective Evolutionary Algorithm:** The Multi-Objective (MO) component and self-adaptive functionality of the Self-Adaptive MOEA are unique to the application of molecular DND. Self-Adaptive MOEA was applied to a number of experiments as a way to validate its performance against eMEGA and across different situations. Self-Adaptive MOEA was used to design molecules that bear similarity to Seliciclib[2] , Tamoxifen[3] , Raloxifen[4] and Ixazomib[5] , across different experiments. We noticed that Self-Adaptive MOEA proposed solutions that merit further evaluation in all problems investigated. As mentioned in Sections 6.1.3, 6.2.3, 6.3.3, 6.4.3 and 6.5.3 further *in-vitro* investigation is required to understand the behaviour of the proposed designed molecules in real environment. Self-Adaptive MOEA is a useful tool for fine tuning the underlying MOEA to approximate a given problem. In the hands of an experienced user it can prove very powerful, as the expert can guide Self-Adaptive MOEA to the range of settings and the algorithm will propose the ones that tackle the problem better.

[2] https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL14762
[3] https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL83
[4] https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL177798
[5] https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL3545432

1.6 Thesis Organization

The remainder of this dissertation is organized as follows: **Chapter 2** provides an overview of the research area of Scientific Workflow Management Systems. **Chapter 3** provides an overview of the research area of Multi-Objective Evolutionary Algorithms. **Chapter 4** describes the research performed for **Li**fe **S**ciences **I**nformatic**s** platform, at an algorithmic and application level. **Chapter 5** describes the research performed in Multi-Objective Evolutionary Algorithms for Drug De Novo Design, at an algorithmic and application level. **Section 5.1** briefly describes Multi-Objective Evolutionary Graph Algorithm. **Section 5.2** briefly describes Parallel Multi-Objective Evolutionary Graph Algorithm. **Section 5.3** describes the proposed Self-Adaptive Multi-Objective Evolutionary Algorithm in detail. **Chapter 6** focuses on the experiments performed for the evaluation of the proposed Self-Adaptive MOEA method and discuses the experimental design followed and the results obtained from the validation tests performed. **Chapter 7** presents open research questions related and/or inspired by this thesis and outlines directions for future research.

Chapter 2

Scientific Workflow Management Systems

Scientific Workflow Management Systems (SWMSs) are powerful tools with enormous possibilities to facilitate the design and execution process of computational experiments. SWMSs enable scientists to plug together problem solving computational components [22] and implement complex *in silico* experiments, such as the analysis of large datasets that arise from sensors or computer simulations and the design and execution of complicated algorithms requiring multiple computationally intensive steps.

2.1 Scientific Workflow

A Scientific Workflow (SW) is the term used to describe the actions needed to be taken in order to complete a complex scientific task. A SW, as shown in Figure 4, is represented as a directed graph where each node represents a step implemented by a software component. This component can be either the execution of a local program or a remote web service (e.g. a query to a database). The edges of the graph represent either data flow or execution dependencies between nodes [42]. The links coordinate the inputs and outputs of the individual steps, forming the data flow. Control flow links occur when two tasks have no data dependencies and therefore the order must be explicitly defined.

Workflow technology is not new. It has long been adopted by the business community. A Business Workflow (BW) is,

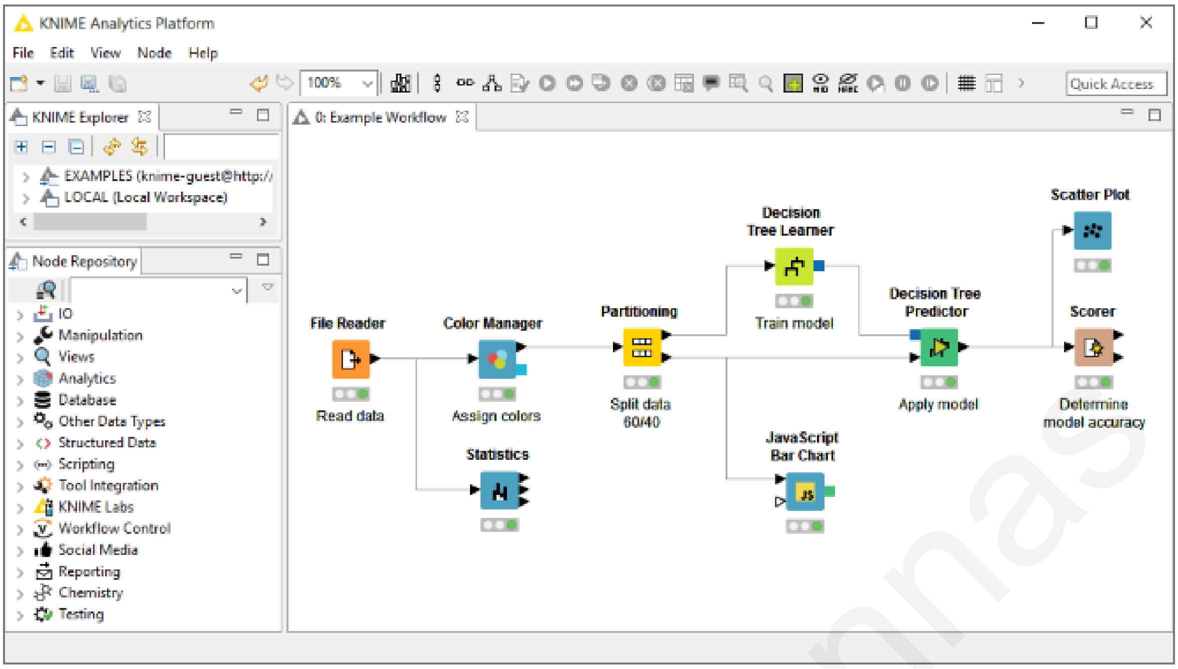


Figure 4: Scientific Workflow (SW) example from Knime [43].

"The computerised facilitation or automation of a business process, in whole or part."

as defined by the Workflow Management Coalition industry consortium in the "Workflow Reference Model" [44]. BW management and business process modelling are mature research areas, whose roots go far back to the early days of office automation systems [25]. However, the term "Scientific Workflow (SW)" became popular after the year 2000, as the existing technology could not support the special characteristics of scientific processes which are data and computationally intensive, highly repetitive and reproducible.

In the case of SW however, experts in the field like Ludascher *et al.* [25] point out that "there seems to be no single set of characteristic features that would uniquely define what a SW is and isn't." Flow control can be considered the most important classification characteristic of SWs. A workflow is either data-flow or control-flow oriented. In control-driven workflows the connections between the tasks represent a transfer of control from one task to the next one. In data-driven workflows connections represent the flow of data from one task to the next one. The workflow representation is focused on data products. As mentioned in [22] most of the current SWs are data-flow oriented as opposed to their predecessors and BWs which are control-flow. According to [45], the reason is that data-flow modelling is the natural way of composing scientific workflows, because they often comprise numerous data transformation steps applying massive parallelism. Another important distinguishing feature of workflows is pipeline parallel processing. A pipeline consists of a collection of steps. Parallelism is achieved by executing these steps simultaneously on different input data sets. The tasks are executed in separate threads, processing input immediately and not waiting for the previous task to complete. The drawback is that pipelined workflows are harder to restart in the case of unforeseen events as the current state of the executed workflow is not as easy to describe and restore [26].

A popular categorization is based on the distinction between high level scientific oriented workflows and lower-level engineering resource oriented (or ”plumbing”) workflows [22]. The first are an implementation of an experimental protocol or a data analysis method, where each task corresponds to the high level tasks of the scientific method. The latter are concerned mostly with the ”plumbing tasks” such as data movement and replication and job management.

A Workflow Model (WM) defines a workflow including its task definition and structure definition. There are two types of WMs, namely abstract and concrete [46]. The abstract model defines a workflow in an abstract form without referring to any resources for task execution. On the contrary in the concrete model the workflow tasks are bound to the designated resources. The user creates the abstract workflow in the workflow modeller component. Mapping the resources is done transparently by the enactment engine to create a concrete executable workflow.

SWs can also be differentiated based on the design focus. In the initial discovery stages of a scientific method, a non-mature workflow is constantly changing while the designer is trying out different approaches and solutions. The design considerations are ease of change and re-usability. Later on, as the workflow becomes mature, it obtains a steady form and can then be used as a production workflow executed on a regular basis. At this point the design considerations shift to speed and efficiency.

2.2 Scientific Workflow Management Systems Paradigms

In theory a SWMS is a combination of a workflow modelling component using an abstract language and a workflow enacting component empowered by an execution engine. In practice a SWMS enables a user to create and then monitor the execution of a workflow by providing the necessary infrastructure. The modelling component enables the user to design, reuse and store WMs while the enacting component invokes, executes and monitors workflow instances [47] deploying them either on a local desktop computer, a web server, or a distributed computing environment such as a cluster or even a cloud infrastructure. Embedded in the workflow design, is the order of the tasks to be executed. The coordination process of this execution is known as orchestration. The execution engine adds the transparency required to allow the domain scientist to model a solution without any concerns of how the solution will be carried through.

This architecture is applied in the Trident SWMS [48]. This Microsoft system allows for independent components for workflow modelling and for execution. Firstly the scientist creates the workflow in an independent workflow composer. Then the workflow is executed in Trident. This is known as centralized execution architecture. Other systems follow a less strict decentralized architecture. For example, in Taverna 2 [49], each processor independently starts its own execution as soon as the input data are available. This allows for inter-processor parallelism as the tasks are executed in separate threads. The need for coordination however exists, so the system offers a façade pattern that relays messages to and from the central monitor. As SWMSs are software environments created specifically for workflows, they encompass a number of functionalities for their management including workflow design, re-engineering, allocation of resources, task scheduling, data movement, data formats, optimizations, execution, monitoring, fault management, analysis, provenance data, storage, collaboration, reuse. Moreover, SWMSs are typically run over middle-ware that provides infrastructure for accessing the applications or resources consumed by the workflow, and facilities like security and access control [47].

2.3 Scientific Workflow Life-cycle

The main design goal of SWMS is to support the workflow life-cycle. Detailed analysis of how each step of the life-cycle can be supported provides an improved understanding of the functionalities that any SWMS must accommodate. The life-cycle of a scientific workflow begins with the Design phase where a new workflow is created either from scratch or from existing workflows. During the following Planning phase the workflow is validated and optimized to user requirements. This phase also includes resource allocation and task scheduling if required. The Execution phase involves invoking and monitoring the workflow, retrieving the data, error handling and keeping measurements. The results of the execution are visualized and tagged in the Analysis phase. Finally, in the Storage phase the workflow is stored along with its provenance data and enabled for sharing [47]. Slightly different scientific workflow life cycles were proposed by experts in the field in [47], [25], [48], [26], [50]. In Figure 5 the scientific workflow life cycle is given as presented in [47].

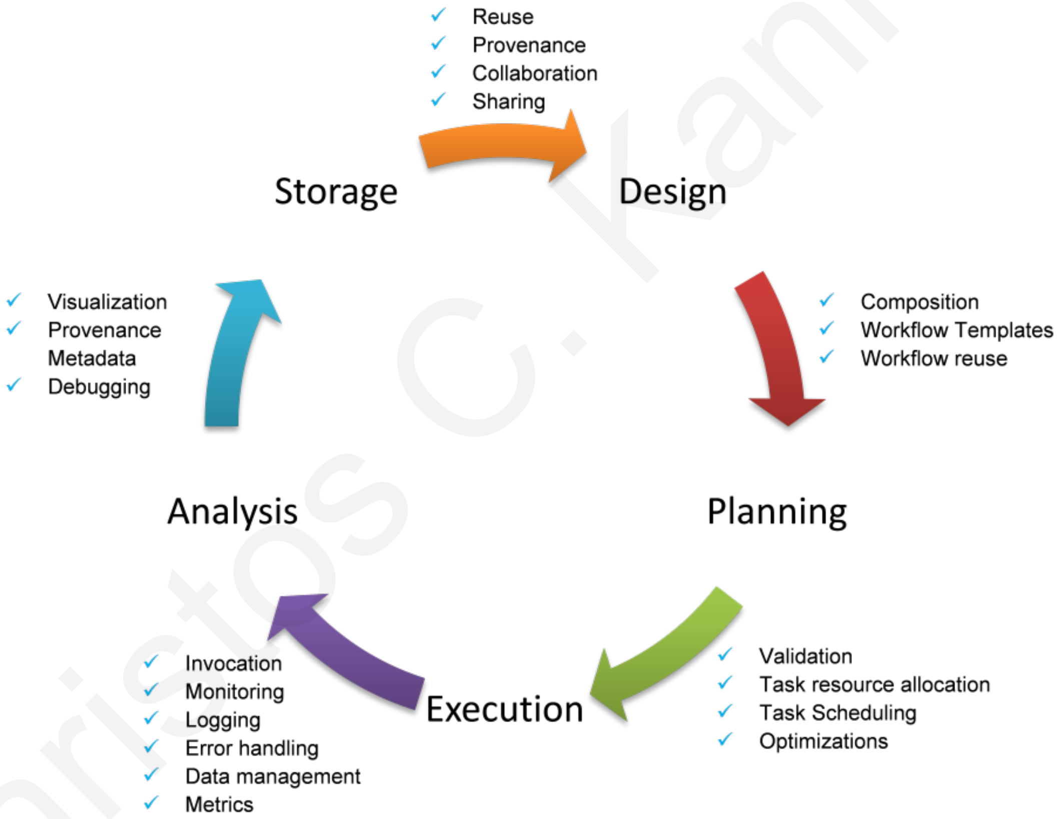


Figure 5: Scientific Workflow (SW) life cycle as proposed by Goble *et al.* [47]

2.4 Scientific Workflow Management Systems for Life Sciences Review

The field of SWMSs has been receiving considerable interest in recent years. Consequently, a number of implementations have been reported and several reviews of such systems have been published. Early on, in 2005, Yu and Buyya [46] presented a taxonomy of grid workflow systems. In

2006, Taylor *et al.* [51] published a book on E-Science workflows, presenting several systems and defining research questions. Tiwari and Sekhar [52] surveyed workflow systems for life sciences. The research questions set down at the National Science Foundation Workshop on Scientific Workflows of 2006 were recorded by Gil *et al.* [53]. In 2008, Barker and van Hemert [22], presented a concise survey of existing workflow technology from the business and scientific domain and made a number of key suggestions. At the same year Curcin and Ghanem [54] reviewed six systems considered state of the art in the field. McPhillips *et al.* [23] prepared a list of Desiderata for scientific workflow systems for scientists. Finally, Goble *et al.* [47] presented the challenges to be met by the advancing workflow technology. In 2009, Ludascher *et al.* [25] in his survey compared SWs to the well-established BWs. At the same year, the same author provides an overview of the characteristic features of scientific workflows and outlines their life cycle [26]. Deelman *et al.* [50] extracts a taxonomy of features from the end users view for the current scientific workflow systems. Sonntag *et al.* in their work in 2011 [48], after reviewing contemporary systems, proposed a conceptual architecture for SW systems based on BW systems, an approach encouraged by the Sixth International Workshop on Scientific Workflows (SWF 2011). This section provides an updated review of the main, most popular SWMSs in order to present the current state of the art in the field.

The list of different workflow management tools used routinely is considerably large, exceeding 50 items [47]. This list includes popular SWMSs like Taverna [49], [55], [56], Triana [57], Kepler [58], Pegasus [59], KNIME [60], [61], Galaxy [62], [63]. [64], Pipeline Pilot [65], InforSense KDE [66] and Microsoft Trident [67] but also BioWBI [68], GridBus [69], ICENI [70], Magenta [71], GridNexus [72], ASKALON [73] and others.

Table 1 presents a snapshot of the main popular representatives of SWMSs used for life sciences and their main characteristics [27]. There are three open source and two commercial SWMSs. The majority of them are desktop based software only Galaxy is web based. Knime and Taverna provide a variety of chem[o]informatics tools in addition to tools for other domains, Galaxy is focused in Bioinformatics. Inforsence and Pipeline Pilot are focused on chemistry and biology oriented domains.

Table 1: List of popular Scientific Workflow Applications for Life Sciences

	Applications	Technology	Scientific Field(s)
Open Source	Taverna	Java	Bioinformatics, Chemistry, Astronomy, Data Mining, Text Mining, Music
	Galaxy	Python	Life Sciences, Bioinformatics
	Knime	Java	Life Sciences, Chem[o]informatics, Bioinformatics, High Performance Data Analysis
Commercial	Inforsence/DiscoveryNet		Life Sciences, Healthcare, Environmental Monitoring, Geo-hazard Modelling
	Pipeline Pilot		Biology, Chemistry, Material Science

The major players in the domain of Life Sciences are KNIME [43] and Pipeline Pilot [65]. KNIME is a free suite but their business model provides licensing for enterprise wide components and servers. Pipeline Pilot is an expensive commercial suite. KNIME is a predictive analytics suite that has tools suited for specific domains including tools for the Life Sciences that were developed by pharmaceutical companies and communities, some of which are provided on a licensing scheme and the rest are provided for free. Pipeline Pilot on the other hand was developed with Life Sciences in mind, initially focused on chem[o]informatics and has expanded into other Life Sciences sub-domains and chemistry related domains. Both were developed as desktop suites with client-server extensions for use in an enterprise environment [74].

These two major players show the trends of the market. They are focused on providing desktop based platforms to scientists and non-scientists. They have an enterprise approach of an in-house client server model. As such there is a gap in the web accessed platforms where the only player is Galaxy and its derivatives.

The available tools and functionality of the systems mentioned in Table 1 enable the design of workflows/pipelines consisting of numerous tools, which some refer to as visual programming. This visual programming approach is an emerging trend of the way data scientists will work. They provide a lot of tools for creating virtual screening workflows of different complexity and functionality. However the functionality of *in silico* molecular design as a dedicated module is rather limited. This can be addressed by creating complex workflows, using their features in a unique and innovative way.

Chapter 3

Multi-Objective Evolutionary Algorithms

Multi-Objective Evolutionary Algorithms (MOEAs) [75], [76] have become an important research field in recent years. Algorithmic advancements combined with the Multi-Objective (MO) nature of many real life problems have motivated researchers to explore, and often adopt, MOEA based methods. In contrast to Single Objective Optimization Problems (SOOPs) where a single, optimal solution suffices, Multi-Objective Optimization Problems (MOOPs) have a set of equivalent solutions that represent different compromises among the various objectives guiding the search. This set of solutions is called the Pareto Front (PF) whereas intermediate solution sets produced during the optimization search are referred to as Pareto Approximations (PAps). MOEAs aim to minimize the difference between the final PAp produced and the true PF of a MOOP. The solutions of a population that comprise its PAp set are characterized by non-domination, i.e, the lack of any other solutions that are better than them in all the objectives. The Pareto ranking mechanism identifies non-dominated solutions and ranks all individuals according to the number of solutions that dominate them [77], [78]. The presence of multiple objectives, typically characterized by complex, multi-modal search spaces, as well the need for processes such as Pareto ranking, increase the complexity of MOOPs and thus the computational resources required to obtain solutions of good quality. Evolutionary Algorithms (EAs) are an increasingly popular population based meta-heuristic optimization method inspired by nature.

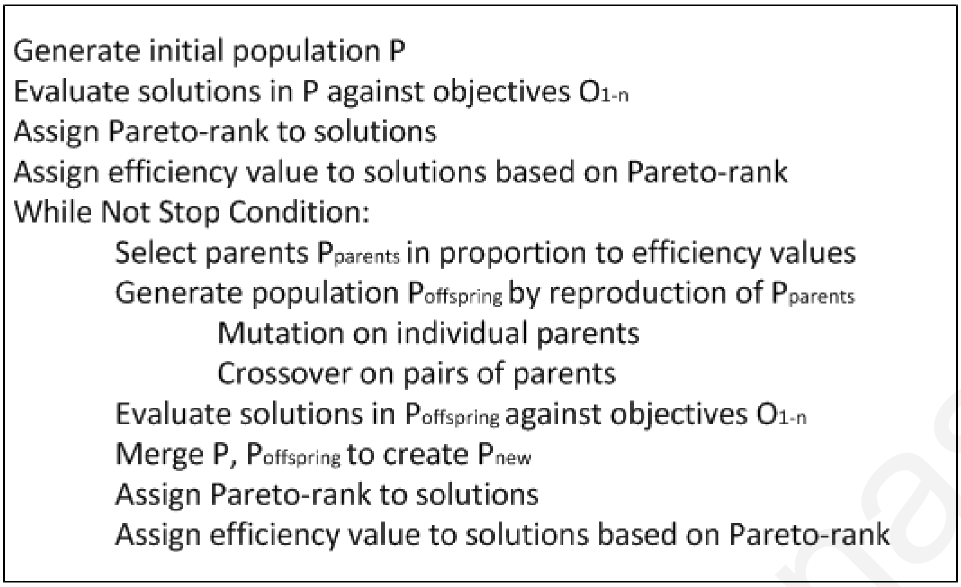
3.1 Introduction

A MOOP involves several conflicting objectives and has a set of Pareto optimal solutions. Among the most popular algorithms used in optimization, including Pareto based MOOP approaches, are EAs [75]. Intensive research efforts during the last two decades have focused on the application of EA methodology to MOOPs with considerable advances being reported in various fields [75]. By evolving a population of solutions, Multi-Objective Evolutionary Algorithms (MOEAs) are able to approximate the Pareto optimal set in a single run. MOEAs have attracted a lot of research effort during the last twenty years, and they are still one of the hottest research areas in the field of Evolutionary Computation (EC).

The popularity of MOEAs is probably due to some inherent algorithmic characteristics. Namely, the population based approach enables the simultaneous search of multiple search space regions and thus the identification of numerous Pareto solutions in a single run. Additionally, EAs impose no constraints on the morphology of the search space and are therefore suitable for complex, multi-modal surfaces such as the ones typically produced by MOOP problems. Algorithmically, MOEAs are an extension of traditional EAs that can address multiple objectives simultaneously by the addition of appropriate components such as Pareto based selection that incorporates fitness assessment on multiple objectives, calculation of domination relations and Pareto rank and definition of a scalar efficiency value for each solution, and the techniques of niching and elitism aiming to maintain population diversity and avoid good solution loss [79]. Figure 6 outlines the main steps of a simple MOEA.

There has been a growing interest in applying EAs to deal with MOOPs since Schaffer's seminal work [80]. By May 2016, more than 10181 publications have been published on Evolutionary Multi-Objective Optimizations (EMOOs). Among these papers, 84.13% (8565) have been published in the last 13 years (2003 - 2016), 46.05% (4689) are journal papers and 37.25% (3793) are conference papers[1] .

[1] The statistical data is based on the paper repository in the EMOO web site, http://delta.cs.cinvestav.mx/~ccoello/EMOO/, which is maintained by Professor Coello Coello.



```
Generate initial population P
Evaluate solutions in P against objectives $O_{1-n}$
Assign Pareto-rank to solutions
Assign efficiency value to solutions based on Pareto-rank
While Not Stop Condition:
        Select parents $P_{parents}$ in proportion to efficiency values
        Generate population $P_{offspring}$ by reproduction of $P_{parents}$
                Mutation on individual parents
                Crossover on pairs of parents
        Evaluate solutions in $P_{offspring}$ against objectives $O_{1-n}$
        Merge P, $P_{offspring}$ to create $P_{new}$
        Assign Pareto-rank to solutions
        Assign efficiency value to solutions based on Pareto-rank
```

Figure 6: A typical Multi-Objective Evolutionary Algorithm (MOEA).

The most recent state of the art survey on MOEAs was published in 2011 by Zhou *et al.* [81], and according to it the key issues that distinguish MOEAs are: (a) Algorithmic framework, (b) Selection and population updating, and (c) Reproduction.

3.2 Algorithmic Frameworks

Algorithmic framework is a key issue when designing a MOEA. Below there is a brief description of the existing algorithmic frameworks and MOEAs representatives.

3.2.1 Pareto non-domination based Multi-Objective Evolutionary Algorithms

The Multi-Objective Genetic Algorithm (MOGA) uses a ranking scheme in which the rank of an individual corresponds to the number of individuals in the current population by which it is dominated. I.e. an individual x_i at generation t which is dominated by $p_i^{(t)}$ individuals, in the current population, has a rank given by Equation 1. All non-dominated individuals are assigned rank 1, while dominated individuals are penalized according to the population density of the corresponding region of the trade-off surface [82].

$$rank(x_i, t) = 1 + p_i^{(t)} \tag{1}$$

In MOGA fitness assignment is performed in the following way: (a) Sort population according to calculated rank, (b) Assign fitness to individuals by interpolating from best (rank 1) to the worst (rank $n \leq N$), and (c) Average the fitnesses of the individuals with the same rank, so that all of them are sampled at the same rate[2].

The Non-dominated Sorting Genetic Algorithm (NSGA) modifies the Pareto ranking and the efficiency calculation step of the algorithm using the non-dominated sorting concept [83]. In NSGA the population is classified into layers, or waves, of non-dominated sets. The process successively defines the non-dominated set of the population, removes its members from the current population and iterates until all solutions have been taken into account. Fitness sharing and solution sampling are performed at the non-dominated layer level starting from the globally non-dominated solution level. Fitness values of solutions in successive layers are reduced to be less than the worst fitness value of the previous layer.

The Niched Pareto Genetic Algorithm (NPGA) method [84] is a further extension of the NSGA where the selection step is based on a modified tournament-based method that uses a larger subset of the population and shared efficiency values of the individuals.

In the original version of NSGA, selection is performed using a stochastic-remainder wheel-like operator while in an updated elitist version, named NSGA-II, selection is performed by choosing the best solutions from a population combining both parents and offspring [85]. The NSGA-II conducts niching through the use of a crowding distance calculated for each solution, used to maintain population diversity during selection by ensuring that selected solutions are sufficiently apart. This keeps the population diverse and helps the algorithm to explore the fitness landscape [79]. The NSGA-II algorithm successfully addresses some of the shortcomings of MOGA, which may introduce a bias towards certain solutions in the search space due to the nature of its rank-based fitness assignment method and thereby allow solutions with substantially better performance at an iteration to dominate the population of later generations, and succeeds in preserving the diversity of the population.

[2] This procedure keeps the global population fitness constant while maintaining appropriate selective pressure, as defined by the function used.

The recently proposed updated version of NSGA-II, NSGA-III, by Jain and Deb [38], is based on the supply of a set of reference points and demonstrated its working in three to 15-objective optimization problems.

3.2.2 Decomposition based Multi-Objective Evolutionary Algorithms

Multi-Objective Evolutionary Algorithm based on Decomposition (MOEAD) [86] is based on conventional aggregation approaches in which an MOOP is decomposed into a number of SOOPs. The objective of each SOOP, also called a subproblem, is a weighted aggregation of the individual objectives. Neighborhood relations among these subproblems are defined based on the distances between their aggregation weight vectors. Each subproblem is optimized in the MOEAD by using information mainly from its neighboring subproblems.

In a simple version of the MOEAD, each individual subproblem keeps one solution in its memory, which could be the best solution found so far for the subproblem. For each subproblem, the algorithm generates a new solution by performing genetic operators on several solutions from its neighbouring subproblems, and updates its memory if the new solution is better than the old one for the subproblem. A subproblem also passes its newly generated solution on to some (or all) of its neighbouring subproblems, which will update their current solutions if the received solution is better. A major advantage of MOEADs is that a single objective local search can be used in each subproblem in a natural way since its task is for optimizing a single objective subproblem.

Recently several improvements on MOEADs have been made. Li and Zhang [87] suggested using two different neighbourhood structures for balancing exploitation and exploration. Zhang *et al.* [88] proposed a scheme for dynamically allocating computational efforts to different subproblems in an MOEAD in order to reduce the overall cost and improve the algorithm performance. This implementation of MOEAD is efficient and effective and has won the Congress on Evolutionary Computation (CEC) 2009 MOEA competition. Nebro and Durillo [89] developed a thread-based parallel version of MOEAD, which can be executed on multi-core computers. Palmers et al. [90]

proposed an implementation of MOEAD in which each subproblem records more than one solution. Ishibuchi *et al.* [91] proposed using different aggregation functions at different search stages.

3.2.3 Preference based Multi-Objective Evolutionary Algorithms

In the majority of cases where Pareto domination is used there is the problem of too many or even infinite optimal solutions residing in the PF. A solution to this is to use a decision manager that has the job to find the preferred solutions within the PF. In order to find these solutions the decision manager has to use some reference information for guidance. The methods to provide the preference information in MOOP can be classified as apriori, posteriori and interactive methods [92].

In apriori method, preference information is given by the decision manager before the solution process. An MOOP can be converted into an SOOP. Then, a single objective solver is applied to find the desired Pareto optimal solution.

A posteriori method uses the decision manager's preference information after the search process. A well distributed approximation of the PF is first obtained. Then, the decision manager selects the most preferred solutions based on the preferences.

In an interactive method, the intermediate search results are presented to the decision manager to investigate; then the decision manager can understand the problem better and provide more preference information for guiding the search.

The earliest attempts on MOEAs based on the decision manager's preference were made by Fonseca and Fleming [82] and Tanino *et al.* [93] in 1993. In these algorithms, the rank of the members of a population is determined by both the Pareto dominance and the preference information from the decision manager. Greenwood *et al.* [94] used value functions to rank the population, and preference information was also used in the survival criteria.

Sakawa and Kato [95] proposed a fuzzy approach to represent preference in the form of reference points. The decision manager is asked to specify a new reference point until satisfactory results are reached. Phelps and Kksalan [96] compared a pair of individuals in terms of their fitness values based on the decision manager's preferences at each iteration. A single substitute objective defined by weighted sum of objectives is used for some generations.

Branke and Deb [97] incorporated the preference information into NSGA-II by modifying the definition of dominance and using a biased crowding distance based on weights. Deb *et al.* [98] further considered the use of reference points to determine preference information. A guided dominance scheme and a biased crowding scheme are also suggested. Deb *et al.* [99] suggested an interactive MOEA based on reference directions. The decision manager provides one or more reference directions to guide the search towards the region of preferred solution.

Deb and Chaudhuri [100] proposed an interactive decision support system, in which a number of existing multi-objective optimization and classical decision making methods can be appropriately adopted for generating solutions in the regions of interest in the Pareto Set (PS).

Li and Silva [101] developed an improved version of an MOEAD combined with Simulated Annealing (SA). The weights can be adaptively changed by the decision manager according to the location of solutions in the current population. The fitness functions with modified weights can guide the search towards different parts of the PF during the search. It can be viewed as an interactive MOEA.

Sanchis *et al.* [102] proposed an MOEA integrated with apriori preferences, which were generated by applying the principle of physical programming. In this algorithm, the preferences are expressed by partitioning the objective space into several levels. The preference functions are built to reflect the decision manager's interests and to use meaningful parameters for each objective. The designer's expert knowledge can be translated into preferences for design objectives. A single objective is automatically built and no weight selection is performed.

Deb *et al.* [103] proposed a progressively interactive MOEA. An approximate value function is progressively generated after every few generations. Periodically, several non dominated points found so far are provided to the decision manager. Based on the decision manager's preference information, all these points are ranked from the worst to the best. Then, a suitable polynomial value function is constructed by solving an SOOP.

Rachmawati and Srinivasan [104] proposed a preference based MOEA to find the knee region in the PF, which is visually a convex bulge in the front. The preference based focus is achieved by optimizing a set of linear weighted sums of the original objectives, and control of the extent of the focus is attained by careful selection of the weight set based on a user specified parameter. The fitness scheme could be easily adopted in any Pareto based MOEA with little additional computational cost.

Thiele *et al.* [105] used the decision manager's preferences expressed interactively in the form of reference points. The information is used in an EA to generate a new population by combining the fitness function and an achievement scalarization function. The selection based on the utility functions with the modified parameters is expected to lead the search to focus on the most interesting parts of the PS. In multi-objective optimization, achievement scalarization functions are widely used to project a given reference point on to the PS.

3.2.4 Indicator based Multi-Objective Evolutionary Algorithms

The quality of an approximated PF could be measured by a scalar indicator such as generational distance and hypervolume. Indicator-based MOEAs use an indicator to guide the search, particularly to perform solution selection.

Zitzler and Künzli [106] first suggested a general Indicator-Based Evolutionary Algorithm (IBEA). This approach uses an arbitrary indicator to compare a pair of candidate solutions. In the IBEA, any additional diversity preservation mechanism such as fitness sharing, is no longer required. In comparison to other MOEAs, the IBEA only compares pairs of individuals instead of entire approximation sets.

Basseur and Zitzler [107] proposed an indicator based model for handling uncertainty, in which each solution is assigned a probability in the objective space. In an uncertain environment, some methods for computing expected indicator values are discussed, and several variants of their ϵ-indicator based model are suggested and empirically investigated.

Brockhoff and Zitzler [108] proposed a general approach to incorporate objective reduction techniques into hypervolume based algorithms. Different objective reduction strategies are studied for improving the performance of hypervolume based MOEAs.

Bader and Zitzler [109] further investigated the robustness of hypervolume based multi-objective search methods. Three existing approaches for handling robustness in the area of evolutionary computing, modifying the objective functions, additional objectives, and additional robustness constraints, are integrated into a multi-objective hypervolume based search. An extension of the hypervolume indicator is also proposed for robust MOOP.

A year later, Bader and Zitzler [110] suggested a fast hypervolume based Many-Objective Evolutionary Algorithm (MaOEA) for Many-Objective Optimization Problem (MaOOP). To reduce the computational overhead in hypervolume computation, a fast method based on Monte Carlo simulations is proposed to estimate the hypervolume value of an approximation set. Therefore, the proposed hypervolume based MaOEA may be applied to problems with many objectives.

3.2.5 Memetic Multi-Objective Evolutionary Algorithms

MOEAs incorporating local search methods have also been investigated [111], [112], [113], [114], [115], [116], [117], [118], [119]. These algorithms are known as Memetic Multi-Objective Evolutionary Algorithms (mMOEAs). mMOEAs are able to offer not only better speed of convergence to the evolutionary approach, but also better accuracy for the final solutions [111]. Ishibuchi and Murata proposed one of the first mMOEAs [112]. The algorithm uses a local search method after classical variation operators are applied, and a randomly drawn scalar function to assign fitness is used for parent selection.

According to Adra *et al.* [120], the best solutions found in each generation are improved by a local search method in the objective space, and the improved solutions are then mapped back to the decision space to predict the corresponding decision variables. A local search operator is used to generate offspring solutions [121]. Similar ideas are also mentioned in [122] and [123].

Knowles and Corne [124] proposed a memetic Pareto archived evolution strategy to solve MOOPs. The algorithm introduces a Pareto ranking based selection method and couples it with a partition scheme in objective space. It uses two different archives to save non-dominated solutions.

Jaszkiewicz [125] proposed a Multi-Objective Genetic Local Search (MOGLS) algorithm for the MOOP 0/1 knapsack problem. At each iteration, a weighted scalarization function is used as the fitness function during selection. The weights are generated in a random way. The mating population in the MOGLS consists of a few individuals selected from the current population in terms of the current scalarization function. An offspring solution is then produced by recombining members in the mating population. A local search procedure is followed to improve the quality of the offspring solution. The current population and an external population including only non-dominated solutions are updated by the improved solutions obtained in the local search.

Caponio and Neri [126] proposed the cross dominant mMOEA, making use of two local search engines to balance the global search and the local search. The choice of local search engines is decided by using the parameter of mutual dominance between non-dominated solutions belonging to consecutive generations.

A memetic version of Co-evolutionary Multi-Objective Differential Evolution (CoMODE) is presented in [127]. In this approach, the population of solutions and promising search directions are evolved synchronously. A local search method is applied to a portion of the population after each iteration.

A Memetic Algorithm based on Differential Evolution (MADE) was proposed by Qian *et al.* [114] to handle Multi-Objective No-wait Flow-Shop Scheduling Problems (MNFSSPs). This algorithm uses several local searchers developed according to the landscape of an MNFSSP to enhance the local exploitation.

Wanner *et al.* [118] employed a local search optimizer as an additional operator in MOEA. The local search technique is able to find more precise estimation of the Pareto optimal surface with a reduced number of function evaluations. Ishibuchi *et al.* [119] studied the use of biased neighbourhood structures for a local search in mMOEAs. The methods assign higher probabilities to more promising neighbours in order to improve the search ability of mMOEAs. More recently, Lara *et al.* [111] investigated a new local search strategy called the Hill Climber with Sidestep (HCS) for mMOEAs. The new point-wise local search procedure is able to move both toward (using hill climber techniques) and along (sidestep) the PS.

MOEAD [86] also belongs to the class of mMOEAs. It optimizes multiple subproblems. Each solution is associated with one weighted scalarization function. A local search procedure can be called for improving a solution. Since MOEAD is a general framework, different heuristic search methods can serve as the local search component. Sanchis *et al.* [102] proposed the use of a SA to improve the current solution of each subproblem. Li and Landa-Silva [128] proposed to optimize each subproblem by the Greedy Randomized Adaptive Search Procedure (GRASP).

3.2.6 Co-evolution based Multi-Objective Evolutionary Algorithms

Co-evolution can be regarded as evolving multiple sub-populations simultaneously to tackle a complicated problem. Algorithms using an archive strategy, such as [129], thus fall into this category because they evolve a population and an archive at the same time to approximate the PF of an MOOP.

However, there is another explanation of co-evolution by using the idea of divide and conquer. Following this idea, a co-evolutionary algorithm breaks down a problem into a set of subproblems in the level of individual coding and evolves multiple sub-populations. Tan *et al.* [130], Goh *et al.*[131] and Goh *et al.*[132] adopt this idea. Among them, the sub-populations are competitive and/or cooperative with each other and the components from different sub-populations are combined to form a complete solution.

3.3 Selection and population updating

The selection of solutions for the next generation plays a key role in MOEAs. The main difference between EAs for SOOPs and MOOPss in algorithm components is the selection procedure. An EA for SOOPs can be directly applied to MOOPs by replacing the selection component. In SOOP, there naturally exists a complete order to differentiate all feasible solutions, i.e., for any two feasible solutions x and y, either $f(x) \leq f(y)$ or $f(y) \leq f(x)$. However, in MOOP, the Pareto dominance, $<$, only defines a partial order in the objective space, and not all the feasible solutions can be compared to each other.

Since the Pareto dominance cannot be naturally used to select solutions, additional strategies need to be considered. The design of selection operators has been gaining significant attention in MOEAs. The previous major works on selection follow the idea of defining complete orders over individuals, and recently some works follow the idea of defining complete orders over populations.

3.3.1 Complete orders over individuals

Since Pareto domination only defines a partial order, extending the partial order to a complete order becomes a natural way to differentiate solutions. To this end, a two-stage strategy is usually employed. In the first stage, a population is partitioned into several clusters by Pareto dominance. Each individual x will be assigned an integer value, called *rank*, and denoted as x_{rnk}. Those with the same *rank* value are equal to each other, and smaller *rank* is preferred. In the second stage, individuals with the same *rank* are further differentiated by assigning each individual a real value, called *density*, and denoted as x_{den}. Those with lower *density* values are preferred. A complete order, denoted as $\prec i$, can thus be defined as follows:

$$\exists x \prec i \, \exists y \iff (x_{rnk} < y_{rnk}) \vee (x_{rnk} = y_{rnk} \wedge x_{den} < y_{den})$$

Domination rank [83], domination count [82], and domination strength [133] are usually used to assign rank values. The widely used methods for density estimation include the niching and fitness sharing strategy [82], crowding distance [85], k-Nearest Neighbours (kNN) method [134], fast sorting [135], and gridding and ϵ-domination method [136], [137],[138], [139] and [140].

A variety of methods [123], [129], [141], [142], [143] and the extension of Pareto domination to fuzzy domination [144], [145] have been proposed to improve the algorithmic performance.

Some new data structures have been proposed to improve the sorting performance [146], [147], because there are many redundant comparisons between individuals in the rank assignment procedure if the definition of Pareto domination is to be followed.

3.3.2 Complete orders over populations

In an MOEA, populations are actually updated from one generation to another. Selection mechanisms based on performance indicators define a complete order over populations. Let $I(P)$ be a quality indicator which assigns a real value to a non-dominated population P. A full order, $\prec p$ is defined as follows:

$$P \prec pQ \iff I(P) < I(Q)$$

where a smaller value of indicator $I(P)$ is preferred.

The idea of using performance to guide the selection was first proposed by Fleischer in [148]. Huband *et al.* [149] proposed the first MOEA with a hypervolume guided selection procedure. Indicator based selection has since then been widely applied in MOEAs [150] and [151]. Zitzler and Künzli generalized the idea and proposed an indicator based MOEA [106]. These methods are called indicator based MOEAs, and they are discussed in Section 3.2.4. A major disadvantage with this kind of selection is that it might be time consuming. More work is needed to improve the efficiency.

3.4 Reproduction

Conventional reproduction operators designed for SOOP EAs could be directly used in MOEAs. The optimal structures of SOOP and MOOP are quite different, i.e., an isolated point or several points with the same objective value in SOOP and a solution set in MOOP. The operators designed for SOOP might not be suitable for MOOP. It has been observed that some widely used reproduction operators did not work well for rotated problems [152]. This difference should be emphasized in MOEA. The characteristics and/or problem specific knowledge should be considered in designing reproduction operators for MOOP [153], [154], [155], [156].

3.4.1 Differential Evolution based approaches

The Differential Evolution (DE) algorithm [157], [158], was introduced by Storn and Price. The DE algorithm was originally designed for SOOP. However, it has since attracted much attention in MOOP because of its simplicity to implement and efficiency for solving problems.

A Pareto frontier differential evolution (PDE) algorithm was proposed by Sarker and Abbass [159]. The major modifications are (a) the step length parameter F is randomly sampled from a Gaussian distribution $N(0, 1)$, and (b) the parents are from the non-dominated set. To find a uniformly distributed, near complete, and near optimal PF, a Multi-Objective DE based on Pareto adaptive dominance and orthogonal design was proposed by Gong and Cai in [160]. A MO DE algorithm with diversity enhancement strategies was proposed by Qu and Suganthan in [135].

The DE algorithm has also been extended to tackle discrete or mixed continuous and discrete MOOPs. A MO DE algorithm was proposed by Alatas, *et al.* in [161] for mining numeric association rules. A memetic algorithm based on DE was proposed by Qian *et al.* in [114] to deal with MNFSSPs.

Since the DE algorithm has two control parameters which are not easy to set properly, self adaptation has also attracted much attention recently. The two control parameters are randomly picked up from predefined ranges. Wang *et al.* in [143] proposed the use of a crowding entropy based diversity measure to maintain an elitist archive.

3.4.2 Immune based approaches

Due to the clonal selection and affinity maturation by hypermutation, the immune system is able to adapt B-cells to new types of antigens. By simulating this phenomenon, artificial Immune Iystems were proposed to deal with optimization problems [162]. Recently, Immune Systems have been extended from SOOP to MOOP. In Multi-Objective Immune Systems (MOISs), clonal selections based on Pareto dominance are usually used to select promising solutions while crossover and mutation operators are widely used to generate new trial solutions.

Most of the MOISs focus on static problems. Coello and Cortes in [163] proposed the use of two mutation operators to mutate antibodies with different qualities. An archive is used to store elitist solutions to approximate the PF. A hybrid MO algorithm based on an Immune System and bacterial optimization was proposed to deal with bi-objective no-wait flowshop scheduling problems, proposed by Tavakkoli *et al.* [164]. A linear combination method was applied to generate antibodies which are improved by using bacterial optimization operations. A non-dominated neighbour immune algorithm was proposed for MOOP by Gong *et al.* in [141]. The selection strategy emphasizes more on less crowded solutions. A hybrid immune MOOP algorithm based on a clonal selection principle was proposed by Chen *et al.* in [115]. In this approach, Gaussian and polynomial mutations are adaptively applied to mutate the new trial solutions after crossover. The selection procedure proposed by Gong *et al.* in [141] is used to update the population directly. A MOIS based on a multiple affinity model was proposed by Hu in [165].

Some immune algorithms have been applied to dynamic and uncertain optimization problems. A MOIS was proposed by Zhang in [166] to deal with dynamic MOOP with constraints. A MOIS was presented by Zuo *et al.* in [167] to find Pareto optimal robust solutions for bi-objective scheduling problems.

3.4.3 Particle Swarm Optimization based approaches

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995 [168] and [169], inspired by the social behaviour of bird flocking or fish schooling.

Moore and Chapman extended this idea to MOOP in 1999 [170]. Since PSO cannot be directly applied to MOOP, there are two issues to be considered when extending PSO to MOOP. The first one is how to select the global and local best particles (leaders) to guide the search of a particle. The second is how to maintain good points found so far. For the latter, a secondary population is usually used to maintain the non-dominated solutions.

In [171], the particles are clustered into swarms, all particles that have their best position in the same cluster form a swarm. In [172], a tournament niche method is introduced to select the global best particle, and the local best particle is updated by the Pareto dominance. In [173], the global best particle is selected from the non-dominated solutions with a roulette wheel selection in which the density values are used as fitness. The SA control parameter is also considered. In [174], a preference order, a generalization of Pareto dominance, is introduced to rank all the particles and thus to identify the global best particle.

Three EA-PSO hybrid algorithms were proposed in [175]. The fitness assignment strategy is based on that of Strength Pareto Evolutionary Algorithm 2 (SPEA2) [134]. The global best particle is selected from the external archive by a tournament selection, and the neighbourhood best particle is selected as the one with lowest strength Pareto fitness.

A MO-PSO was designed to tackle MO mixed-model assembly line sequencing problems in [176]. To this end, a coding strategy and a local search are introduced. The global best particle is the non-dominated solution in the archive with the highest crowding distance in the archive.

A multiple swarms algorithm was proposed by Leong and Yen in [116]. Several components, such as cell-based rank density estimation, population growing and declining strategies, and adaptive local search, are designed to improve the algorithmic performance. A leader selection was proposed to assign a leader for each group.

Coello *et al.* in [139] proposed the use of an archive to maintain the non-dominated solutions found so far, and the use of a mutation operator to keep the population diversity. To choose a global best particle, the non-dominated ones in sparse areas are emphasized.

In [177], a fuzzy clustering-based PSO was proposed to tackle electrical power dispatch problems. A fuzzy clustering technique is applied to maintain the external archive. A self-adaptive mutation operator is also used to generate new trial solutions. A niching mechanism is designed to find the global best particle for each particle and thus to emphasize less explored areas. Finally, a fuzzy decision rule is used to assist decision making.

A Multi-Objective Comprehensive Learning Particle Swarm Optimizer (MOCLPSO) was presented by Huang *et al.* in [178]. MOCLPSO uses a learning strategy whereby all other particles' historical best information is used to update a particle's velocity. This strategy enables the diversity of the swarm to be preserved to discourage premature convergence.

Two-Local-Best Multi-Objective Particle Swarm Optimization (2LB-MOPSO) technique was proposed by Zhao and Suganthan in [179]. Different from canonical Multi-Objective PSO, 2LB-MOPSO uses two local bests instead of one personal best and one global best to lead each particle.

The two local bests are selected to be close to each other in order to enhance the local search ability of the algorithm. Compared to the canonical Multi-Objective PSO, 2LB-MOPSO shows great advantages in convergence speed and fine-searching ability.

In [180], PSO is used in the MOEAD framework. Each particle is responsible for solving one subproblem.

More works on MO-PSO are presented in [181].

3.4.4 Probabilistic Model based approaches

The main feature of these algorithms is that they do not use traditional crossover or mutation operators to generate new solutions. Instead, they explicitly extract global statistical information from their previous search and build a probability distribution model of promising solutions. Based on the extracted information, new solutions are sampled from the model thus built. Compared to traditional EA methods, they emphasize the population distribution information rather than the individual location information. The key issues in these methods include model selection before executing the algorithm and model building and sampling in the running process. The following methods share the above basic ideas and they differ from each other on origins.

Ant Colony Optimization (ACO) [182], introduced by Dorigo in 1992, takes inspiration from the behaviour of real ant colonies and is used to solve optimization problems. Ants deposit pheromone on the ground in order to mark some favourable paths followed by other members of the colony with higher probability. ACO exploits a similar mechanism by constructing a probability matrix, named the pheromone model, to denote the probability to choose an edge in a graph and thus sampling new solutions. The structure of ACO probability model makes it a natural choice for discrete optimization. In the case of MOOP, ACO has been applied to Travelling Salesman Problems [183, 184], Vehicle Routing Problems [185], Flow-Shop Scheduling Problems [186], Portfolio Selection [187, 188] and others.

The Cross Entropy (CE) method [189] was proposed by Rubinstein and Kroese, originated from the field of rare event simulation involving the estimation of parameters for a number of probability distributions associated with some rare events. CE methods iteratively generate sample points from the probability model and update the model parameters on the basis of the data. Currently, however, there are not many reports on applying CE for MOOP. In [190], a CE-based approach was proposed for MOOPs. In the approach, a population is partitioned into several clusters, and a CE method with a Gaussian model is utilized in each cluster.

The Quantum-inspired Genetic Algorithm (QGA) was first proposed by Han and Kim in 2000 [191]. The QGA simulates the quantum mechanism and uses a Q-bit vector to represent a solution. The Q-bit vector actually denotes probability distributions of all Q-bits to be 0 or 1. A quantum gate is used to generate new individuals. IA MO-QGA was proposed, by Wei *et al.* in [192], to deal with hardware - software co-synthesis problems in embedded systems. Another version of the MO-QGA was proposed to deal with flow-shop scheduling problems by Li and Wang [193].

The Estimation of Distribution Algorithm (EDA) was first introduced by Mühlenbein and Paaβ in 1996 [194]. Most EDAs aim to discover the variable linkage information from the population to benefit offspring generation. To this end, different models with univariate, bivariate, and/or multivariate variable linkages have been widely studied [195]. Depending on the models used, EDAs are suitable for both combinatorial and continuous optimization. In the case of continuous MOOP, Okabe *et al.* [196] proposed a Voronoi model-based method. Bosman and Thierens [197] proposed an EDA method based on a mixture univariate Gaussian model. Dong and Yao [198] proposed a multivariate Gaussian model-based method. Igel et al. [151] extended the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) for dealing with MOOPs. In the case of combinatorial MOOP, Laumanns *et al.* [199] proposed a Bayesian network-based method for knapsack problems. Pelikan *et al.* [200] designed a method with hierarchical Bayesian networks to study building boxes for binary coding problems.

The PF and PS of a continuous MOOP are piecewise continuous (m 1)-dimensional manifolds under mild conditions [201]. Based on this regularity property, Zhang et al. [155] proposed a Regularity Model-based Multi-Objective Estimation of Distribution Algorithm (RM-MEDA) for continuous MOOPs with variable linkages. In some cases, a good approximation to both the PF and the PS is required by a decision maker. To this end, the RM-MEDA has been extended in [156] to tackle a class of MOOPs in which the dimensionalities of the PF and the PS manifolds are different. RM-MEDAs have been applied to static MOOPs [155], [156], dynamic [202] MOOPs, MOOPs with local PFs [203], MOOPs with high search dimensions [204]. Recently, the RM-MEDA has been improved by combing it with some other techniques [205]. A basic idea behind RM-MEDAs is to use statistical and machine-learning techniques to guide the search of EDAs. Dimension-reduction techniques are thus used in RM-MEDAs. Some other ways to use this regularity property are referred to in [206], [207]. The research work on RM-MEDAs is among very few efforts to design MOEAs based on mathematical programming theory.

3.4.5 Simulated Annealing based approaches

Simulated Annealing (SA) is a single-point-based global optimization technique which is inspired by annealing in metallurgy [208]. Due to its simplicity, SA has been incorporated into MO frameworks for dealing with MOOPs.

Like some other MOEAs, MO SAs also need to maintain an archive to store current non-dominated solutions and to use reproduction operators to generate new solutions. The main difference between MOSAs and other MOEAs is on how to update a solution when the offspring individual is dominated by the parent. The SA updating rule is usually used in such case.

In [209], the SA updating rule is used to choose the next individual when an offspring individual is dominated by the parent. A similar method was proposed in [210], in which a domination based energy function is used to calculate the probability to accept a dominated new trial solution. In [211], the domination relationship between an offspring point and its parent as well as archive points is systematically studied. A MOSA with a single point was introduced in [212]. In this approach, each objective is assigned a different cooling schedule, taking into account the prioritization of that objective. The probability to accept a new solution which is worse than the parent is controlled by using SA rules.

3.4.6 Heuristic based approaches

There are also many other heuristics which are originally designed for scalar objective optimization. By incorporating with the Pareto domination and/or population (archive) updating strategies, these heuristics could also be extended to tackle MOOPs. These meta-heuristics include tabu search [117], [213], scatter search [214], and the GRASP approach [215].

3.5 Self-Adaptive Multi-Objective Evolutionary Algorithms

Self-adaptation is based on the following principle; good solutions more likely result from good than from bad strategy variable values. Bound to the objective variables, these good parametrizations have a high probability of being selected and inherited to the following generation. Self-adaptation becomes an implicit evolutionary search for optimal strategy variable values. These values define properties of the evolutionary algorithm, e.g., mutation strengths, or global parameters like selection pressure or population sizes.

Reed *et al.* in 1967 [28], developed parameter adaptation mechanisms for an EA that learnt to play poker. The genome contained strategy parameters determining probabilities for mutation and crossover with other strategies.

Later in 1970, Rosenberg [29] proposed an approach to adapt the probability for applying crossover.

Weinberg and Berkus in 1970 [30] and later Mercer and Sampson in 1978 [32] introduced meta-evolutionary approaches. In meta-evolutionary methods an outer EA controls the parameters of an inner one that optimizes the original problem.

In 1973, Rechenberg [31] introduced the 1/5th rule, an adaptation mechanism for step size control of Evolutionary Strategies (ES). Self-adaptation term was originally introduced by Rechenberg and Schwefel [216] for ES, and later by Fogel [217] for Evolutionary Programming (EP).

Grefenstette in 1986 [33], suggested a two level adaptive system for tuning the primary optimisation algorithm's parameters. Parameters available for optimisation in Genetic Algorithms (GAs) were: (i) Population Size (N), (ii) Crossover Rate (CR), (iii) Mutation Rate (MR), (iv) Generation Gap (G), and (v) Scaling Window (SW). Also the following Performance Metrics were used: (a) **Online**

Performance: Average performance of all tested structures over the course of the search. (b) **Offline Performance:** Average of best performance achieved in a time interval.

Eiben *et al.* [34] proposed the following taxonomy regarding parameter setting for EAs: (i) Tuning: (a) By Hand, (b) Design of Experiments, and (c) Meta-Evolution; (ii) Control: (a) Deterministic, (b) Adaptive, and (c) Self-Adaptive (stochastic on-line parameter free).

Spears and Jong in 1991 [35] proposed to use a uniform crossover operator as an adaptive operator technique.

Saravanan *et al.* in 1995 [36] compared a Gaussian perturbation self-adaptive mechanism with Log normal perturbation self-adaptive mechanism. They concluded that Log normal distribution has an advantage over Gaussian in some objective functions and Gaussian has an advantage over Log normal in other objective functions. In general Log normal yields better convergence results and is more robust.

Batista *et al.* in 2010 [37] proposed a Chaotic differential mutation function as self-adaptive mechanism. The proposed self-adaptive algorithm when compared to NSGA-II is outperforming it in 14 out of 17 tests performed having lower median Inverted Generational Distance (IGD) values.

Jain and Deb in 2013 [38], proposed an improved version of NSGA for MaOOP (NSGA-III). NSGA-III [218] and [219] is a new version of NSGA that is used to approximate problems with three or more objectives, thus MaOOPs where the use of reference points to guide it during the search process is applied [220]. This improved NSGA-III uses an adaptive approach for reallocating the reference points, named A^2-NSGA-III. Comparing A^2-NSGA-III to NSGA-III and Adaptive NSGA-III (A-NSGA-III), showed to outperform them in terms of solutions distribution and hypervolume measurements.

During the same year (2013), Olive *et al.* proposed a Self-Adaptive MOGA that was applied to Multi-Objective Optimization (MOO) of Airfoil [39]. From their experiments they concluded that the proposed Self-Adaptive MOGA outperforms NSGA-II and Multi-Objective Tabu Search (MOTS) in XFoil[3] application. Though, MOTS suggested more feasible solutions.

[3] http://web.mit.edu/drela/Public/web/xfoil/

Shahsavar *et al.* in 2015 [40] proposed three self-adaptive GAs for a triple-objective project scheduling problem. The three Self-Adaptive GA, were namely: (i) A two-stage Multi-Population Genetic Algorithm (MPGA), (ii) A two-phase Subpopulation Genetic Algorithm (TPSPGA), and (iii) A Non-Dominated Ranked Genetic Algorithm (NRGA). The Self-Adaptive operator technique was implemented as proposed by Spears and De Jong in 1991 [35]. The Self-Adaptive parameters technique was implemented as proposed by Grefenstette in 1986 [33]. All proposed algorithms were two stage processes, the first stage was to find the optimal set of operators to be used, and the second stage was where the optimal parameters are identified, using the optimal set of operators. The project scheduling problem involved three basic objectives (Resource-Constrained Project Scheduling Problem (RCPSP), Resource Investment Problem (RIP), and Resource Levelling Problem (RLP)), in the presence of scarce time and scarce resources was first modelled into a triple objective zero-one formulation. The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) procedure [221], [222] and [223] was hired to order the algorithms. According to TOPSIS, the Self-Adaptive NRGA, Self-Adaptive TPSPGA, and Self-Adaptive MPGA were preference ranked from 1 to 3, respectively (with 1 being the best).

3.6 Concluding Remarks

The majority of real life problems demand MOO solutions that could be derived efficiently and effectively using MOEAs. MOEAs enable the searching of multiple space regions simultaneously looking for possible solutions satisfying multiple objectives.

Over the years numerous MOO algorithms have been proposed, developed and tested in various problems. They cover different algorithmic frameworks from Pareto based, to Decomposition based, to Preference based, to Indicator based, to Co-Evolution based and finally Memetic based approaches. The algorithms are then further diversified by their evolutionary process strategy used. This diversification is what enables us to define MOEA approaches that work well on very specific problems, where other approaches are not that effective.

MOEAs fine tuning is based on experience and multiple runs for defining their numerous search parameters and evolutionary operators. The use of self-adaptive techniques comes to aid or automate in the decision of selecting their search parameters and evolutionary operators.

Chapter 4

LiSIs: Life Sciences Informatics platform

The work on **Li**fe **S**ciences **I**nformatic**s** (LiSIs) platform, described below, has been partially supported through the EU-FP7 GRANATUM project, *"A Social Collaborative Working Space Semantically Interlinking Biomedical Researchers, Knowledge and data for the design and execution of In Silico Models and Experiments in Cancer Chemoprevention"*, contract number 270139.

My contributions in the project were: (a) development of all the chem[o]informatics tools, shown in *red rectangles* in Figure 7, (b) supported the development of tools for preparing and performing docking experiments and tools for preparing and performing property prediction via the use of machine learning algorithms, shown in the *blue rectangles* in Figure 7, (c) management of tools integration to LiSIs and integration of LiSIs with the other platforms of the GRANATUM project, and (d) represented the LiSIs development team and the consortium in project meetings and conferences.

LiSIs is accessible at [1] where you have to create an account to gain access. Once you register an account you can start using LiSIs, though it is advisable to check our online help pages at [2] .

4.1 Introduction

LiSIs is a Virtual Screening (VS) platform based on Scientific Workflow (SW) modelling for Life Sciences. LiSIs aims to provide a set of tools to create, update, store and share SWs for the discovery of active compounds for biomedical researchers. The system is available via a web interface

[1] `http://lisis.cs.ucy.ac.cy`

[2] `http://lisis.cs.ucy.ac.cy/u/user-info/p/online-help-pages-index`

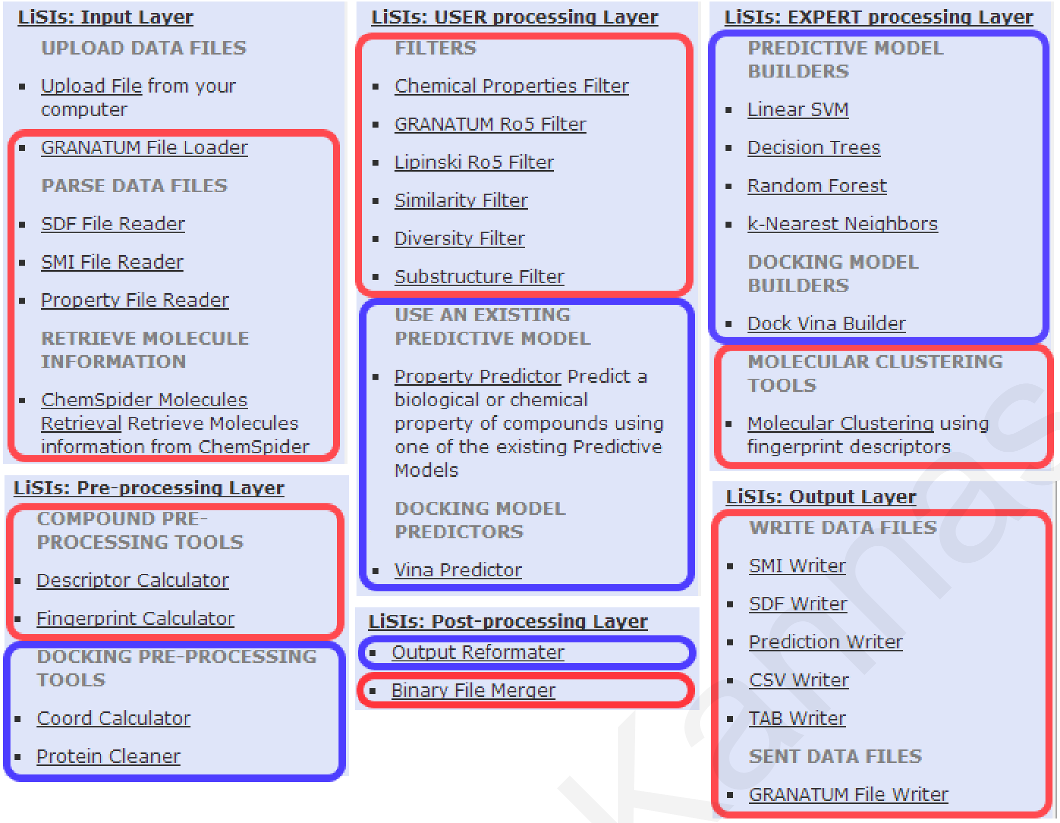
through a password protected, tiered login process. Specifically, the login process provides different level access to platform functionalities based on the user profile. The user is able to assemble SWs utilizing available *in silico* models and tools loaded into the platform. Depending on the user profile and associated permissions, users may also construct new models and tools through the development of custom workflows made available by the system for this purpose. Workflows execute on the system server. The execution results can also be stored on the user's GRANATUM[3] workspace, where the user is able to access, manipulate or share them with other users.

The LiSIs platform is based on the Galaxy web based Scientific Workflow Management System (SWMS) [64], [63], [62] and is comprised of five (5) major layers of functionalities, i.e. Input, Pre-Processing, Processing, Post-Processing and Output, shown in Figure 7. Each layer hosts a collection of components categories essentially implementing a variety of functionalities. A component category may implement different variations of the same functionality. In addition there are numerous tools that are available in the original Galaxy distribution.

The following tools were developed by Kannas C., shown in red rectangles in Figure 7:

- Input Layer:
  - GRANATUM File Loader,
  - SDF File Reader,
  - SMI File Reader,
  - Property File Reader,
  - ChemSpider Molecules,
- Pre-Processing Layer:
  - Descriptor Calculator,
  - Fingerprint Calculator,
- User Processing Layer:

[3] `www.granatum.org`



LiSIs: Input Layer

UPLOAD DATA FILES

- Upload File from your computer
- GRANATUM File Loader

PARSE DATA FILES

- SDF File Reader
- SMI File Reader
- Property File Reader

RETRIEVE MOLECULE INFORMATION

- ChemSpider Molecules Retrieval Retrieve Molecules information from ChemSpider

LiSIs: Pre-processing Layer

COMPOUND PRE-PROCESSING TOOLS

- Descriptor Calculator
- Fingerprint Calculator

DOCKING PRE-PROCESSING TOOLS

- Coord Calculator
- Protein Cleaner

LiSIs: USER processing Layer

FILTERS

- Chemical Properties Filter
- GRANATUM Ro5 Filter
- Lipinski Ro5 Filter
- Similarity Filter
- Diversity Filter
- Substructure Filter

USE AN EXISTING PREDICTIVE MODEL

- Property Predictor Predict a biological or chemical property of compounds using one of the existing Predictive Models

DOCKING MODEL PREDICTORS

- Vina Predictor

LiSIs: Post-processing Layer

- Output Reformater
- Binary File Merger

LiSIs: EXPERT processing Layer

PREDICTIVE MODEL BUILDERS

- Linear SVM
- Decision Trees
- Random Forest
- k-Nearest Neighbors

DOCKING MODEL BUILDERS

- Dock Vina Builder

MOLECULAR CLUSTERING TOOLS

- Molecular Clustering using fingerprint descriptors

LiSIs: Output Layer

WRITE DATA FILES

- SMI Writer
- SDF Writer
- Prediction Writer
- CSV Writer
- TAB Writer

SENT DATA FILES

- GRANATUM File Writer

Figure 7: Overview of **Li**fe **S**ciences **I**nformatics (LiSIs) tools. Tools highlighted in *red rectangle* show tools that were developed by Kannas C. Tools highlighted in *blue rectangle* show tools where their development was supported by Kannas C.

  - Chemical Properties Filter,
  - GRANATUM Ro5 Filter,
  - Lipinski Ro5 Filter,
  - Similarity Filter,
  - Diversity Filter,
  - Substructure Filter,

- Expert Processing Layer:
  - Molecular Clustering,
- Post-Processing Layer:
  - Binary File Merger,
- Output Layer:

  - SMI Writer,
  - SDF Writer,
  - Prediction Writer,
  - CSV Writer,
  - TAB Writer,
  - GRANATUM File Writer.

Additionally Kannas C. helped in the following tools, shown in blue rectangles in Figure 7:

- Pre-Processing Layer:
  - Coord Calculator,
  - Protein Cleaner,
- User Processing Layer:
  - Property Predictor,
  - Vina Predictor,
- Expert Processing Layer:
  - Linear SVM,
  - Decision Trees,
  - Random Forest,
  - k-Nearest Neighbors
- Post-Processing Layer:
  - Output Reformater.

4.2 Input Layer

The Input Layer consists of the following two component categories:

Data File Input: provides tools which support parsing different chemical and biological data files. File formats currently supported include Chemical Data Files, which are sdf (Structure Data File (SDF)), smi (Simplified Molecular Input Line Entry Specification (SMILES)), pdb (Protein Data Bank), pdbqt (AutoDock Protein and Ligand data files) and Biological Data Files which are csv (Comma Separated Values), tab (Tab Separated Values).

GRANATUM File Input: A component which provides GRANATUM's platform users to upload on LiSIs files located at GRANATUM platform.

4.3 Pre-Processing Layer

The Pre-Processing Layer consists of the following four component categories:

Descriptors Calculation: This component category provides tools for the calculation of various descriptors of chemical compounds. Currently the platform enables the calculation of whole-compound descriptors with the use of RDKit [224]. Example descriptors include Molecular Weight (MW), number of Hydrogen Bond Acceptors (HBA) and Hydrogen Bond Donors (HBD), Topological Surface Polar Area (TPSA), number of rings, calculated Octanol-Water partition coefficient (cLogP), molecular complexity based on the method proposed by Barone [225] and molecular flexibility, as well as molecular fingerprints which can be one of Morgan (circular) fingerprints [226], MACCS [227], atom-pair [228], topological torsion [229], and topological fingerprints, a Daylight like fingerprint based on hashing molecular sub-graphs[4] .

Compound Fragmentation: This component category provides tools to identify chemical substructures present in compounds through the *in silico* fragmentation of chemical compound structure. Various compound fragmentation methods are available: Retro-synthetic Combinatorial Analysis Procedure (RECAP) [230], Ring System Decomposition (RSD) and Molecular Frameworks [231].

Docking Preparation: This component category provides the following tools:

[4] `www.daylight.com/dayhtml/doc/theory/theory.finger.html`

- *3D Coordinate Calculator:* A tool for preparing compounds for docking experiments, by calculating their 3D coordinates and creating the appropriate files required by the docking software used by LiSIs.

- *Protein Cleaner:* A tool provided by AutoDock, which is used to automate the process of cleaning a protein to create the required files used by AutoDock Vina[5] .

4.4 Processing Layer

The Processing Layer consists of the following five component categories:

Attribute Filtering: This component category provides tools for implementing filters for selecting compounds based on their chemical and biological attributes. Specifically, these components allow users to enter ranges of acceptable values on available compound properties (including properties calculated by the Chemical Descriptors component and properties provided externally from the Data Input Layer).

Compound Similarity: This component category provides tools for implementing filters for selecting compounds based on chemical structure similarity to other compounds indicated by the user.

Substructure Matching: This component category provides tools for implementing filters for selecting compounds based on whether they contain (or not) the chemical substructure(s) indicated by the user.

Docking Prediction: This component category provides tools for implementing filters for selecting compounds based on predicted binding affinity of a compound to a target protein using *in silico* docking prediction. Our platform currently uses AutoDock Vina, a popular docking application, freely available to the academic research community. AutoDock Vina attempts to find the best receptor-ligand docking pose by employing a scoring function that takes into consideration both intra-molecular and intermolecular contributions, as well as an optimization algorithm [232].

Predictive Modelling: The primary aim of this component is to provide the user with the tools to construct data-driven predictive models based on available information on a set of compounds. These

[5] `vina.scripps.edu`

models are used to predict biochemical properties of interest of new compounds and to select those with an acceptable profile.

The component currently makes use of four popular predictive modelling algorithms widely used by the chem[o]informatics community: Decision Trees (DT), Random Forests (RF), Support Vector Machines (SVM), and k-Nearest Neighbours (kNN) [233].

4.5 Post-Processing Layer

The Post-Processing Layer consists of the following component category:

Binary File Merging: This component category provides tools for merging binary files, containing chemical structure objects with processing component results, into one binary file.

4.6 Output Layer

The Output Layer consists of the following three component categories:

Reporting: This component category provides tools for the formatting of the processing results from one or more *in silico* experiments and for basic visualization.

Storage: This component category provides tools to store results in various formats for future reuse and sharing.

Output Reformatting: This component category provides tools to convert results in various formats supported by OpenBabel [234].

4.7 Third Party Tools used by LiSIs

The LiSIs platform uses several, freely available to the research community tools to expedite development and maximize resources. Specifically, the following 3rd party tools are used:

Galaxy [64], [63], [62], an open, web-based platform for data intensive biomedical research, used for the customized SWMS platform;

RDKit [224], an open source chem[o]informatics toolkit, used to support all the chem[o]informatics related functionalities;

Pybel [235], a Python wrapper for the OpenBabel chem[o]informatics toolkit, used for chemical file format transformations;

R [236], a statistical environment used to support data mining, machine learning and statistics related functionalities; caret (Classification and Regression Training) package[233] is used for the generation of Predictive Models;

AutoDock Vina [232] docking application used to support docking experiments functionalities.

4.8 Showcase and Results

LiSIs has been used for the implementation of a VS experiment in order to identify molecules able to bind to Estrogen Receptor-α (ER-α) and/or Estrogen Receptor-β (ER-β).

This experiment was designed to combine the experience of our cancer chemopreventive biomedical experts of the two biomedical research groups. The Cancer Biology and Chemoprevention Laboratory [6] provided the experience with ER-α and ER-β and the Cancer Chemoprevention and Epigenomics group [7] provided the experience with DNA Methyltransferase (DNMT). As such the targets for the experiment were ER-α and ER-β and the compounds that were to be investigated were a collection from Indofine [8] , as the Cancer Biology and Chemoprevention Laboratory were using them extensively and the compounds from Medina-Franco *et al.* [237] research were investigated by the Cancer Chemoprevention and Epigenomics group.

Figure 8 illustrates the complete workflow, in an abstract layer, used by LiSIs for the showcase described. Figure 9 shows the workflow that was designed and executed on LiSIs. At the Input Layer, parsing of the input datasets takes place. To start with the initial datasets in SMILES format include 2414 compounds from Indofine, 55 compounds characterized as DNMT inhibitors by Medina-Franco *et al.* [237] and 21 known ER ligands retrieved from PubChem[9] , shown in Table 2, which were used as a positive control dataset for the validation of docking tools. Tools were used to read chemical input files and create compound object structures for further processing by the Pre-Processing and

[6] https://www.ucy.ac.cy/biol/en/research/20-en-topm/50-andreasioannoucostantinou

[7] https://www.dkfz.de/en/tox/cancer_chemoprevention.html

[8] www.indofinechemical.com

[9] pubchem.ncbi.nlm.nih.gov

Processing Layers. The total number of unique compounds pushed to the next layer were 2413 from Indofine (one was found to contain erroneous molecular information), 54 from Medina-Franco (two were found to be identical) and 21 from PubChem's ER agonists and antagonists (one was found to contain two disconnected fragments), datasets for a total of 2488 compounds.

Table 2: Known ER ligands used as positive controls for the validation of the *in silico* results

A/A	Estrogen Ligand	Docking Score ER-α	Docking Score ER-β
1	Raloxifene	-11.70	-8.72
2	Lilly-117018	-11.53	-3.80
3	3-HydroxyTamoxifen	-11.02	N/A
4	Nafoxidine	-10.88	N/A
5	ICI-182780	-10.73	N/A
6	Pyrolidine	-10.04	N/A
7	Clomiphene A	-10.01	N/A
8	Nitrofinene Citrate	-9.87	N/A
9	ICI-164384	-9.82	-9.13
10	Moxestrol	-9.38	-9.77
11	Naringenine	-8.55	-7.80
12	Triphenylethylene	-8.50	N/A
13	Afema	-8.15	-7.78
14	Danazol	-6.99	N/A
15	Ethamoxytriphetol	-6.67	N/A
16	4-HydroxyTamoxifen	-6.60	N/A
17	Dioxin	-6.22	N/A
18	Estralutin	-5.86	-3.80
19	Cyclopentanone	-4.88	N/A
20	Miproxifene Phosphate	-4.48	N/A
21	EM-800	N/A	N/A

Note: The list was retrieved from PubChem and it includes compounds characterized as estrogen ligands. N/A; no binding affinity.

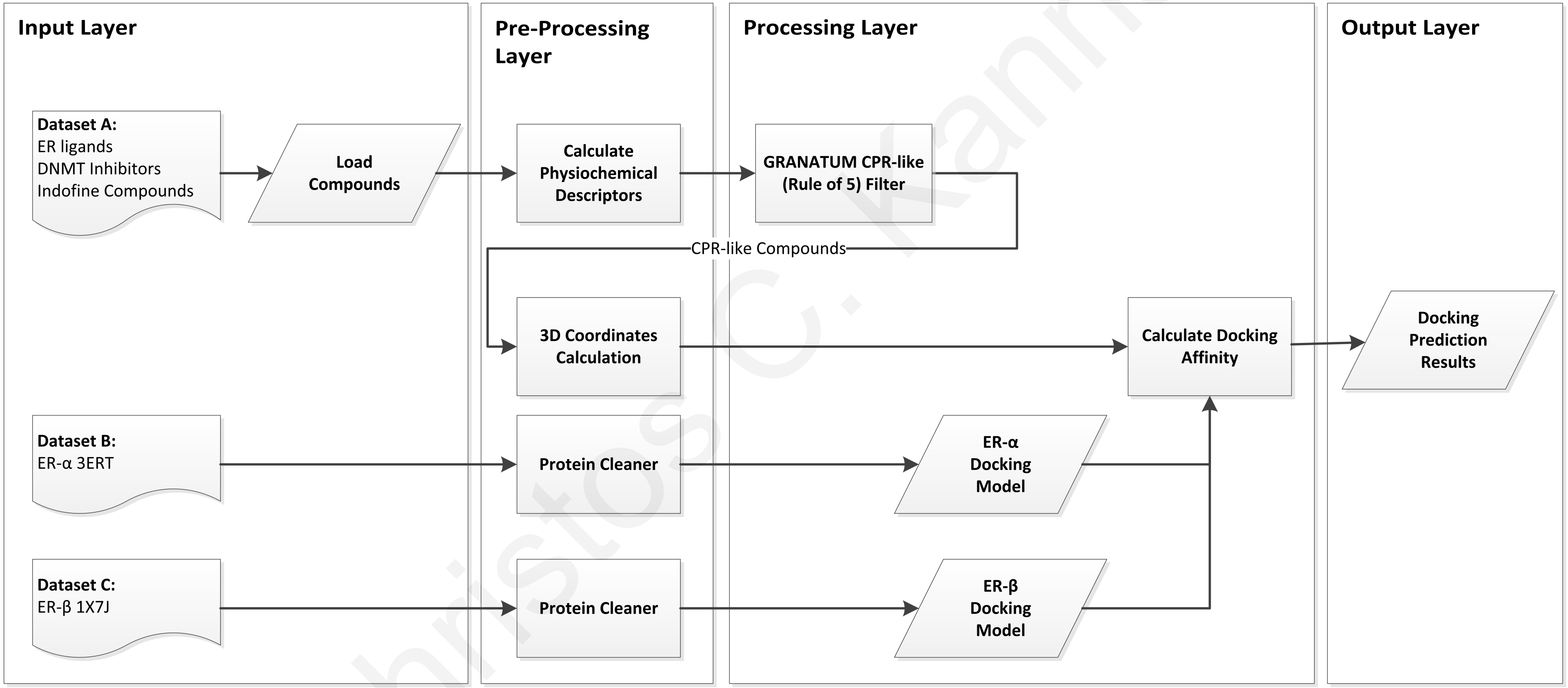
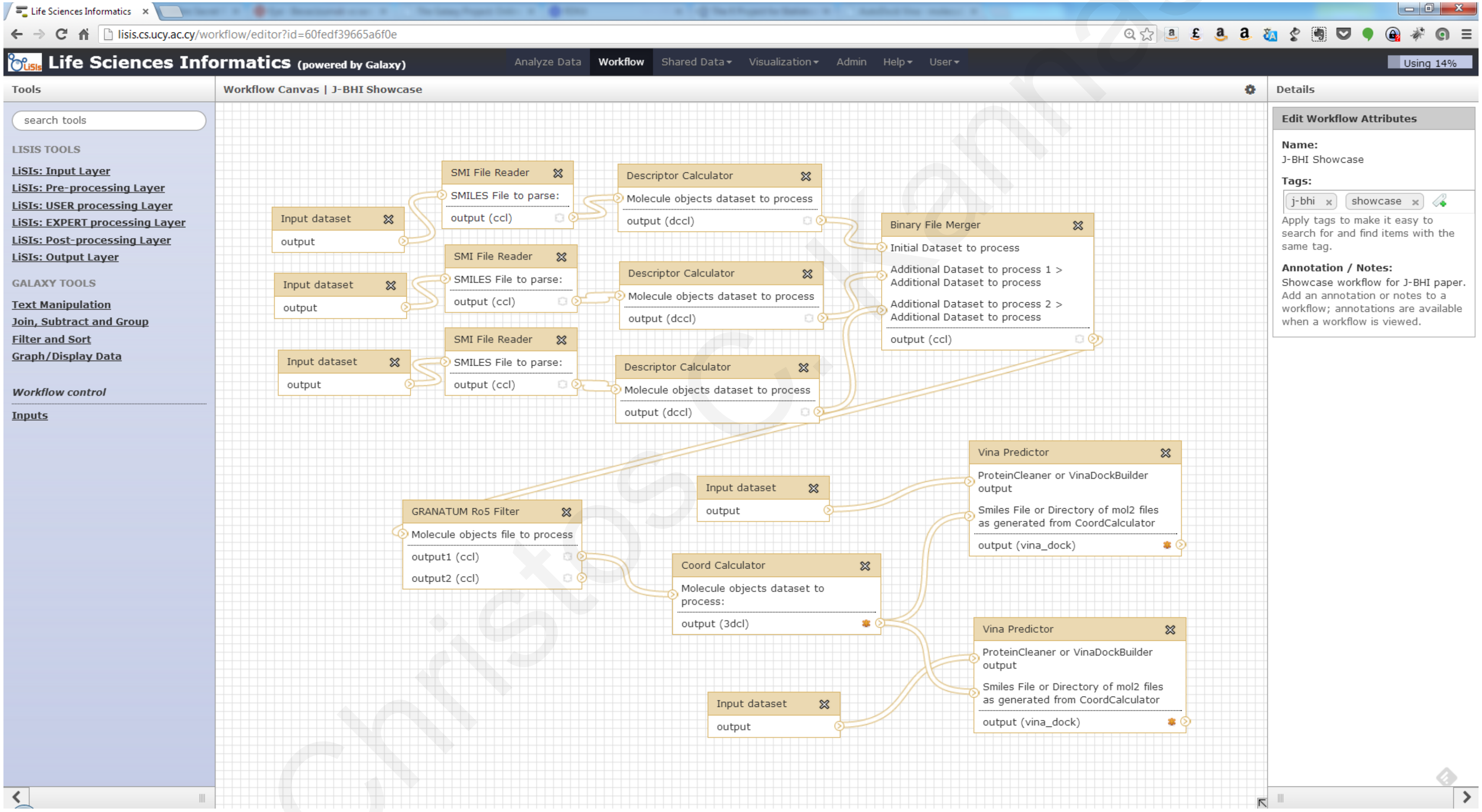


Figure 8: Schematic of the workflow for current showcase, provided by Kannas *et al.* [238].


Life Sciences Informatics
lisis.cs.ucy.ac.cy/workflow/editor?id=60fedf39665a6f0e
Life Sciences Informatics (powered by Galaxy)
Analyze Data
Workflow
Shared Data
Visualization
Admin
Help
User
Using 14%
Tools
search tools
LISIS TOOLS
LiSIs: Input Layer
LiSIs: Pre-processing Layer
LiSIs: USER processing Layer
LiSIs: EXPERT processing Layer
LiSIs: Post-processing Layer
LiSIs: Output Layer
GALAXY TOOLS
Text Manipulation
Join, Subtract and Group
Filter and Sort
Graph/Display Data
Workflow control
Inputs
Workflow Canvas | J-BHI Showcase
Input dataset
output
SMI File Reader
SMILES File to parse:
output (ccl)
Descriptor Calculator
Molecule objects dataset to process
output (dccl)
Binary File Merger
Initial Dataset to process
Additional Dataset to process 1 > Additional Dataset to process
Additional Dataset to process 2 > Additional Dataset to process
output (ccl)
GRANATUM Ro5 Filter
Molecule objects file to process
output1 (ccl)
output2 (ccl)
Coord Calculator
Molecule objects dataset to process:
output (3dcl)
Vina Predictor
ProteinCleaner or VinaDockBuilder output
Smiles File or Directory of mol2 files as generated from CoordCalculator
output (vina_dock)
Details
Edit Workflow Attributes

Name:
J-BHI Showcase

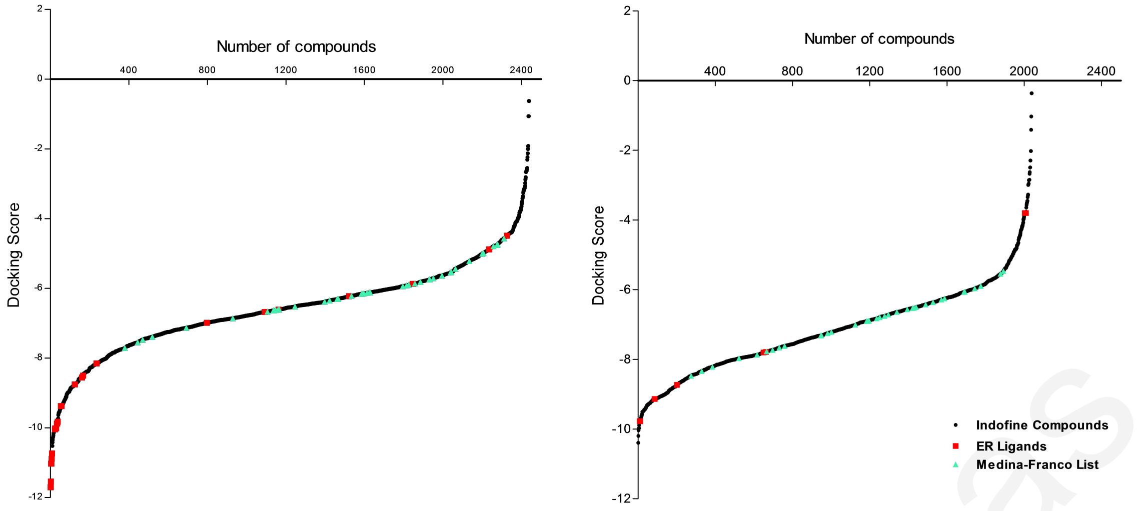
Tags:
j-bhi × showcase ×

Apply tags to make it easy to search for and find items with the same tag.

Annotation / Notes:
Showcase workflow for J-BHI paper.

Add an annotation or notes to a workflow; annotations are available when a workflow is viewed.

Figure 9: **Li**fe **S**ciences **I**nformatics (LiSIs) workflow for current showcase, provided by Kannas *et al.* [238].



(a) Estrogen Receptor-α Docking Score

(b) Estrogen Receptor-β Docking Score

Figure 10: Compounds were tested against Estrogen Receptor-α (a) and Estrogen Receptor-β (b) using *in-silico* docking tools, provided by Kannas *et al.* [238]. Docking score for our library of compounds was calculated against the crystal structures 3ERT (Estrogen Receptor-α) and 1X7J, (Estrogen Receptor-β) shown. The red dots represent known ER ligands as listed in Table 2 and the cyan dots represent DNA Methyltransferase inhibitors characterized in [237]. The lower (most negative) the value of the docking score the higher the predicted binding affinity.

At the Pre-Processing Layer (see Figure 9), a set of physiochemical molecular descriptors were calculated including Molecular Weight, Hydrogen Bond Donors, Hydrogen Bond Acceptors, Topological Surface Polar Area and calculated Octanol-Water partition coefficient.

At the Processing Layer, the following tools were used:

1. *GRANATUM Rule of Five (Ro5) filter (see Figure 9 Processing Layer):*

   (a) Molecular Weight (MW) between 160 and 700,

   (b) Hydrogen Bond Donors (HBD) less or equal to 5,

   (c) Hydrogen Bond Acceptors (HBA) less or equal to 10,

   (d) Topological Surface Polar Area (TPSA) less than 140, and

   (e) calculated Octanol-Water partition coefficient (cLogP) between -0.4 and 5.6.

   This filter was defined by Chemoprevention Research (CPR) experts participating to the GRANATUM project.

The filtering resulted in 1834 compounds with CPR-like features and 654 compounds without CPR-like features. The compounds with CPR-like features were pushed for docking experiments.

2. *Docking experiment against ER-α and ER-β (see Figure 9 Processing Layer):* LiSIs uses AutoDock Vina [232] and has been setup to provide us with the maximum docking affinity score. The current key aim of the GRANATUM project was to identify ER-α antagonists and ER-β agonists. Docking experiments on the filtered combined dataset have been performed by employing receptors ER-α 3ERT and ER-β 1X7J. The appropriate Docking Models were created using protein structures obtained from the PDB database[10] and related LiSIs tools for automated Protein Cleaning (see Figure 9 Pre-Processing Layer) and Docking Model Preparation.

   Figure 10a is a graphical representation of the docking affinity score predicted by LiSIs docking experiment tool for ER-α, and Figure 10b is a graphical representation of the docking affinity score predicted by LiSIs docking experiment tool for ER-β. The predicted binding affinity scores of the known ER inhibitors, depicted with red colour in Figure 10a, 10b, indicate the validity of the docking models prepared and the ability of these models to assign a lower score to inhibitors and reproduce ground truth. Consequently, the models are applicable in a VS context, i.e. for the prioritization of unknown compounds based on their predicted binding affinity to estrogen receptors.

Finally a selection of molecules highly ranked was hand-picked; a small sample of those is shown in Table 3. These molecules have undergone *in vitro* investigation to provide feedback for the calibration of the tools available on LiSIs platform and also to select a small set for further research.

As shown in Table 3, three novel flavones, 3',4'-dihydroxy-a-naphthoflavone (Compound 2), 3,5,7,3',4'-pentahydroxyflavanone (Compound 5), and 4'-hydroxy-a-naphthoflavone (Compound 6) were among those with high binding scores for ER-α and ER-β as indicated from the *in silico* docking score. Flavones, a class of flavonoids, have previously been demonstrated to possess estrogenic

[10] www.rcsb.org

activity in a number of hormonally responsive systems. Their estrogenic and antiestrogenic activities appear to correlate directly with their capacity to displace Estradiol from ER [239]. Our *in vitro* results showed that Compound 2 had the highest affinity for both receptors while Compound 5 also displayed similar affinity for both ER-α and ER-β. However Compound 6 was found to bind only weakly to ER according to the binding affinity assay. Furthermore, results from the *in silico* experiments showed that three previously uninvestigated coumarins, 3(2'-chlorophenyl)-7-hydroxy-4-phenylcoumarin (Compound 3), 3(3'-chlorophenyl)-7-hydroxy-4- phenylcoumarin (Compound 4) and 4-benzyl-7-hydroxy-3-phenylcoumarin (Compound 7) can potentially bind ER-α and ER-β based in their docking scores. Coumarins are natural or synthetic benzopyranic derivatives that form a family of active compounds with a wide range of pharmacological properties, including estrogen-like effects [240]. *In vitro* results showed that Compound 3 has greater affinity for ER-α while Compound 4 can bind with high affinity to both receptors. However, Compound 7 was not able to bind to either receptor as determined by the ER binding affinity assay.

4.9 Discussion

In recent years, many high-throughput methods have been established in the effort to identify novel Estrogen Receptor binders with anticancer activity. However, *in vitro* assays often produce disappointing results due to the small percentage of novel active Estrogenic compounds discovered. To identify novel compounds that act as effective ER-α coactivator binding inhibitors (CBIs), Gunther *et al.* applied a time-resolved fluorescence resonance energy transfer (TR-FRET) assay developed in a 384 well format [241]. This assay measures the binding of a Cy5-labeled SRC-1 nuclear receptor interaction domain to the ligand binding domain (LBD) of labeled ER-α leading to FRET signal generation. Compounds that interfere with the FRET signal are identified as potential coactivator binding inhibitors (CBIs) or conventional ligand antagonists. Based on this method, only 1.6% of the total compounds screened were identified as active as reported in (Pubchem ID 629[11]).

[11] http://pubchem.ncbi.nlm.nih.gov/assay/assay.cgi?aid=629

In the present study, we used a VS Workflow implemented using the LiSIs platform to screen the Indofine database of 2413 compounds. Based on their drug-like criteria and docking results we selected 18 potential ER ligands. These were further investigated *in vitro* with the ER binding assay described by Gurer-Orhan *et al.* [242] with minor modifications. In this manner it was found that five agents displayed strong affinity for ER-α, three showed selectivity for ER-β and seven were able to bind to both receptors with similar affinity. In total 15 out of 18 compounds (83.3%) were experimentally confirmed active. Therefore, the use of LiSIs platform may allow researchers to execute complex biomedical studies and *in silico* experiments on largely available and high quality data repositories in order to facilitate the selection and prioritize the investigation of novel chemopreventive compounds *in vitro*.

Compounds with high binding affinity to the ERs based on the *in silico* results, display structural characteristics that are similar to Estradiol-17β (E2). All contain a phenolic ring which is indispensable for binding to the estrogen receptor [243]. The phenolic ring of Compounds 2 - 7 contains at least one hydroxyl group which mimics the 3'-OH of E2. Furthermore, all compounds have low molecular weight comparable to that of E2 (MW equal to 272). All agents are highly hydrophobic which is required for binding in the ER binding pocket [244]. The differences observed in the binding affinities of compounds may be attributed to differences in structural characteristics. The lower ER binding affinity of Compound 5 (when compared to Compound 2) may be attributed to the hydrophilic hydroxyl group at C-11 of Compound 5 which, due to steric hindrance, lowers its binding affinity for both receptors [244].

4.10 LiSIs Evaluation

LiSIs was evaluated by 7 biomedical researchers (end-users). LiSIs evaluation revealed that 43% of the people who participated found it quite easy to design and execute a VS workflow, 43% found it easy but they had some problems in doing so while 14% had some troubles in doing this. Regarding the difficulty of uploading a new dataset in LiSIs platform, all of the people who evaluated LiSIs found it very easy, in detail 29% had no problems at all and 71% had minor problems. When questioned if they were able to gather results from their VS workflows, 43% answered that they had some problems, 14% answered that they had minor problems and 43% had no problems at all. To address this issue LiSIs was updated, introducing a *"How To"* section for describing the usage and functionality of each tool[12] . Additionally, various workflows and histories have now been created and uploaded on the platform to guide users for using LiSIs.

The evaluation regarding the tools to create and use Docking and Property Prediction Models showed that the ability to perform Property Prediction using existing Prediction Models from the LiSIs platform was as follows: 29% were able to perform the prediction with some difficulties, 57% were able to do so with minor difficulties and 14% were able to perform the prediction without any difficulties. Regarding the ability to perform Docking Prediction using existing Docking Models from the LiSIs platform, 14% were able to perform the prediction but with difficulties, mostly due to the known limitations of the automated process that is being used, 57% were able to do so with minor difficulties and 29% were able to perform the prediction without any difficulties. Based on the user feedback, to facilitate the end-users, LiSIs was updated, introducing a *"How To"* section for describing the usage and functionality of each tool[13] . When questioned about the usefulness and the difficulty of creating and saving a Property or a Docking Prediction Model, 29% answered that they had some difficulties, 57% answered that they had minor difficulties and 14% answered that they were able to do so with ease.

The evaluation regarding the re-usability of useful workflows revealed that 17% of the users found the functionality of saving a workflow for a future use a task with some difficulties, 67% replied that task is performed with minor difficulties and 17% were able to perform the task without any difficulties. When participants were asked about the usefulness of the platform 57% replied that it is very useful and 43% replied that is extremely useful.

The evaluation revealed that the objectives of the LiSIs platform have been fulfilled by providing the users an easier means for performing VS workflows and facilitating their experiments. The web-based approach for the integration of the tools and the use of standards turned out to be the right

[12] http://lisis.cs.ucy.ac.cy/u/user-info/p/online-help-pages-index

[13] http://lisis.cs.ucy.ac.cy/u/user-info/p/online-help-pages-index

decision; it guarantees the platform independent access to the applications and resources. The evaluation survey showed that participants were able to interact with LiSIs/GRANATUM using different established Web browsers. The variety of tools fulfilled the objective of creating and performing VS workflows for identification of interesting molecules with chemopreventive properties.

Through the development process and the user evaluation, several lessons learnt were identified:

1. To make it more transparent the software should provide a tool for depiction of results, to underline the quality of results, and a threshold manager where scientists can work more on an expert level.

2. Researchers need a way to see the progress of each active tool. This will be considered for future research.

3. Visualizing the results in various stages can be added to the current arsenal of LiSIs. There are ways to implement this in the near future by using additional 3rd party software.

4. Providing tools targeted to expert users for manual modification of protein structures prior to docking experiments is also a much requested feature. This requires the implementation of a fully interactive visualization tool that will show the protein structure and provide the end user with the option of modifying its structure.

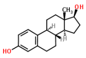
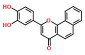
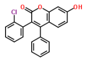
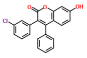
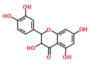
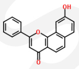
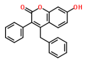
5. LiSIs administration requires the user group based access of tools. This is an upcoming feature of Galaxy.

6. The complete tool set of LiSIs should be made available in a Tool Shed. Tool Shed is a repository like platform of Galaxy that enables the hosting of tools, assigning 3rd party dependencies, data types configurations, tools dependencies and an easy installation across Galaxy powered platforms.

4.11 Conclusion

The LiSIs platform aims to fill the current void in the application of advanced chem[o]informatics and computational chemistry technology in determining efficacy and predicting possible mechanism of action or identifying a possible receptor for a chemopreventive agent in life sciences research. Its successful deployment may have a substantial impact on enabling biomedical researchers to utilize state of the art computational techniques to search for promising chemical compounds that may lead to the discovery of novel agents with chemopreventive properties. We have shown that by utilizing the LiSIs platform in conjunction with a widely used docking program we identified compounds that can bind to ER-α and/or ER-β with a high degree of success rate. This *in silico* approach is expected to facilitate the process of identification of lead compounds with estrogenic or anti-estrogenic activity and to enhance considerably the discovery process for new therapeutic agents.

Table 3: Selection of highly ranked compounds from the final virtual screening results

A/A	Chemical Structure	Molecular Weight (g/mol)	Concentration (μM)	ER-α LBD		ER-β LBD	
				Binding Affinity	Docking Score	Binding Affinity	Docking Score
1	 17β-Esrtradiol	272.38	10	1	-9.4	1	-10
2	 3',4'-dihydroxy-a-naphthoflavone	304.29	1 10	0.11 0.22	-7.59	0.05 0.34	-10.39
3	 3(2'-chlorophenyl)-7-hydroxy-4-phenylcoumarin	348.78	1 10	0.21 2.71	-9.73	N/A 0.34	-10.03
4	 3(3'-chlorophenyl)-7-hydroxy-4-phenylcoumarin	348.78	1 10	0.24 2.23	-10.34	0.13 2.75	-9.67
5	 3,5,7,3',4'-pentahydroxyflavanone	304.26	1 10	N/A 0.27	-8.81	0.06 0.18	-9.61
6	 4'-hydroxy-a-naphthoflavone	228.29	1 10	N/A N/A	-8.18	0.05 N/A	-9.88
7	 4-benzyl-7-hydroxy-3-phenylcoumarin	328.37	1 10	N/A N/A	-10.13	N/A N/A	-9.13

Note: Comparison between the *in silico* docking scores and the *in vitro* binding affinities of selected compounds. The binding affinity was normalized to that of 17-β Estradiol which was set as 1 representing 100% binding. LBD; Ligand Binding Domain. N/A; No (binding) Affinity.

Chapter 5

Multi-Objective Evolutionary Algorithms for Molecular De Novo Design

Recently we proposed an algorithmic framework for the problem of Multi-Objective (MO) Optimal Graph Design (OGD) for labelled, undirected graphs [245]. Solutions to this problem were graphs consisting of genes from two sets, the set of vertices and the set of edges. Multiple types/labels of vertices and edges were allowed and therefore the problem suffers from the combinatorial explosion of the number of potential graph solutions. Additionally, the OGD problem usually has a complex, multi-modal solution space due to the multiple potentially conflicting objectives that need to be satisfied by the solution graphs and, the combinatorial nature of the problem. Consequently, from a computational optimization perspective, the problem corresponds to searching the huge space of valid graphs to discover and select the few designs satisfying, or compromising in the case of conflicts, the objectives imposed. In this context validity of the resulting graphs is problem specific and, as such, the inclusion of problem domain knowledge to the process can facilitate the process. The role of diversity in the population of solutions also assumes increased importance; since multiple solutions, and not only the single best one, are being sought, the process needs to ensure that the population is -to the degree feasible- representative of the range of solutions existing in the various regions of the search space. To solve the problem a search strategy capable of global exploration while paying special attention to the diversity of the population and the ability to converge to individuals in promising localities of the space was implemented.

In Table 4 we list some notable Multi-Objective Evolutionary Algorithms (MOEAs) for Molecular De Novo Design that have been proposed since 2008. The method used for the MO selection is shown in the *MO Method* column, and the algorithmic approach used is shown in *Search Method* column. From these two columns some interesting insights are derived. The first is that the majority of the approaches use a weighted function as their MO selection method in order to overcome the shortcomings of Pareto based selection method for Many-Objective Optimization Problems (MaOOPs). The second is that approaches proposed prior to 2010 are based on Evolutionary Algorithms (EAs) but the modern approaches (from 2010 and onwards) are based on workflow based approaches which gives the opportunity to use specialised software for each required step. The column *Remarks* describes the design methodology used. The *Ligand* term defines ligand based design, the *Structure* term defines structure based design, the *Pharmacophore* term defines pharmacophore based design and the *ADME related properties* term defines a design approach guided by Absorption, Distribution, Metabolism, Excretion (ADME) related properties.

Table 4: Multi-Objective Evolutionary Algorithms (MOEAs) for Molecular De Novo Design (DND)

Name	Year	Multi-Objective Method	Search Method	Remarks	Reference
EA-Inventor	2008	Weighted	Evolutionary Algorithm	Ligand	[246]
GANDI	2008	Weighted	Parallel Evolutionary Algorithm	Structure	[247]
FOG	2009	Weighted	Evolutionary Algorithm	Ligand	[248]
MEGA	2009	Pareto based	Evolutionary Algorithm	Ligand & Structure	[78]
PLD	2010	Pareto based	Evolutionary Algorithm	ADME related properties	[249]
NovoFLAP	2010	Weighted	Evolutionary Algorithm	Ligand	[250]
PhDD	2010	Weighted	Workflow	Pharmacophore	[251]
DOGS	2012	Weighted	Workflow	Ligand	[252]
LiGen	2013	Weighted	Workflow	Ligand, Structure & Pharmacophore	[253]
MOARF	2015	Weighted	Workflow	Ligand & Structure	[254]
Synopsis	2016	Pareto based	Evolutionary Algorithm	Ligand & Structure	[255]

5.1 Multi-Objective Evolutionary Graph Algorithm

Nicolaou *et al.* in [78] and [245] proposed Multi-Objective Evolutionary Graph Algorithm (MEGA), a framework which combines evolutionary techniques with graph data structures to directly manipulate graphs and perform a global search for promising solutions. Additionally, MEGA can incorporate problem-specific knowledge and local search heuristics and techniques, to improve performance and scalability.

MEGA initiates with the supply of a set of molecular building blocks, the implemented objectives to be used for scoring the graphs and a set of attributes controlling mutation and crossover methods and probabilities, selection method, hard filters for solution elimination, etc. Optionally, a set of molecules to be used as the initial population may be supplied as well. The supplied data are used to initiate internal data structures, for example to create graph-based chromosomes representing the molecules and to construct a list of building block objects to use in subsequent steps.

Next the algorithm applies the objectives on the initial population to obtain a list of scores for each individual. The list of scores may be used for the elimination of solutions with values outside the range allowed by the corresponding active hard filters.

In the next step, the list of scores is subjected to a Pareto ranking procedure as described in [82]. According to this procedure the rank of an individual is set to the number of individuals that dominate it incremented by 1, thus non-dominated individuals are assigned rank order 1 (see Figure 11).

At this phase the algorithm proceeds to calculate a Multi-Objective Fitness (MOFit) score for each individual. There are two ways of calculating such a score, controlled by user preferences: (1) The first simply uses a linear transformation function that assigns a higher score to solutions with low Pareto rank. This method operates exclusively on phenotypes, i.e. in solution space. (2) The second method invokes a niching mechanism that performs diversity analysis of the population via clustering of the genotypes, i.e. the chemical structures, and subsequently prepares a two-valued MOFit score that consists of both the linear transformation of the Pareto rank and the cluster assignment of the individual.

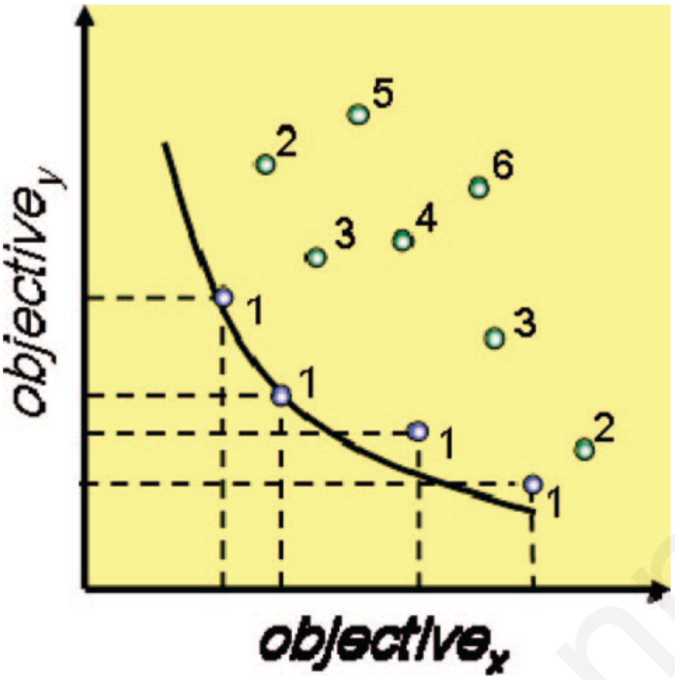


Figure 11: Pareto front dominance, provided by Nicolaou *et al.* [78]. In a Multi-Objective Optimization Problem (MOOP) several equivalent, non-dominated solutions may exist representing compromises among the different objectives. Typically solutions to the problem are ranked according to the number of other solutions dominating them, i.e. solutions that are better in all objectives. Non-dominated solutions are labelled with '1'. The curved line represents the Pareto Front (PF). Note that both objectives in the example shown should be minimized.

An additional optional step at the users' disposal is the application of elitism which creates and maintains an external archive of Pareto optimal solutions found during all previous iterations. If elitism is enabled, then the archive of Pareto solutions is merged with the current population before the MOFit calculation step to form an extended population. Recalculation of the Pareto rank and diversity analysis are performed on the extended set to calculate the MOFit score of the solutions. The non-dominated solutions of the extended population are then stored in the Pareto archive.

Following, MEGA checks for the termination conditions, typically if the number of pre-set maximum allowed iterations has been reached; if satisfied the process terminates. However, if this is not the case, then the process moves to select the parent subset population.

Parent selection is performed using one of the ”best”, ”roulette”, or ”tournament” methods on the MOFit scores of the solutions. The ”best” method simply selects the subset of solutions with the highest transformed Pareto rank score, whereas the ”roulette” method selects solutions via a probabilistic mechanism that assigns higher selection probability to solutions with higher transformed

Pareto rank. The ”tournament” method picks random pairs of solutions and selects the one with the highest transformed Pareto rank score.

If the niching mechanism is not enabled, then the chosen parent selection method is applied once on the entire set of candidate solutions to generate the parent sub-population. If the niching mechanism is enabled, and thus the MOFit scores consist of the transformed Pareto rank and the cluster assignment of the individual, then the selection methods are applied on the clusters rather than the entire population. The process picks one solution from each cluster starting from the most populous cluster and proceeding to clusters containing the fewest compounds. The process traverses the set of clusters until the number of parents is selected.

The parents are then subjected to mutation and crossover according to the probabilities indicated by the user. The new population is formed by merging the original population and the newly produced mutants and crossover children.

The process then iterates, and the new population is subjected to fitness calculation against all objectives, hard filtering and Pareto ranking.

Following, MEGA proceeds to reduce the new population to the user defined population size using a ”roulette” like method. The method is essentially identical to the ”roulette” parent selection method described previously except that it assigns a higher selection probability to the worst performing solutions. Best performing solutions, i.e. non-dominated solutions, have a selection probability of zero. In the special case where the number of best performing solutions exceed the user defined population size an adequate number is randomly selected and marked as ”excess” solutions. If elitism is enabled, then these solutions are treated as normal members of the population in the next steps of the algorithm. However, if elitism is not enabled, then these solutions are removed from the current population prior to the parent selection step. Figure 12 summarizes the MEGA framework process. While MEGA has been designed to search for solutions compromising multiple objectives it can also be used in a Single Objective Optimization Problem (SOOP) mode simply by eliminating the Pareto rank step and replacing the transformed Pareto rank score with the transformed single objective score in all following steps. Correspondingly, the diversity score is then used in the same way as in the standard MOOP case.

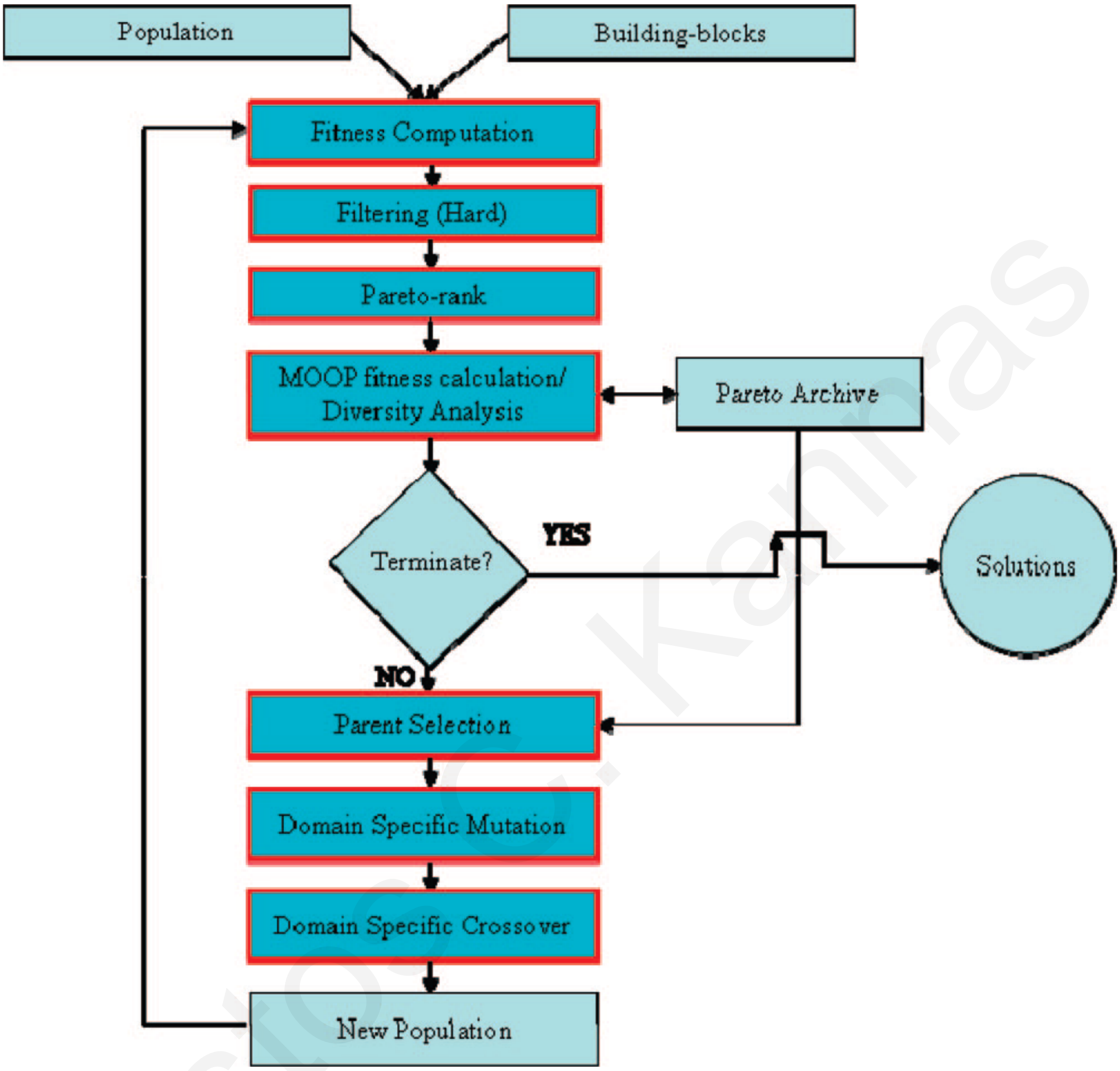


Figure 12: The Multi-Objective Evolutionary Graph Algorithm (MEGA) framework, provided by Nicolaou *et al.* [78]. Note the Pareto archive component storing an elite population of solutions at each generation.

Further details can be found in [256], where Nicolaou and Kannas describe the use of MEGA for the design of a molecular library of novel compounds, based on specific criteria.

An overview of the MOOP methods and tools that are used in DND procedure can be found in Nicolaou *et al.* [257] and [258], and in Nicolaou and Brown [259].

The MEGA algorithm was developed in context of the Ph.D. of Dr. C. Nicolaou [260]. The objective of this thesis was the development of the Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) framework, described in Section 5.3, where MEGA is the inner algorithm. Thus my contribution was directed in various modules of the MEGA framework, such as parts of the algorithm, the scoring functions modules, data preparation modules, supplementary modules that enabled us to create hybrid versions of MEGA, batch processing module that enabled us to run numerous experiments of various MEGA versions and experiments with different settings, etc [1] .

5.2 Parallel Multi-Objective Evolutionary Graph Algorithm

Parallel Multi-Objective Evolutionary Graph Algorithm (PMEGA) is our parallel version of MEGA that facilitates the co-evolution framework to evolve multiple sub-populations in parallel, which has been implemented using the multiprocessing framework of Python 2.6 - 2.7 [261]. The aim of the research presented here was to exploit the potential benefits presented by utilising multi-core CPUs in MOEAs [262]. An additional goal was to investigate the effectiveness of parallel MOEAs in the problem of molecular DND.

In a Parallel Evolutionary Algorithm (PEA) model the entire population available needs to be in a distributed or shared form. In coarse-grained or distributed PEAs, there exist multiple independent or interacting sub-populations, while in fine-grained PEAs there is only one population where each population member can be processed in parallel. In a coarse-grained PEA, the populations are divided into several sub-populations. These sub-populations evolve independently of each other for a certain number of generations (isolation time). Upon completion of the isolation time a number of the resulting individuals is distributed between the sub-populations, a process often referred to as migration. The number of exchanged individuals (migration rate), the selection method of the individuals for migration and the scheme of migration determines how much genetic diversity can occur in the

[1] In a nutshell as my experience was growing I was involved even more in the whole framework of MEGA and its supporting modules. Since 2009 I maintain a private source code repository (hosted on BitBucket) for the whole package of Noesis Cheminformatics Ltd. Suite, which MEGA framework is part of it. Similarly I'm the maintainer of the source code repository for **Li**fe **S**ciences **I**nformatic**s** (LiSIs) and I act as the administrator of our LiSIs server, which is live since 2012.

sub-population as well as the exchange of information between sub-populations. The selection of the individuals for migration typically takes place using one of the following two methods:

- Uniformly at random (i.e. pick individuals for migration in a random manner),

- Fitness-based (i.e. select the best individuals for migration).

Additionally, several possibilities exist for the migration scheme of individuals among sub-populations. Common migration schemes include:

- **Complete topology**, unrestricted net topology which exchanges individuals among all sub-populations (Figure 13),

- **Ring topology**, where exchange of individuals is allowed only to a specific sub-population (Figure 14), and

- **Neighbourhood topology**, where individuals are exchanged across a neighbourhood (Figure 15).

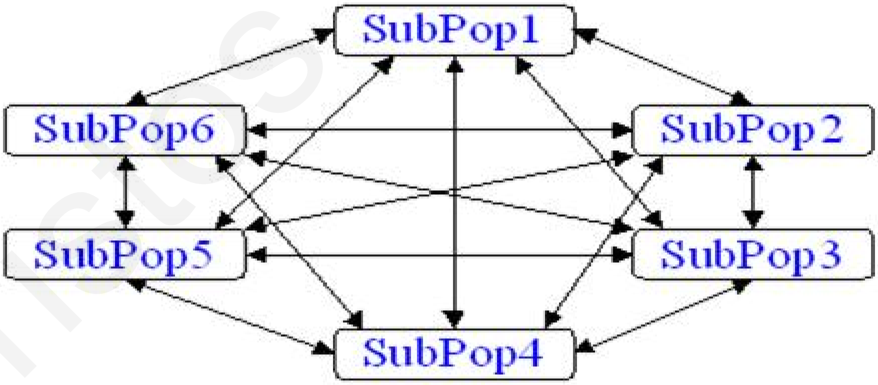


Figure 13: Sub-populations Model for a coarse-grained Parallel Evolutionary Algorithm (PEA) with complete migration topology, provided by [263].

In a fine-grained PEA, also known as global model or Master/Slave, the population is not divided. Instead, the global model employs the inherent parallelism of evolutionary algorithms, i.e. the presence of a population of individuals, and features of the classical evolutionary algorithm. The calculations where the whole population is needed - Pareto-ranking and selection - are performed by the master. All remaining calculations, which are performed for one or two individuals at a time, are

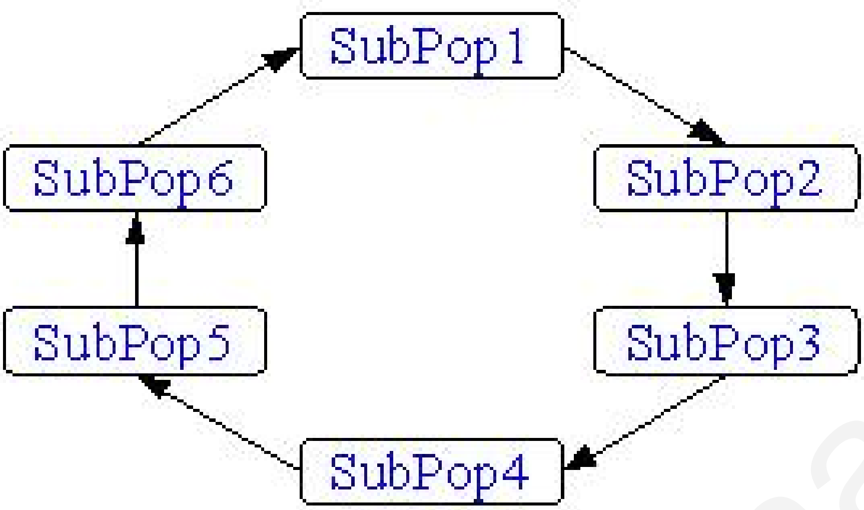


Figure 14: Sub-populations Model for a coarse-grained Parallel Evolutionary Algorithm (PEA) with ring migration topology, provided by [263].

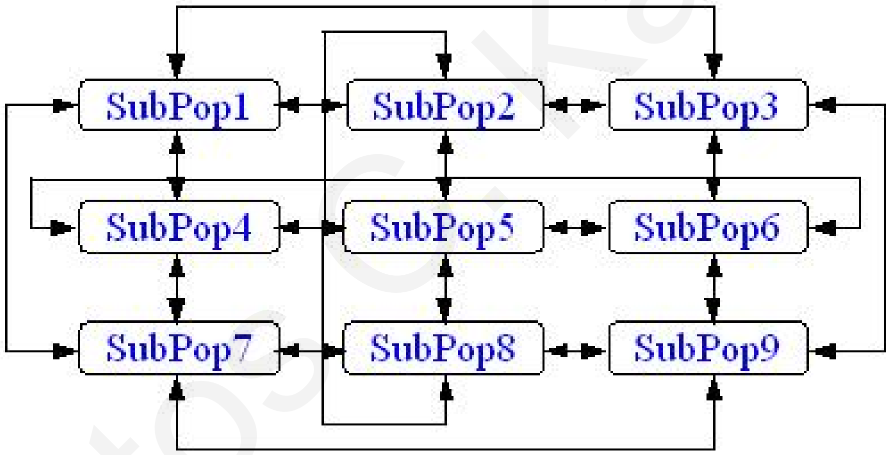


Figure 15: Sub-populations Model for a coarse-grained Parallel Evolutionary Algorithm (PEA) with neighbourhood migration topology, provided by [263].

distributed to a number of slaves. The slaves perform recombination, mutation and the evaluation of the objective function separately. This is known as synchronous master-slave structure, shown in Figure 16 [263], [264] and [265].

PMEGA operates on one population set referred to as working population. The algorithm randomly splits the working population to several sub-populations and uses a predefined pool of processes, to which it assigns tasks for execution. An example of a task is the independent evolution of a sub-population set. Sub-populations are evolved independently for a specific number of iterations

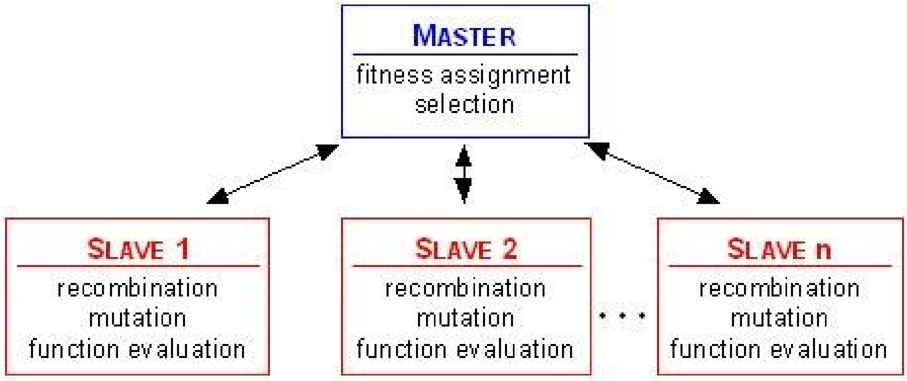


Figure 16: Sub-populations Model for a coarse-grained Parallel Evolutionary Algorithm (PEA) with neighbourhood migration topology, provided by [263].

defined by a user-supplied epoch_counter, which is set to a percentage of the total iterations the algorithm has to run. The default setting of PMEGA is set to 10% of total iterations. The independent evolution of each sub-population is a scaled-down execution of MEGA algorithm as shown in Figure 12. Specifically, during execution time a pre-constructed process from the pool of processes is assigned a task i.e. to execute a scaled-down MEGA. The working population of the process/task is set to a sub-population set and the number of iterations is set to the epoch_counter. During the evolution of sub-populations, migrations are not permitted between the sub-populations. Upon completion of the task, the process returns the results produced and gets assigned a new task, if one is pending. When all sub-populations complete their evolution, their results are gathered and merged. The new working population is created from the merger of the resulting populations, provided by the set of task executions. Following PMEGA checks for the termination conditions; if satisfied the process terminates. However, if this is not the case the process moves to repeat the previous steps. A diagram of PMEGA is shown in Figure 17.

In our article about PMEGA by Kannas *et al.* [266], based on the experiments performed we concluded the following: (a) With respect to the quality of the solutions produced, MEGA and PMEGA behave comparably, (b) The differences observed between the final Pareto-front approximations produced, are partly due to the way PMEGA splits the working population into sub-populations, it splits the population in a random fashion without using any knowledge related to the morphology of the Pareto-approximation and the density of solutions at any region of the search space, (c) With respect

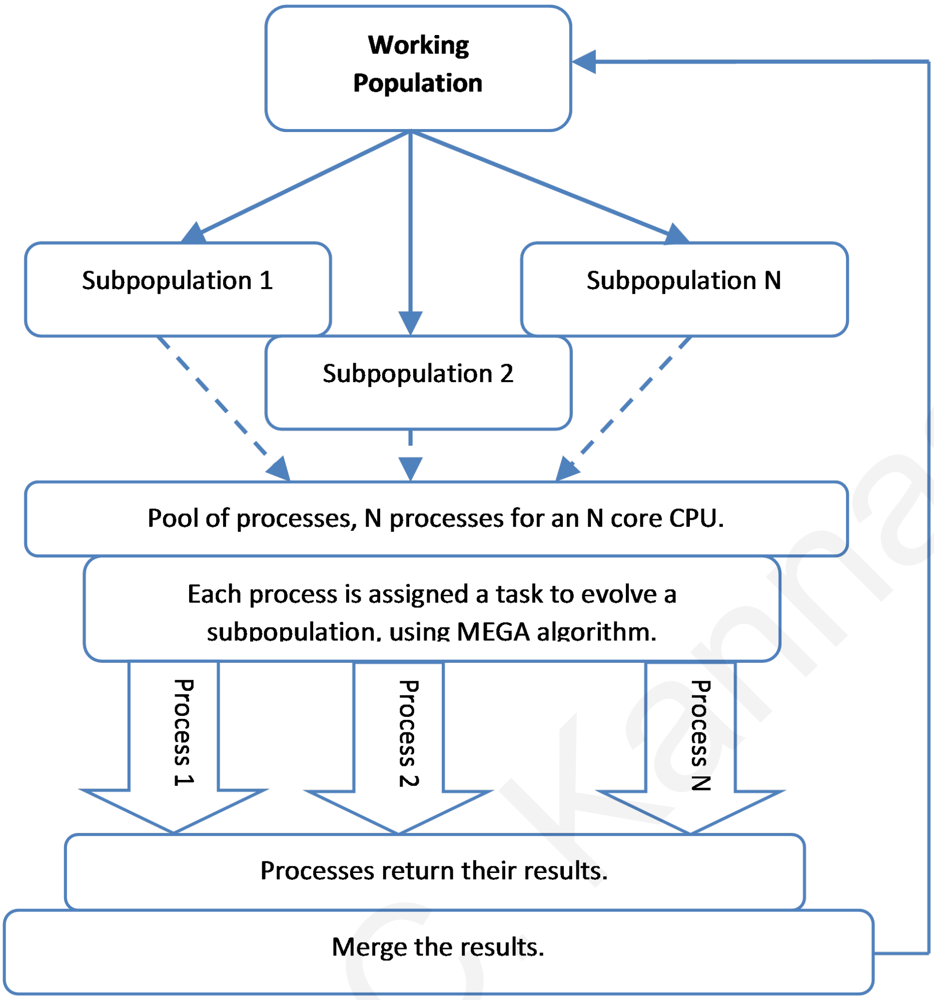

Working Population
Subpopulation 1
Subpopulation 2
Subpopulation N
Pool of processes, N processes for an N core CPU.
Each process is assigned a task to evolve a subpopulation, using MEGA algorithm.
Process 1
Process 2
Process N
Processes return their results.
Merge the results.

Figure 17: The Parallel Multi-Objective Evolutionary Graph Algorithm (PMEGA) framework, developed by Kannas *et al.* [266].

to execution times PMEGA achieves a speedup of almost 1.6 on a common dual-core CPU which is considerable, especially for large experimental applications, and (d) Using PMEGA can provide us with equivalent solution sets in substantially less time.

Future work on PMEGA will focus on algorithmic improvements in the way sub-populations are selected with the aid of knowledge-driven approaches in order to improve the quality of the optimization search and reduce the number of iterations needed for convergence.

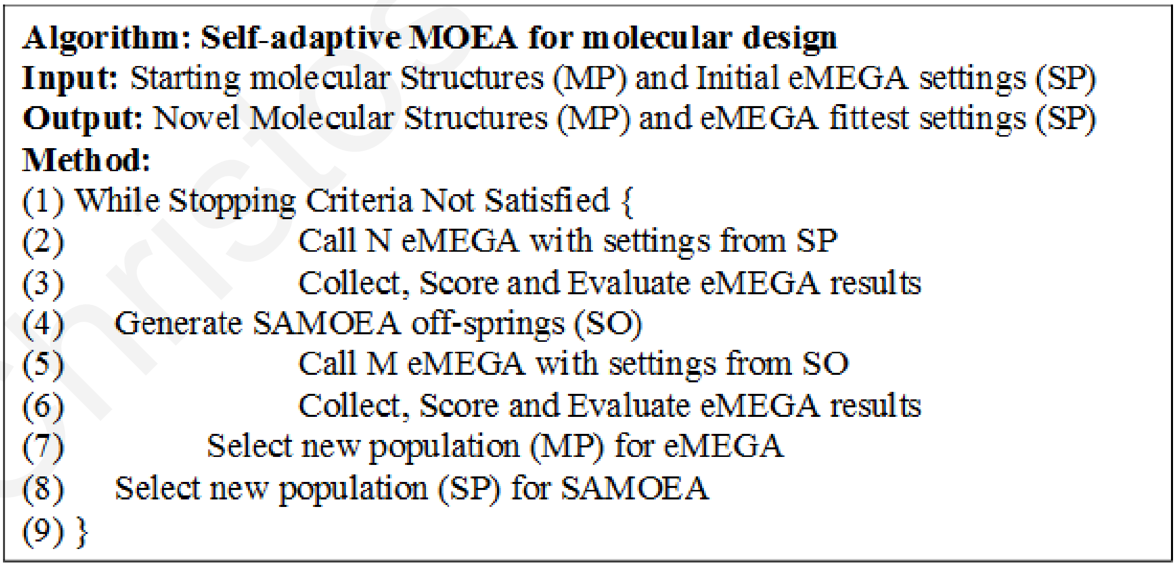
5.3 Self-Adaptive Multi-Objective Evolutionary Algorithm

The Self-Adaptive Multi-Objective Evolutionary Algorithm proposed in this dissertation is an algorithm based on the research from Grefenstette [33] and Shahsavar *et al.* [40] to implement a Self-Adaptive version of elite Multi-Objective Evolutionary Graph Algorithm (eMEGA) proposed in

[78] and [245]. The goal is to have a self-adaptive eMEGA to run a smaller experiment based on a large experiment that we would like to run with eMEGA [78] and [245], for identifying a range of settings that we can use for our eMEGA with the problem we are trying to solve, in order to get better solutions.

The proposed Self-Adaptive MOEA is a meta-level algorithmic approach influenced by Grefenstette [33] and Shahsavar *et al.* [40]. The meta-level/outer level is the algorithm that is responsible for the self adaptive techniques and is a Multi-Objective Genetic Algorithm (MOGA) implementation. The inner level is the actual eMEGA, shown in dotted rectangles in Figure 12. The algorithmic framework of Self-Adaptive MOEA is shown in Figure 19.

The outer level of Self-Adaptive MOEA is a MOGA that operates on two population sets, the working population and the Pareto Archive (PA) set. The working population consists of individuals subjected to objective performance calculation and obtained through evolution in a single iteration. The PA supports a form of elitism aimed at preserving promising solutions found throughout evolution and ensuring that the final Pareto approximation will contain the best solutions found [267]. The pseudocode of Self-Adaptive MOEA is shown in Figure 18.



```
Algorithm: Self-adaptive MOEA for molecular design
Input: Starting molecular Structures (MP) and Initial eMEGA settings (SP)
Output: Novel Molecular Structures (MP) and eMEGA fittest settings (SP)
Method:
While Stopping Criteria Not Satisfied {
            Call N eMEGA with settings from SP
            Collect, Score and Evaluate eMEGA results
    Generate SAMOEA off-springs (SO)
            Call M eMEGA with settings from SO
            Collect, Score and Evaluate eMEGA results
        Select new population (MP) for eMEGA
    Select new population (SP) for SAMOEA
}
```

Figure 18: The Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) pseudocode.

The individuals the algorithm operates on, are chromosomes of fixed length that are generated from an alphabet where each gene has a different type and value ranges. The chromosome is shown in Figure 20. In regards to the chromosome details, the gene in position 0 represents the mutation probability, which is a random selection from a Cauchy distribution in a user provided range, the gene in position 1 represents the crossover probability, which is a random selection from a Cauchy distribution [268] in a user provided range, the gene in position 2 represents the selection type of eMEGA, which can be one of ”best”, ”tournament” and ”roulette”, and finally the gene in position 3 represents the diversity type of eMEGA, which can be one of ”phenotype” and ”genotype”. In Figure 21 examples of Self-Adaptive MOEA’s chromosome are depicted, and their crossover outcome.

Self-Adaptive MOEA due to its chromosome nature and size it uses only one type of mutation, flip bit/gene, and one point crossover.

During the development of Self-Adaptive MOEA we implemented the following objective fitness functions to evaluate its population. These objective fitness functions are:

- **Non Dominated Solutions Percentage:** where for each individual it calculates the percentage of non dominated solutions over the total number of solutions.

- **Unique Solutions Percentage:** where for each individual it calculates the percentage of unique solutions over the total number of solutions.

- **Pareto Front Hypervolume:** where for each individual it calculates the hypervolume [41] of its PFs. Hypervolume measures the space covered by each PF from a reference point. This might be the target if it is known or a starting point from the initial population. i.e. if the reference point is a starting point then the PF with the larger hypervolume value yields better results.

At initialization phase of Self-Adaptive MOEA (Figure 19) the algorithm initializes the Cauchy Distribution sample ranges for the mutation and crossover probabilities, initializes the alphabet to be used, initializes the starting working population in a random manner and then initializes the MOOP objective fitness function scorer.

Once Self-Adaptive MOEA initializes then it generates the starting working population for the second/inner level eMEGAs (Figure 19).

At the training phase, the first step includes running one iteration of several eMEGAs in parallel, collecting and evaluating their results, then applying Self-Adaptive MOEA's MOOP objective fitness function scorer to obtain the list of fitness scores for each individual in Self-Adaptive MOEA's working population. The list of fitness scores are used to select the parents to be reproduced, and later are used in the Pareto ranking procedure to set the rank for each individual. The second step includes Self-Adaptive MOEA's reproduction process, where it is running offsprings eMEGAs in parallel, using a set of generated offsprings settings, collecting and evaluating their results. Then in the third step, the algorithm merges the working population with the offspring population and applies Self-Adaptive MOEA's MOOP objective fitness function scorer to obtain the list of fitness scores for each individual in Self-Adaptive MOEA's merged population. The combined population forms the new working population. The algorithm then proceeds to calculate an efficiency score for each individual. The efficiency score of each individual is then used to update the PA. The current PA is replaced with a subset of the working population that favours individuals with high efficiency score. Following that the algorithm selects the new working population. After that Self-Adaptive MOEA checks for stopping criteria.

When the algorithm meets its stopping criteria returns Self-Adaptive MOEA's last working population, which are the proposed eMEGA settings and the last working population of eMEGAs instances.

Self-Adaptive MOEA inherits features that are found in MEGA line-up of algorithms. In order to avoid duplicate work and the resulting performance degrade, Self-Adaptive MOEA incorporates two additional mechanisms worth special mention. The first mechanism is a chromosome cache that contains each and every chromosome evaluated during the execution of the algorithm. This includes all members of the initial population as well as the complete set of offspring generated in all iterations. The size of the cache is limited since it only includes the identity (ID) and fitness scores of the chromosome measured in some previous iteration. An associative memory hash data structure is used to store the cache to ensure negligible cost to the execution run time. When new chromosomes need to be evaluated against the set of objectives the cache is used to identify whether a specific chromosome has been previously scored and, if so, omit the potentially costly fitness evaluation process and return the values calculated previously. The choice of a chromosome ID is crucial to the success of this scheme since it needs to guarantee that different chromosomes have different IDs and identical chromosomes have the same ID. The second mechanism, active during the evolutionary steps, simply checks and removes those offspring that are identical to some parent chromosomes.

Pareto archiving is an elitist mechanism designed specifically to preserve good non-dominated solutions from getting lost [133]. The mechanism uses a secondary population where non-dominated solutions found during previous iterations are stored. In each iteration, Self-Adaptive MOEA merges the PA with the current population before the efficiency score calculation step and uses this larger set as the current, working population. This extended population is used during the parent selection step. The PA is then reset based on the efficiency scores of the extended working population. Note that the size of the PA is typically set to a large number so as to allow the storage and preservation of a number of solutions exceeding the user-defined population size. When the number of non-dominated solutions exceeds the size of the PA, clustering of the solutions is used to appropriately reduce the number of the elite solutions. Specifically, solutions are eliminated from the most populous clusters while care is exercised to preserve solutions from under-represented clusters. The mechanism is a result of observations made during runs of initial versions of Self-Adaptive MOEA where some promising solutions were lost due to the large number of Pareto solutions found. This paradox, partly caused by the success of Pareto based MOOP methods in generating large, dense populations with multiple non-dominated solutions, resulted in the obligatory elimination of good solutions since the number of non-dominated individuals exceeds the size of the population. Zitzler [133] already identified the problem and proposed techniques based on PF archiving and creation of an elite population of solutions.

Self-Adaptive MOEA has a unique feature that separates it from MEGA line-up of algorithms, as briefly described above, it uses multi-alphabet based chromosomes where each gene has different type and value ranges. The chromosome has been described above and is shown in Figure 20. The genetic operators that can be applied on Self-Adaptive MOEA's chromosomes are due to their nature:

- **Flip gene mutation**, selects a random gene and then changes its value with a random selection from the appropriate set of sample values, and
- **One point crossover**, where two chromosomes are split in a random point and then their respective parts are rejoined to form two new chromosomes.

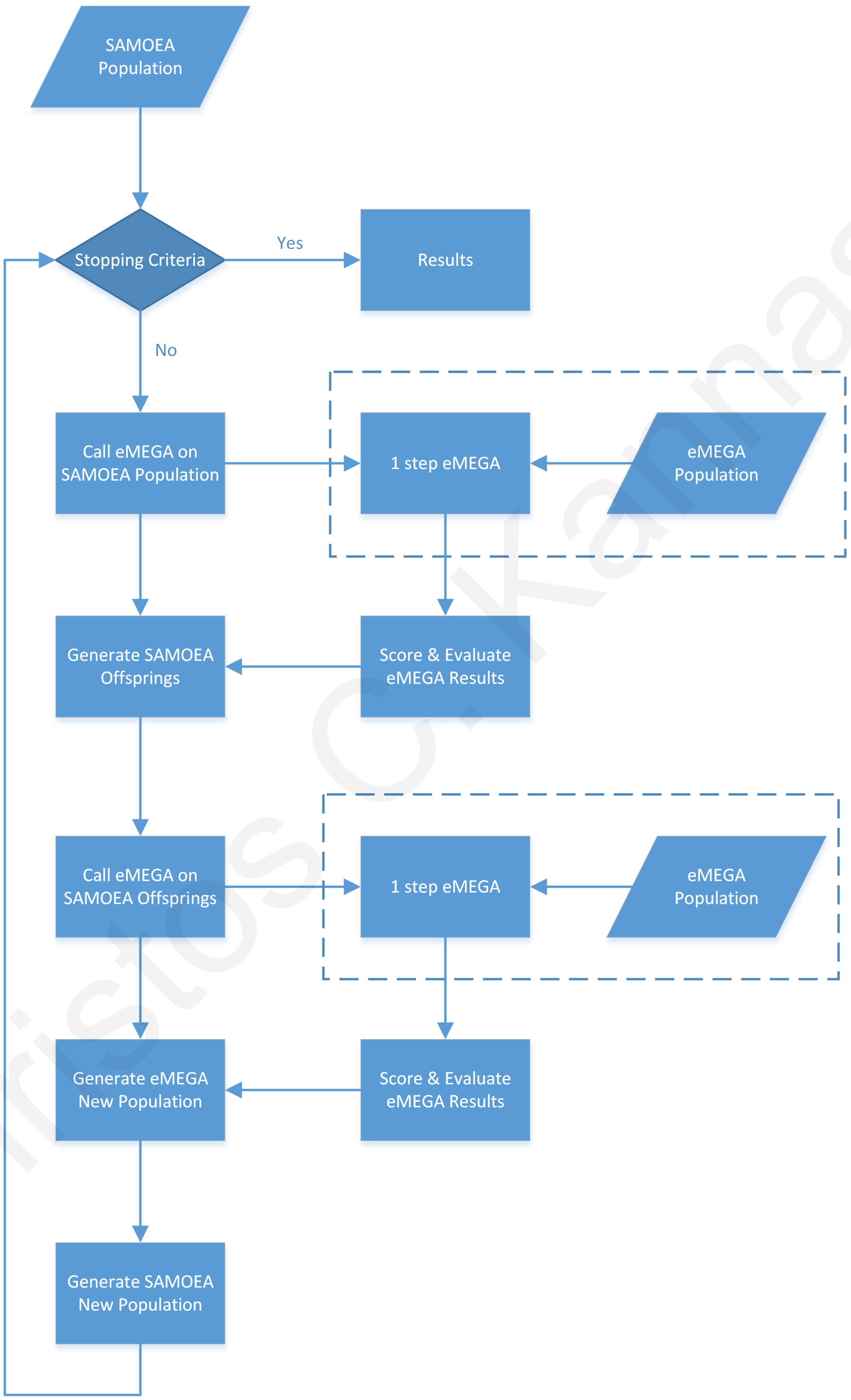

SAMOEA Population
Stopping Criteria
Yes
Results
No
Call eMEGA on SAMOEA Population
1 step eMEGA
eMEGA Population
Generate SAMOEA Offsprings
Score & Evaluate eMEGA Results
Call eMEGA on SAMOEA Offsprings
1 step eMEGA
eMEGA Population
Generate eMEGA New Population
Score & Evaluate eMEGA Results
Generate SAMOEA New Population

Figure 19: The Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) framework.

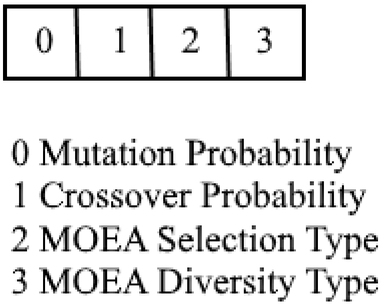


Figure 20: The Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) chromosome, in detail. The gene in position 0 represents the mutation probability, which is a random selection from a Cauchy distribution in user provided range, the gene in position 1 represents the crossover probability, which is a random selection from a Cauchy distribution in user provided range, the gene in position 2 represents the selection type of Multi-Objective Evolutionary Algorithm (MOEA), which can get one of ”best”, ”tournament” and ”roulette”, and finally the gene in position 3 represents the diversity type of Multi-Objective Evolutionary Algorithm (MOEA), which can get one of ”phenotype” and ”genotype”.

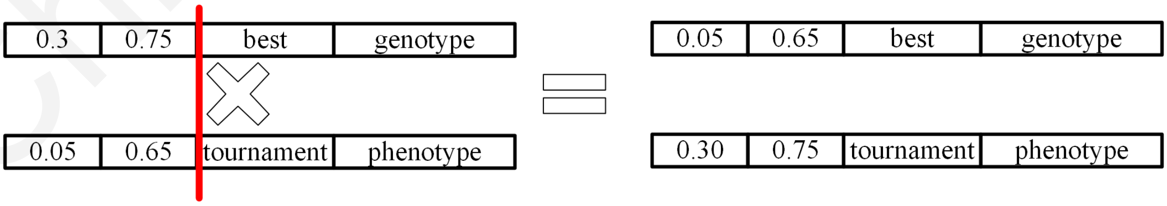


Figure 21: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) chromosome examples. Example depiction of crossover outcome.

Chapter 6

Results and Discussion for Self-Adaptive Multi-Objective Evolutionary Algorithm in Molecular De Novo Design

This chapter describes the tests performed and the results obtained. The purpose of the first experiment (Section 6.1) Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) was to compare it with elite Multi-Objective Evolutionary Graph Algorithm (eMEGA) and Multi-Objective Algorithm for Replacement of Fragments (MOARF) [254] in a problem with a known target. In the second experiment (Section 6.2), Self-Adaptive MOEA was used to design Estrogen Receptor-α (ER-α) inhibitors based on structural similarity to Tamoxifen and structural dissimilarity to Ibuproxam. In the third experiment (Section 6.3), Self-Adaptive MOEA was used to design ER-α inhibitors based on structural and chemical properties similarity to Tamoxifen. In the fourth experiment (Section 6.4), Self-Adaptive MOEA was used to design ER-α inhibitors based on structural and chemical properties similarity to Raloxifene. In the fifth experiment (Section 6.5), Self-Adaptive MOEA was used to design Proteasome B5 inhibitors based on structural and chemical properties similarity to Ixazomib.

For the experiment described in Section 6.1, Self-Adaptive MOEA uses the following two objective fitness functions:

- **Non Dominated Solutions Percentage:** where for each individual it calculates the percentage of non dominated solutions over the total number of solutions.

- **Unique Solutions Percentage:** where for each individual it calculates the percentage of unique solutions over the total number of solutions.

For the experiments described in Sections 6.2 to 6.5, Self-Adaptive MOEA uses the following two objective fitness functions:

- **Non Dominated Solutions Percentage:** where for each individual it calculates the percentage of non dominated solutions over the total number of solutions.
- **Pareto Front Hypervolume:** where for each individual it calculates the hypervolume [41] of its Pareto Front (PF)s.

All experiments were performed on a Linux Virtual Machine, with the specifications shown in Table 5. A note in regards to the way eMEGA and Self-Adaptive MOEA work and how they utilise the machine they run on: eMEGA utilises only a single process while Self-Adaptive MOEA utilises three processes, one is used by the outer loop algorithm and the remaining two are used to run up to 2 instances of the inner loop algorithm. Self-Adaptive MOEA can be enabled to utilise all available cores of the machine but we restricted it to using only three cores as we run to memory usage issues.

Table 5: Specifications of the computational system the experimental runs were performed

Linux Virtual Machine	
CPU	4x Virtual CPU @ 2GHz
RAM	16GB
OS	CentOS 6

6.1 Validation of Self-Adaptive Multi-Objective Evolutionary Algorithm

This experiment is a means to compare Self-Adaptive MOEA with eMEGA and MOARF in a well defined problem where there is a single known target. There are documented solutions from MOARF that approximate the target in both chemical properties and structure [254]. The target of the experiment is the known CDK2 inhibitor Seliciclib (CYC202, R-roscovitine)[1] seen in Figure 22.

[1] `https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL14762`

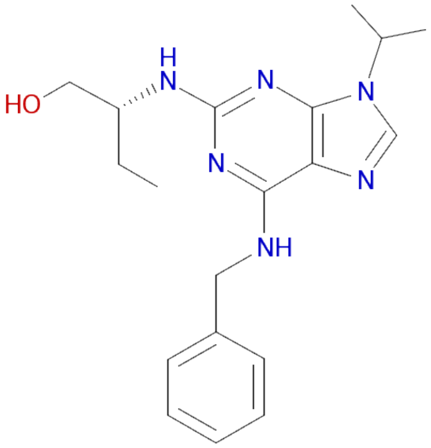


Figure 22: Seliciclib (CYC202, R-roscovitine).

6.1.1 Methodology

The objective of the experiment was to design molecules that have structural and chemical descriptors similarity to the target molecule of Seliciclib (CYC202, R-roscovitine) (Figure 22). For Self-Adaptive MOEA there is a secondary objective that is to propose near optimal settings for eMEGA that can be used to obtain hopefully better results by running a tuned eMEGA later.

Structural similarity is an objective fitness function that calculates a fitness score for a molecule to the target molecule, computed as the graph distance based on their Maximum Common Substructure (MCS)[2] , an in-house implementation based on RApid Similarity CALculation (RASCAL) [269].

Chemical descriptors similarity is an objective fitness function that calculates a fitness score for a molecule to the target molecule, computed as the distance of their chemical descriptors vector. The chemical descriptors are based on an in-house implementation of atom pairs and topological torsions calculations for each molecule [270] and [271].

Table 6: elite Multi-Objective Evolutionary Graph Algorithm (eMEGA) experimental design settings

Dataset	Objectives	Population	Iterations	Evolutionary Operations
Dataset 1 Dataset 2	Structural Similarity Chemical Descriptors Similarity	500	500	Mutation Probability: **15%** Crossover Probability: **80%** Selection Type: **Roulette** Diversity Type: **Genotype**

[2] `https://en.wikipedia.org/wiki/Maximum_common_subgraph`

The objectives for the molecular design algorithm were, structural similarity based on Soergel distance [272] and chemical descriptors similarity based on Euclidean distance [272] to the target molecule of Seliciclib (Figure 22). The optimization process was aimed to minimize both of those objectives to 0.

For input we used two freely available commercial molecule datasets, the first is Maybridge's Screening Library[3] that contains 53953 molecules (Dataset 1), and the second is Asinex's Elite Libraries[4] that contains 104577 molecules (Dataset 2).

The experiment was divided into two sub-experiments one for each dataset. The algorithms initial population was selected at random from each dataset. The sub-experiments for eMEGA used a population size of 500. Mutation probability was set to 15% while crossover probability was set to 80%. Parent selection was set to roulette. For the elitist generation selection eMEGA was set to use genotype diversity. Multiple runs, a total of five, were performed for each parameter settings combination with different initial populations to avoid drawing conclusions from chance results produced by single runs. Results were assessed after 500 iterations. A synopsis of eMEGA settings can be found in Table 6.

For Self-Adaptive MOEA the settings were slightly different, due to reasons that are stated in Discussion section 6.1.3. Self-Adaptive MOEA has to initialize two Multi-Objective Evolutionary Algorithms (MOEAs) the first level is responsible the self-adaptive technique and the second level is a set of eMEGAs that perform the molecular design.

The first level MOEA (here on referred as Self-Adaptive MOEA) works on a population size of 20 that are the settings for the second level eMEGAs. Self-Adaptive MOEA's chromosome is shown in Figure 20. Self-Adaptive MOEA operates with a mutation probability set to 15% while the crossover probability was set to 80%. Parent selection was set to roulette. For the elitist generation selection Self-Adaptive MOEA was set to use phenotype diversity. The second level eMEGAs operate on a population size of 100. Multiple runs, a total of five, were performed for each parameter settings

[3] http://www.maybridge.com/portal/alias__Rainbow/lang__en/tabID__146/DesktopDefault.aspx

[4] http://asinex.com/libraries-html/

combination with different initial populations to avoid drawing conclusions from chance results produced by single runs. Results were assessed after 100 iterations. A synopsis of Self-Adaptive MOEA settings can be found in Table 7.

Table 7: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) experimental design settings

<table>
<tr><th colspan="5">Self-Adaptive MOEA</th></tr>
<tr><th>Dataset</th><th>Objectives</th><th>Population</th><th>Iterations</th><th>Evolutionary Operations</th></tr>
<tr><td>Dataset 1</td><td rowspan="2">Non Dominate Solutions Percentage
Unique Solutions Percentage</td><td rowspan="2">20</td><td rowspan="2">100</td><td rowspan="2">Mutation Probability: <b>15%</b>
Crossover Probability: <b>80%</b>
Selection Type: <b>Roulette</b>
Diversity Type: <b>Phenotype</b></td></tr>
<tr><td>Dataset 2</td></tr>
<tr><th colspan="5">eMEGAs</th></tr>
<tr><td>Dataset 1</td><td rowspan="2">Structural Similarity
Chemical Descriptors Similarity</td><td rowspan="2">100</td><td rowspan="2">1</td><td rowspan="2">Defined during run time. Based on Self-Adaptive MOEA's chromosomes.</td></tr>
<tr><td>Dataset 2</td></tr>
</table>

6.1.2 Results

eMEGA runs using the Maybridge dataset returned relatively good results with the majority of their final Pareto front solutions in the range of 0 to 0.3 for both of the objectives.

In Figure 23 we are showing the consolidated results from Maybridge dataset from all runs, shaded by their dominance rank, with a total of 2381 molecules. In Figure 24 we are showing the top 10 non dominated solutions from Maybridge dataset from all runs. All solutions within the range of 0 to 0.5 for both objectives are considered good solutions, though we decided to use the non dominated solutions for the docking experiment just for sake of simplicity. Figure 25 shows the target molecule and the Top 10 molecules, the red highlighted part of the molecules is their common core.

Similarly eMEGA results with the Asinex dataset returned relatively good results with the majority of their final Pareto front solutions in the range of 0 to 0.3 for both of the objectives.

In Figure 26 we are showing the consolidated results from Asinex dataset from all runs, shaded by their dominance rank, with a total of 1835 molecules. In Figure 27 we are showing the top 10 non dominated solutions from Asinex dataset from all runs. Figure 28 shows the target molecule and the Top 10 molecules, the red highlighted part of the molecules is their common core.

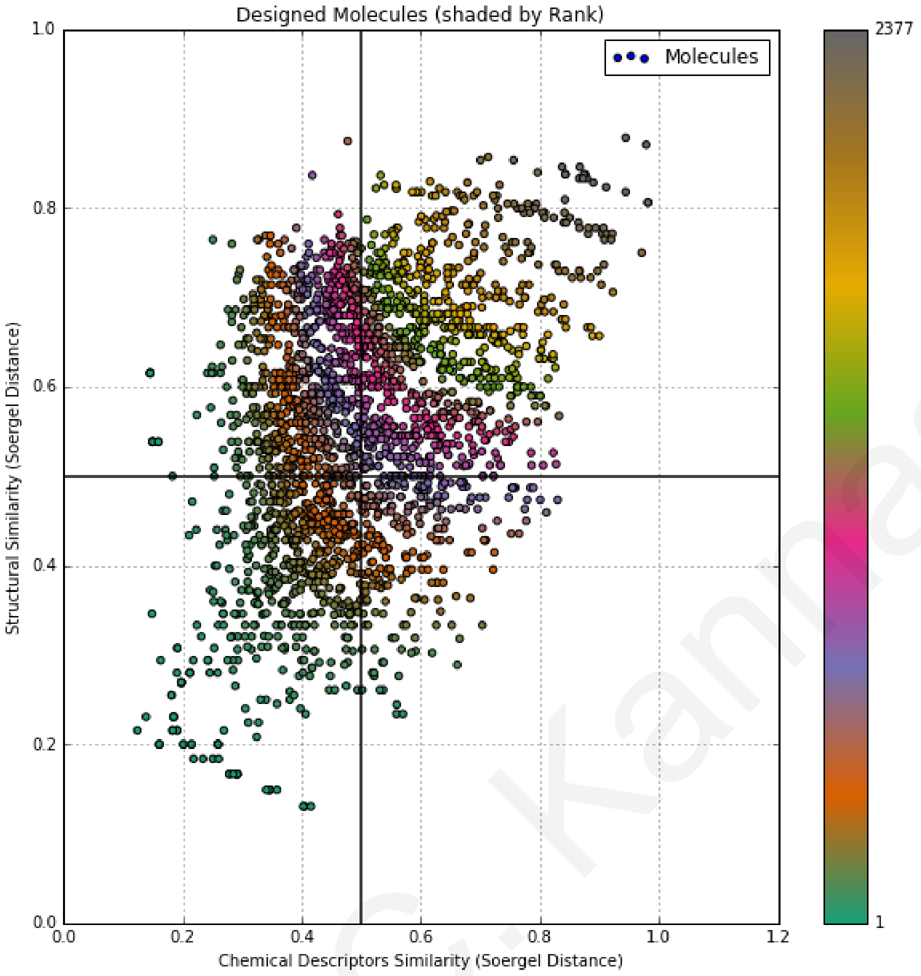


Figure 23: elite Multi-Objective Evolutionary Graph Algorithm (eMEGA) results for Maybridge dataset.

The execution time for each run of eMEGA using Dataset 1 and Dataset 2 are shown in Figure 29.

Self-Adaptive MOEA runs using the Maybridge dataset returned relatively good results with the majority of their final Pareto front solutions in the range of 0 to 0.4 for both of the objectives.

In Figure 30 we are showing the consolidated results from Maybridge dataset from all runs, shaded by their dominance rank, with a total of 473 molecules. In Figure 31 we are showing the top 10 non dominated solutions from Maybridge dataset from all runs. Figure 32 shows the target molecule and the Top 10 molecules, the red highlighted part of the molecules is their common core.

The other important output of Self-Adaptive MOEA are the proposed settings for eMEGA for the given problem. In Table 8 there are the Top 10 proposed settings, collected from all runs of Self-Adaptive MOEA with the Maybridge dataset.

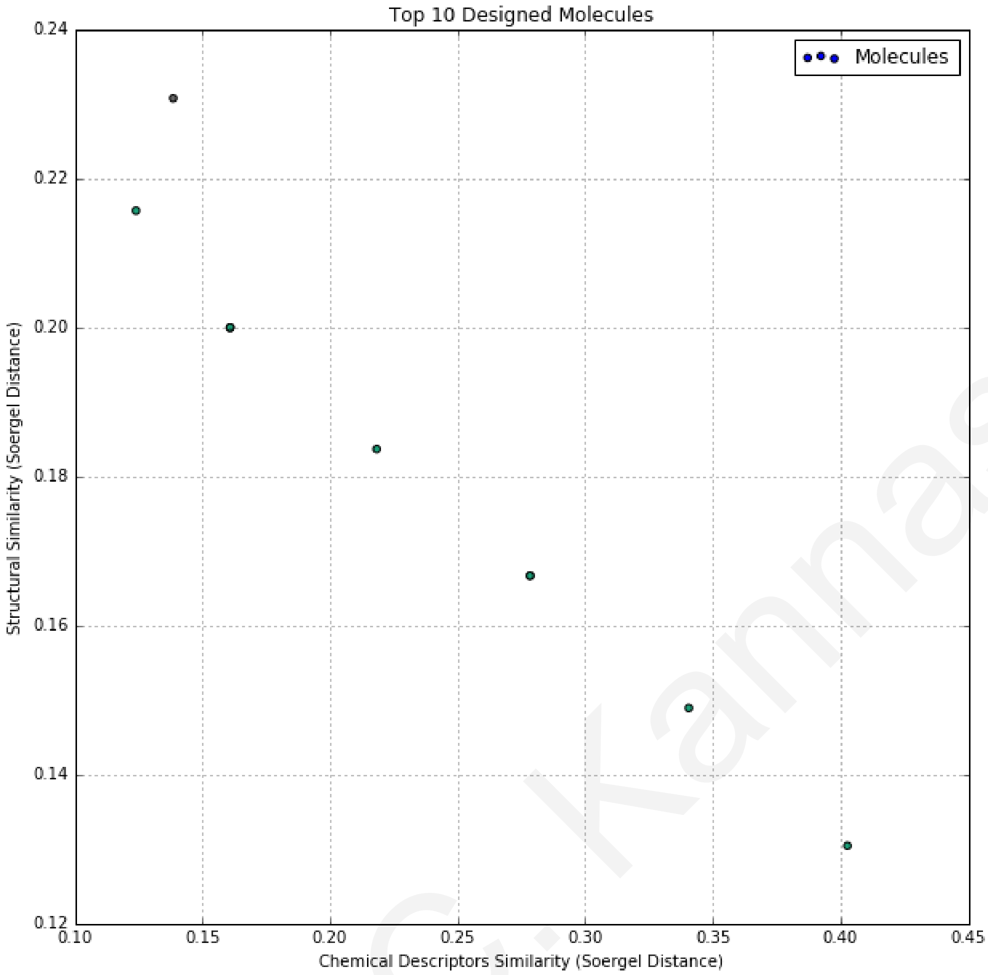


Figure 24: elite Multi-Objective Evolutionary Graph Algorithm (eMEGA) Top 10 results for Maybridge dataset.

Self-Adaptive MOEA runs using the Asinex dataset returned relatively good results with the majority of their final Pareto front solutions in the range of 0 to 0.4 for both of the objectives.

In Figure 33 we are showing the consolidated results from Asinex dataset from all runs, shaded by their dominance rank, with a total of 496 molecules. In Figure 34 we are showing the top 10 non dominated solutions from Asinex dataset from all runs. Figure 35 shows the target molecule and the Top 10 molecules, the red highlighted part of the molecules is their common core.

The other important output of Self-Adaptive MOEA are the proposed settings for eMEGA for the given problem. In Table 9 there are the Top 10 proposed settings, collected from all runs of Self-Adaptive MOEA with the Asinex dataset.

The execution time for each run of Self-Adaptive MOEA using Dataset 1 and Dataset 2 are shown in Figure 36.

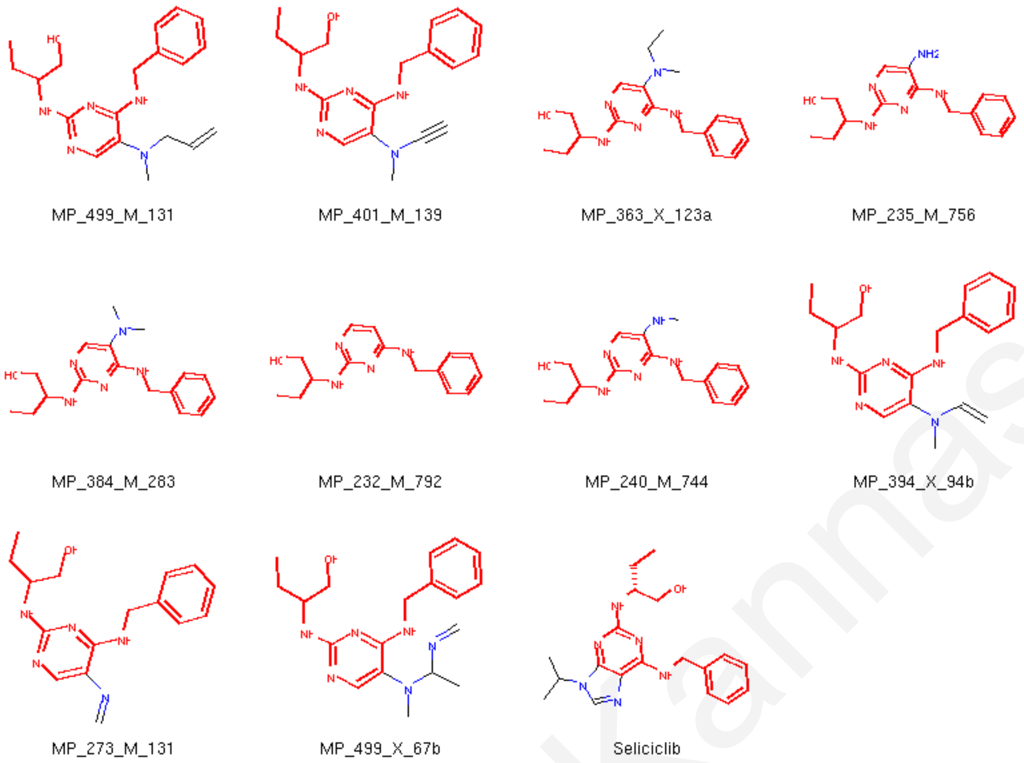


Figure 25: elite Multi-Objective Evolutionary Graph Algorithm (eMEGA) Top 10 results for Maybridge dataset compared with Seliciclib, the red highlighted part of the molecules is their common core.

Finally we compared all the Top 10 results together and with MOARF's results [254]. Figure 37 is a chart of the results compared together and relative to the target, Seliciclib.

MOARF's results molecular structure is shown with comparison to Seliciclib in Figure 38.

6.1.3 Discussion

Population size enables an Evolutionary Algorithm (EA) to prescribe the search space. For example, a larger population size will facilitate a larger search space coverage for the algorithm. Certainly, the search functionality is also dependent on the objectives and the reproduction operators, as they are the functions that guide the search for solutions.

eMEGA and Self-Adaptive MOEA usually have different population sizes due to their different nature of operation. eMEGA needs to have a population size to ensure that it covers satisfactorily the molecular search space. While Self-Adaptive MOEA needs to have a population size to ensure that it covers satisfactorily the search parameters space of the internal eMEGAs. The larger the

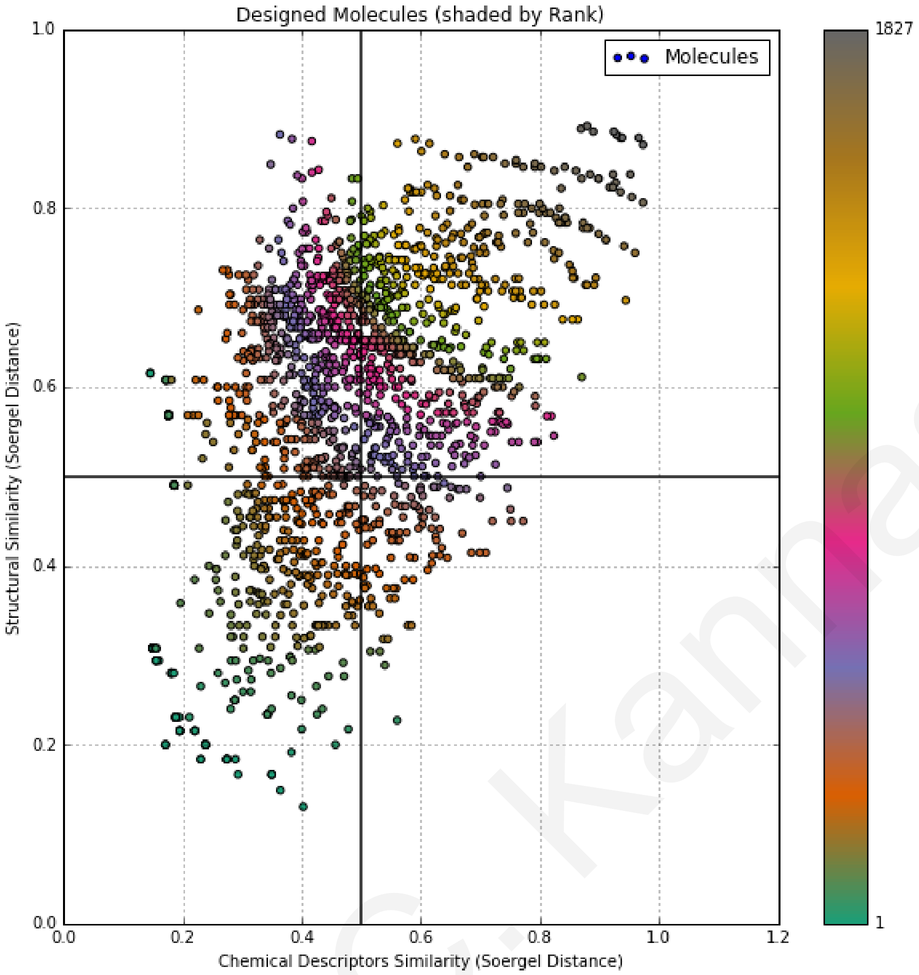


Figure 26: elite Multi-Objective Evolutionary Graph Algorithm (eMEGA) results for Asinex dataset.

population size is, more data have to be stored in memory, more offsprings will have to be generated, more scoring and evaluations will have to be performed and finally more comparisons will have to be computed during the clustering that is required for the selection of the next generation.

From Figures 25 and 28 we can see that the molecules generated from eMEGA are very similar to the target molecule, Seliciclib (Figure 22). The molecules generated from Maybridge dataset (Figure 25) have a different common core with Seliciclib, than the molecules generated from Asinex dataset (Figure 28). Similarly Figures 32 and 35 depict Self-Adaptive MOEA's molecules in comparison with Seliciclib, where we can see that are very similar to the target molecule, Seliciclib (Figure 22). Again the molecules generated from Maybridge dataset (Figure 32) have a different common core with Seliciclib, than the molecules generated from Asinex dataset (Figure 35). In Figure 38 we are seeing that MOARF's molecules are very similar to the target molecule, Seliciclib (Figure 22). Though their common core is different from the common cores from eMEGA and Self-Adaptive MOEA results.

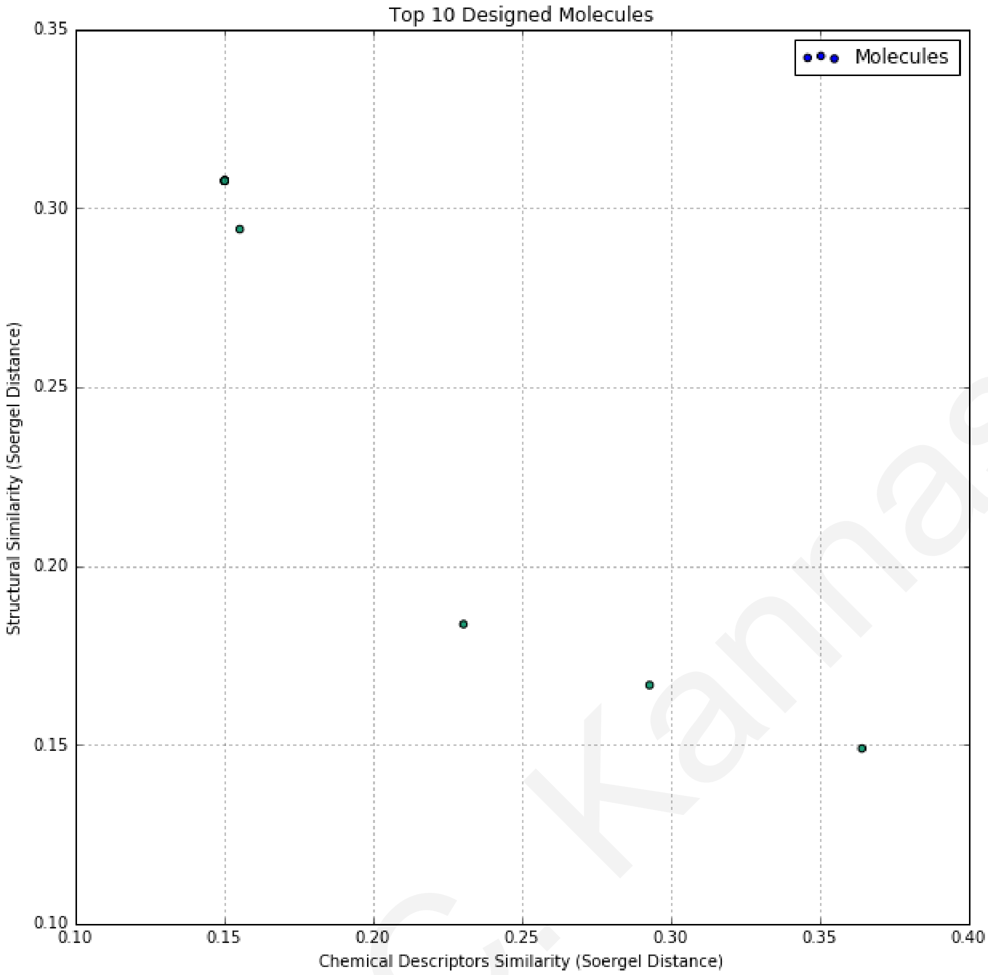


Figure 27: elite Multi-Objective Evolutionary Graph Algorithm (eMEGA) Top 10 results for Asinex dataset.

The common core between Seliciclib and each respective experiment and algorithm is different, there are only minor similarities between them. When further more comparing all common cores between them from Figures 25, 28, 32, 35 and 38 we see that the common cores are different. Each algorithm and experiment finds a different common core, the only similarity is the central aromatic ring with the two nitrogens.

From Figure 37 we can see that MOARF approximates the target molecule better than eMEGA and Self-Adaptive MOEA, because it generates new molecules: (a) in a more chemical correct way, with less stochastic operations, and (b) it starts from a selected core for the target and then attaches new fragments on to it. Self-Adaptive MOEA seems to explore the space better than eMEGA and MOARF, despite the fact that its proposed solutions are not as good as MOARF's or eMEGA's ones. That is because Self-Adaptive MOEA during the iterative process it bounces all over the place while

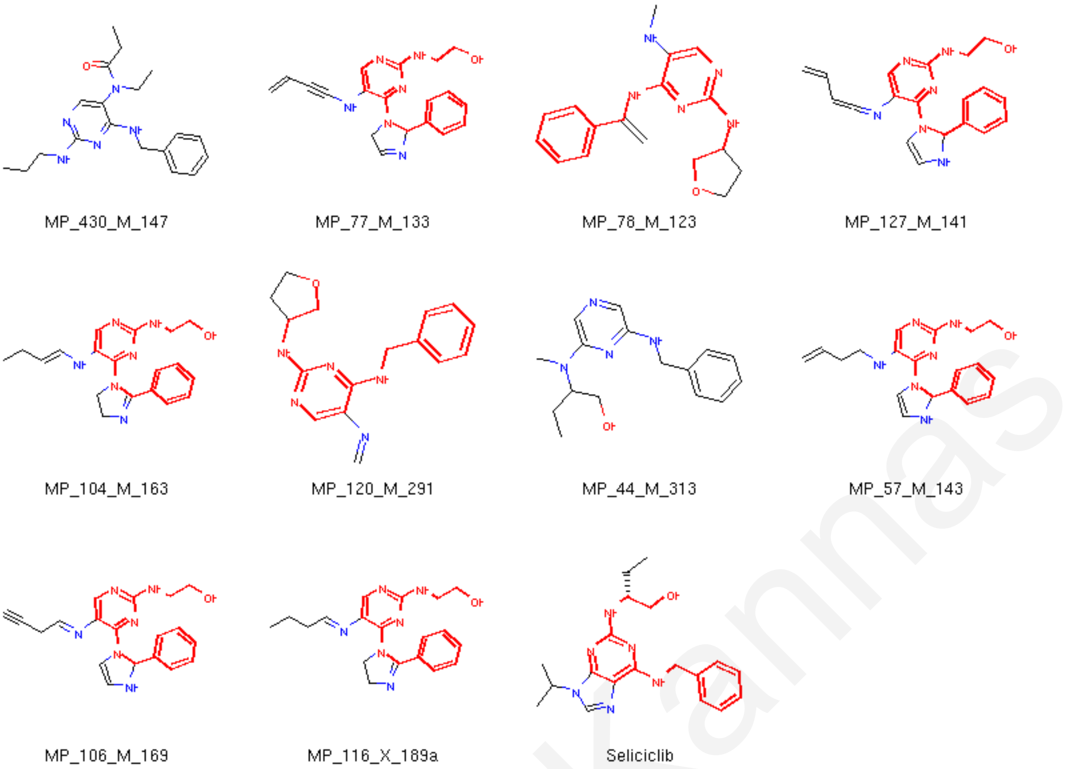
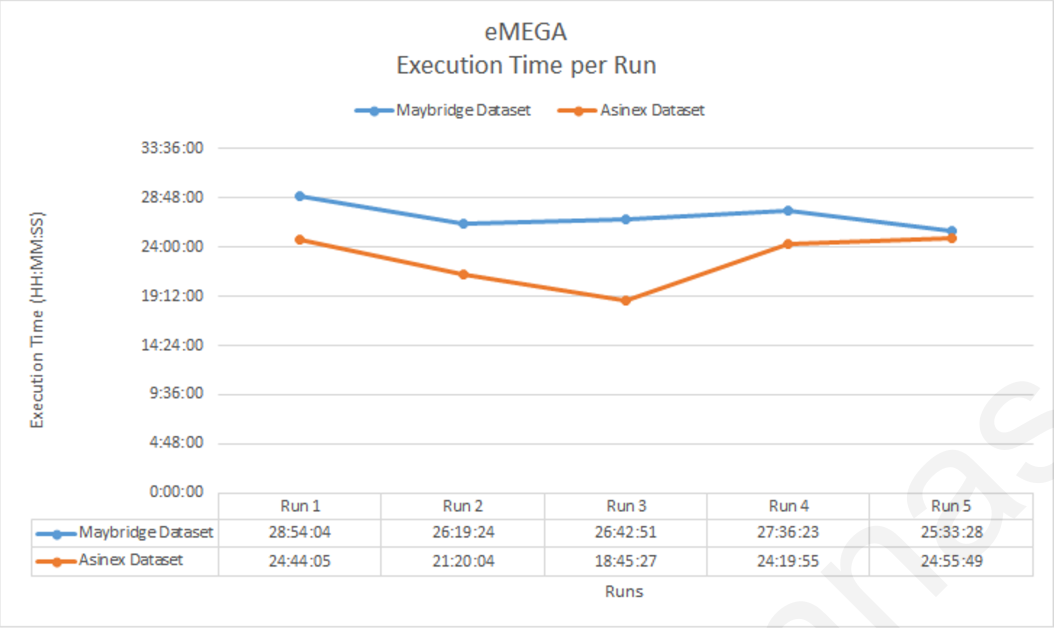


Figure 28: elite Multi-Objective Evolutionary Graph Algorithm (eMEGA) Top 10 results for Asinex dataset compared with Seliciclib, the red highlighted part of the molecules is their common core.

trying to select both the fittest solutions from eMEGAs and the fittest settings for those eMEGAs that most of the time contradict from one eMEGA instance to another.

From Tables 8 and 9 we can see that different settings are favoured for each dataset. For the Maybridge dataset a mutation probability around 17%, a crossover probability around 80%, for selection type either roulette or tournament and for diversity type both selections are valid ones, are the preferred options. For the Asinex dataset a mutation probability around 10%, a crossover probability above 96%, for selection type either best or tournament and for diversity type both selections are valid ones, are the preferred options. From this we can understand that the datasets behave differently for the given problem, Maybridge seems to prefers more balanced settings, while Asinex dataset seems to prefers crossover more which means that the algorithm prefers to combine parts of molecules to changing a molecule slightly. It is a surprising result to see that for Asinex dataset the algorithm prefers best as selection type over roulette. Tournament selection in general is known to be more versatile than best and roulette, so there is no surprise why it is preferred for both datasets. Something important that we notice from these results, is that the objective fitness scores for the proposed



	Run 1	Run 2	Run 3	Run 4	Run 5
Maybridge Dataset	28:54:04	26:19:24	26:42:51	27:36:23	25:33:28
Asinex Dataset	24:44:05	21:20:04	18:45:27	24:19:55	24:55:49

Figure 29: elite Multi-Objective Evolutionary Graph Algorithm (eMEGA) execution time per run for Maybridge and Asinex datasets. Time is wall clock time.

settings are very high, which means that the actual percentage is really low, below 5%. From this we can conclude the following: (a) the eMEGA instances generate a large number of identical solutions, despite the fact that they have different configurations, this is something that we noticed with previous experiments when comparing Multi-Objective Evolutionary Graph Algorithm (MEGA), eMEGA and Multi-Objective Genetic Algorithm (MOGA) [245], and (b) the objective fitness functions we choose to use in Self-Adaptive MOEA compete with each other, which means that having eMEGAs generating a high number of unique and non dominated solutions (above 20%) proves to be a difficult task.

Self-Adaptive MOEA experiments were configured to have a lower population size and iterations, because: (a) it uses multiple instances of eMEGA during each iteration, which means more solutions are generated and evaluated per iteration, (b) its purpose is to perform a smaller version of the experiment, which will guide us to the configuration of eMEGA for a larger experiment, (c) due to eMEGA's current unoptimised state it requires a lot of resources (especially RAM) for large runs, taking into account that Self-Adaptive MOEA runs multiple instances of eMEGA. During the process of performing the experiments and acquiring results, there were a lot of failures to the process mainly

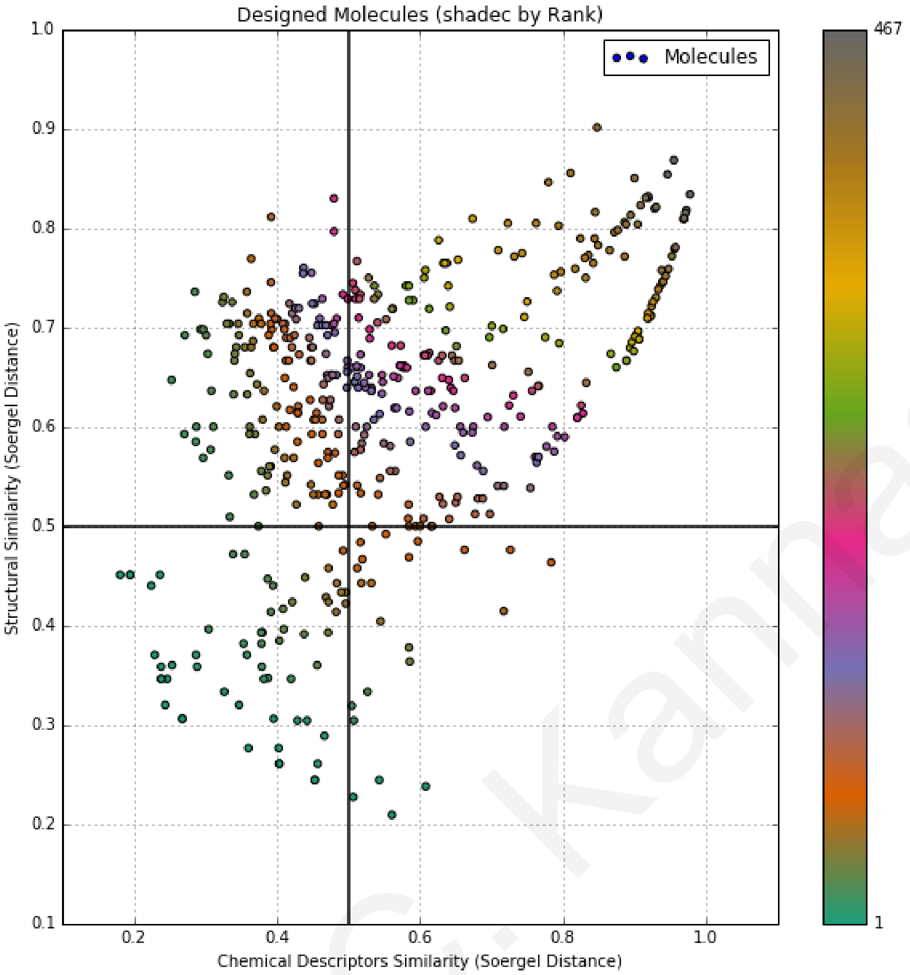


Figure 30: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) results for Maybridge dataset.

due to running out of memory, and a couple of times unfortunately due to unknown reasons, where there was not enough information to specify the problem.

From the execution times of eMEGA for Dataset 1, shown in Figure 29 we can see that eMEGA requires 26 to 27 hours to complete. Similarly for Dataset 2 it requires 24 to 25 hours. When we compare the execution times we see that they are consistent across runs. Dataset size and content might have some influence on the execution time. At the beginning, the dataset size affects the time required to sample the starting population. Further on, the dataset contents affect the evolutionary process.

Similarly the execution times of Self-Adaptive MOEA, shown in Figure 36, are consistent across runs (excluding the outliers). For this experiment the execution time for Maybridge dataset is around 4 to 5 hours and for Asinex dataset around 6 to 7 hours. As to eMEGA dataset size and content might

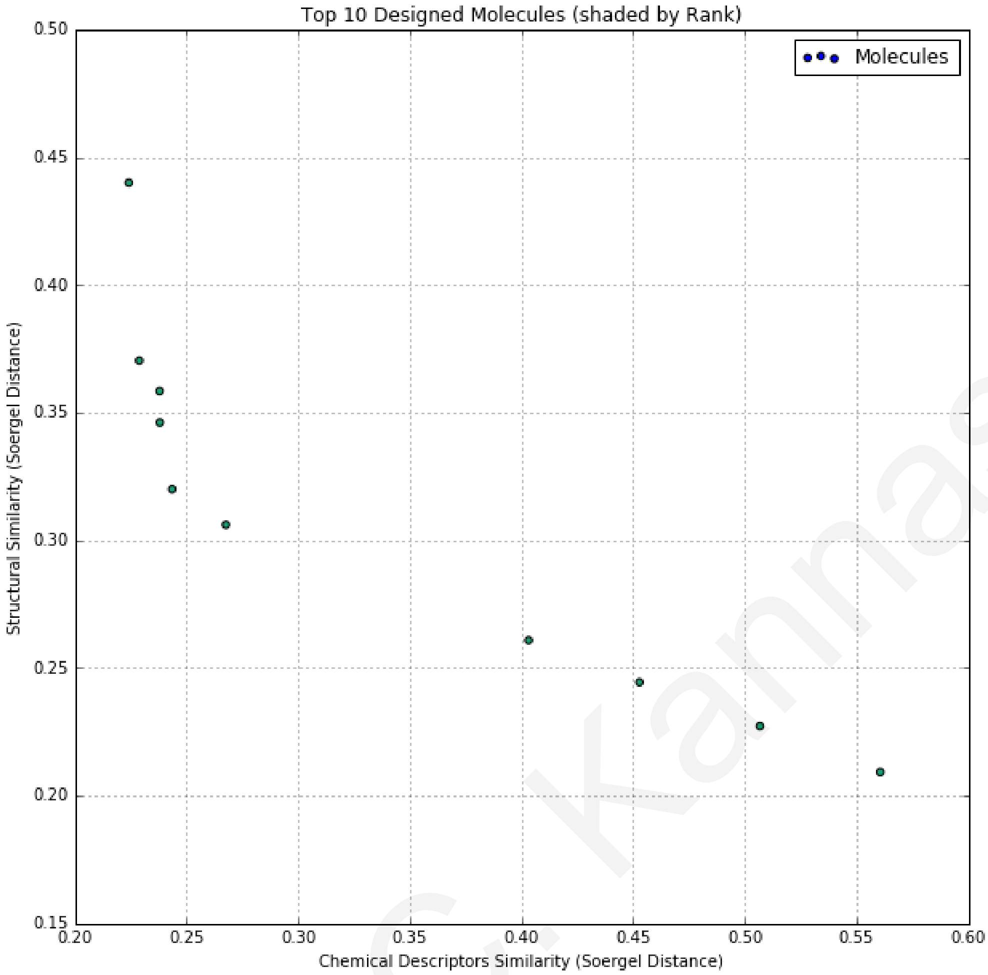


Figure 31: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) Top 10 results for Maybridge dataset.

have some influence on the execution time. At the beginning, the dataset size affects the time required to sample the starting population. Further on, the dataset contents affect the evolutionary process.

When comparing the execution times of Self-Adaptive MOEA and eMEGA we have to do an extrapolation for the execution time of Self-Adaptive MOEA, since for this experiment it performed 100 iterations in comparison to the 500 iterations performed by eMEGA. As such the extrapolated estimate of the total time required by Self-Adaptive MOEA, assuming that each iteration needs the same amount of time, is at least 25 hours.

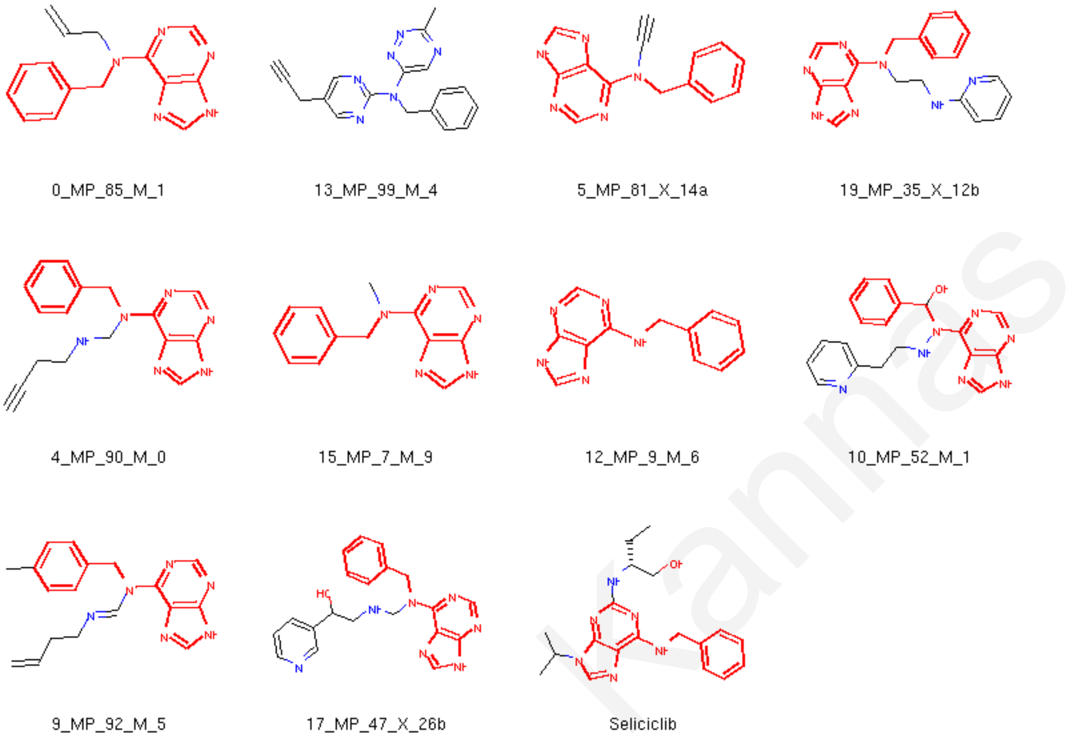


Figure 32: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) Top 10 results for Maybridge dataset compared with Seliciclib, the red highlighted part of the molecules is their common core.

Table 8: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) Top 10 proposed settings for elite Multi-Objective Evolutionary Graph Algorithm (eMEGA) for Maybridge dataset

Mutation Probability	Crossover Probability	Selection Type	Diversity Type	Non Dominated %	Unique Solutions %	Rank
0.03	0.69	roulette	genotype	0.90	0.99	1
0.17	0.82	roulette	phenotype	0.91	0.96	1
0.17	0.81	tournament	phenotype	0.93	0.95	1
0.03	0.69	roulette	phenotype	0.93	0.96	1
0.0	0.96	roulette	phenotype	0.98	0.85	1
0.18	0.81	roulette	phenotype	0.92	0.96	1
0.08	0.73	tournament	phenotype	0.95	0.95	1
0.09	0.80	tournament	genotype	0.98	0.93	1
0.17	0.82	best	genotype	0.91	0.97	2
0.17	0.82	roulette	genotype	0.93	0.96	2

Note: The numbers for 'Non Dominated %' and 'Unique Solutions %' are 1 minus the actual %. The smaller the number listed here the better. 'Rank' is their non dominance rank.

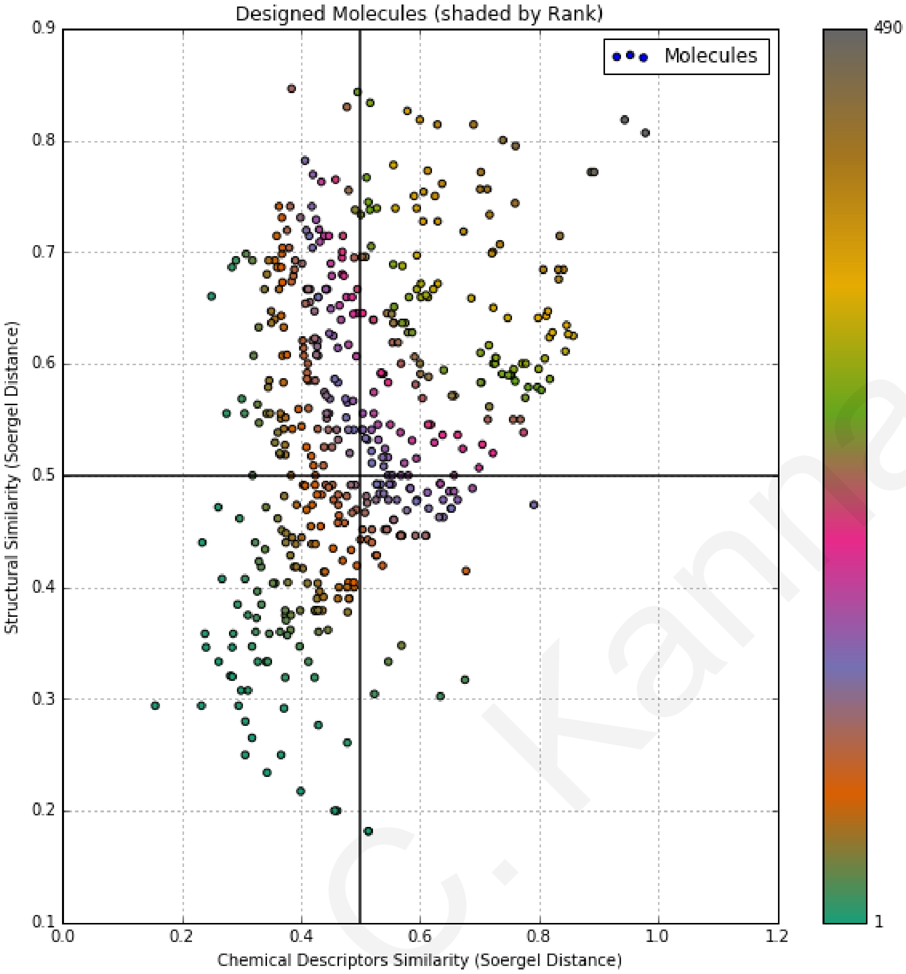


Figure 33: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) results for Asinex dataset.

Table 9: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) Top 10 proposed settings for elite Multi-Objective Evolutionary Graph Algorithm (eMEGA) for Asinex dataset

Mutation Probability	Crossover Probability	Selection Type	Diversity Type	Non Dominated %	Unique Solutions %	Rank
0.11	1.0	best	phenotype	0.99	0.93	1
0.14	0.96	tournament	phenotype	0.96	0.96	1
0.09	0.69	tournament	genotype	0.98	0.94	1
0.14	0.97	best	phenotype	0.96	0.96	1
0.11	0.69	tournament	genotype	0.96	0.96	1
0.10	1.0	best	phenotype	0.99	0.94	1
0.09	0.69	tournament	genotype	0.96	0.96	1
0.14	0.97	roulette	phenotype	0.97	0.95	1
0.09	0.71	tournament	genotype	0.96	0.96	2

Note: The numbers for 'Non Dominated %' and 'Unique Solutions %' are 1 minus the actual %. The smaller the number listed here the better. 'Rank' is their non dominance rank.

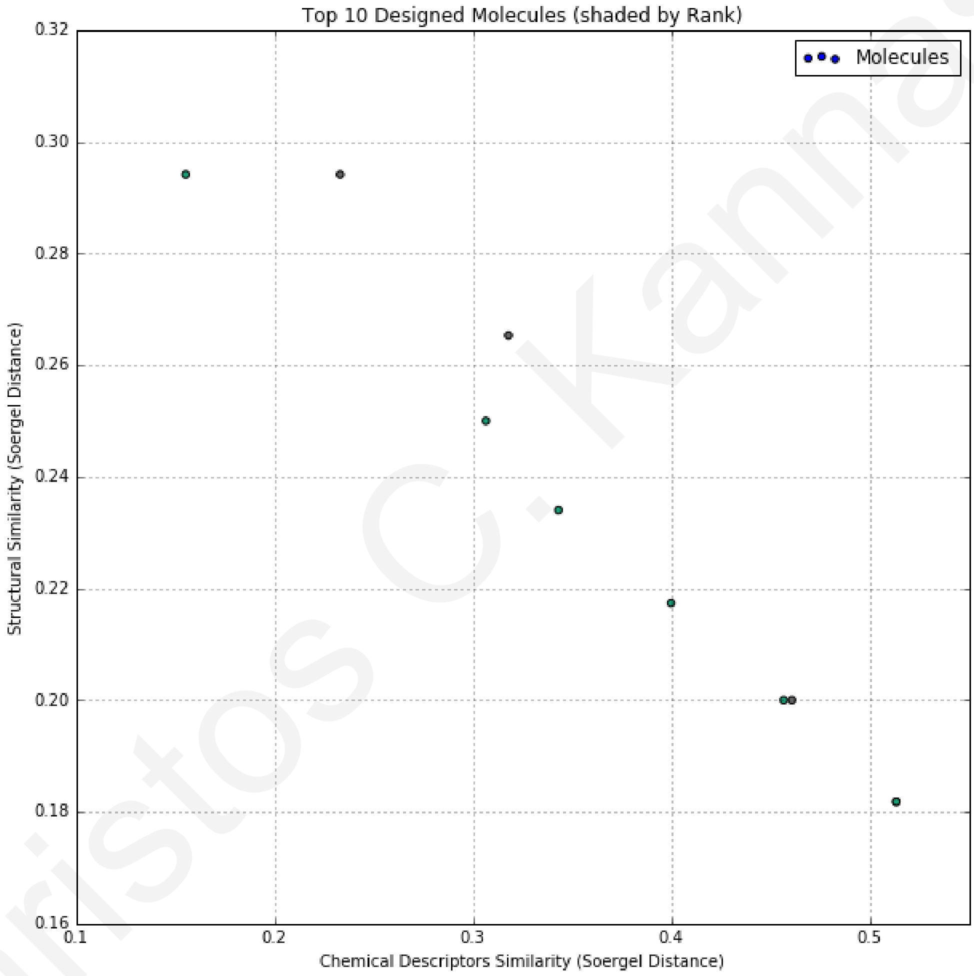


Figure 34: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) Top 10 results for Asinex dataset.

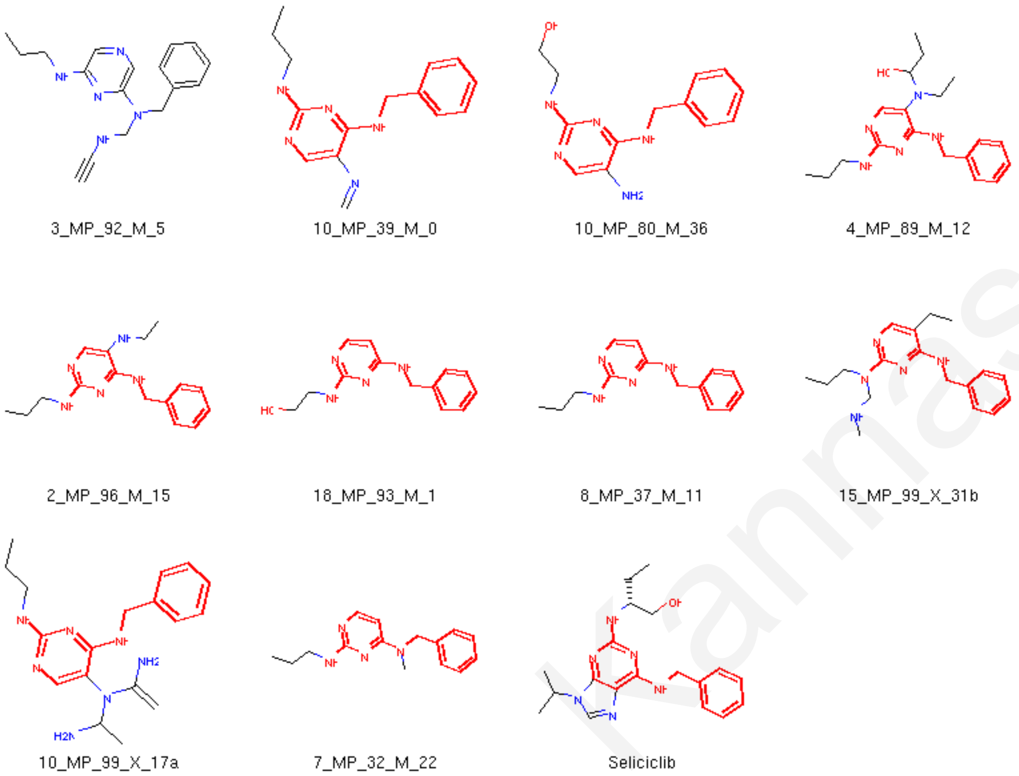
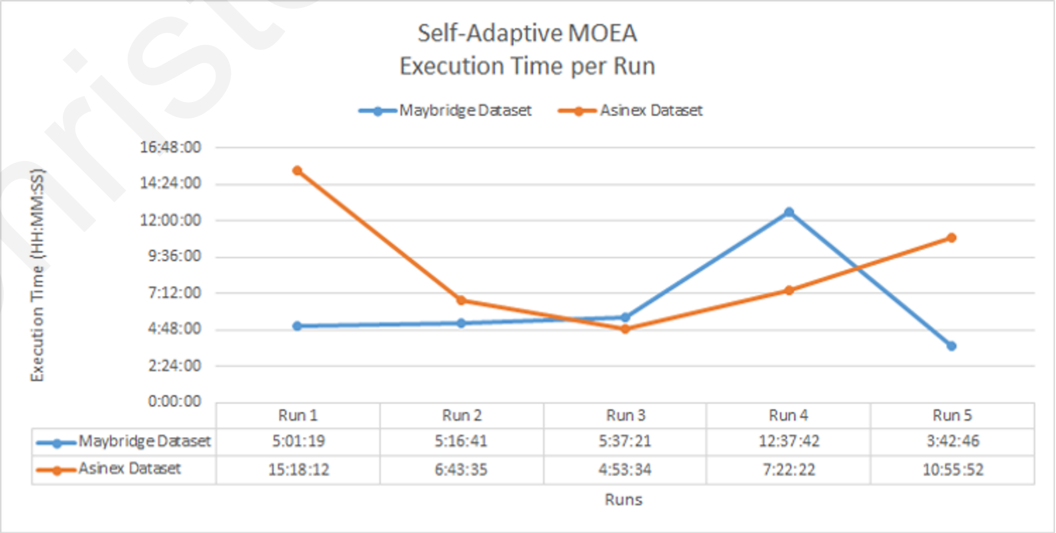


Figure 35: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) Top 10 results for Asinex dataset compared with Seliciclib, the red highlighted part of the molecules is their common core.



	Run 1	Run 2	Run 3	Run 4	Run 5
Maybridge Dataset	5:01:19	5:16:41	5:37:21	12:37:42	3:42:46
Asinex Dataset	15:18:12	6:43:35	4:53:34	7:22:22	10:55:52

Figure 36: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) execution time per run for Maybridge and Asinex datasets. Time is wall clock time.

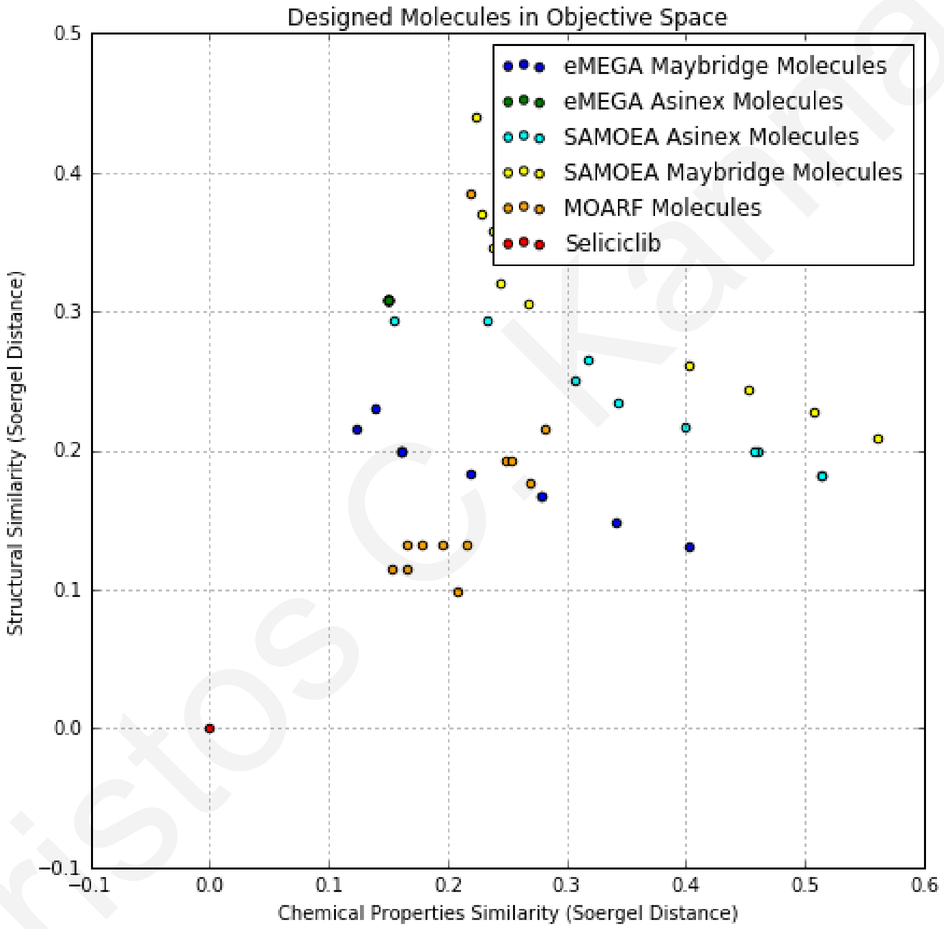


Figure 37: Compare all Top 10 results with Multi-Objective Algorithm for Replacement of Fragments (MOARF)'s results and Seliciclib.

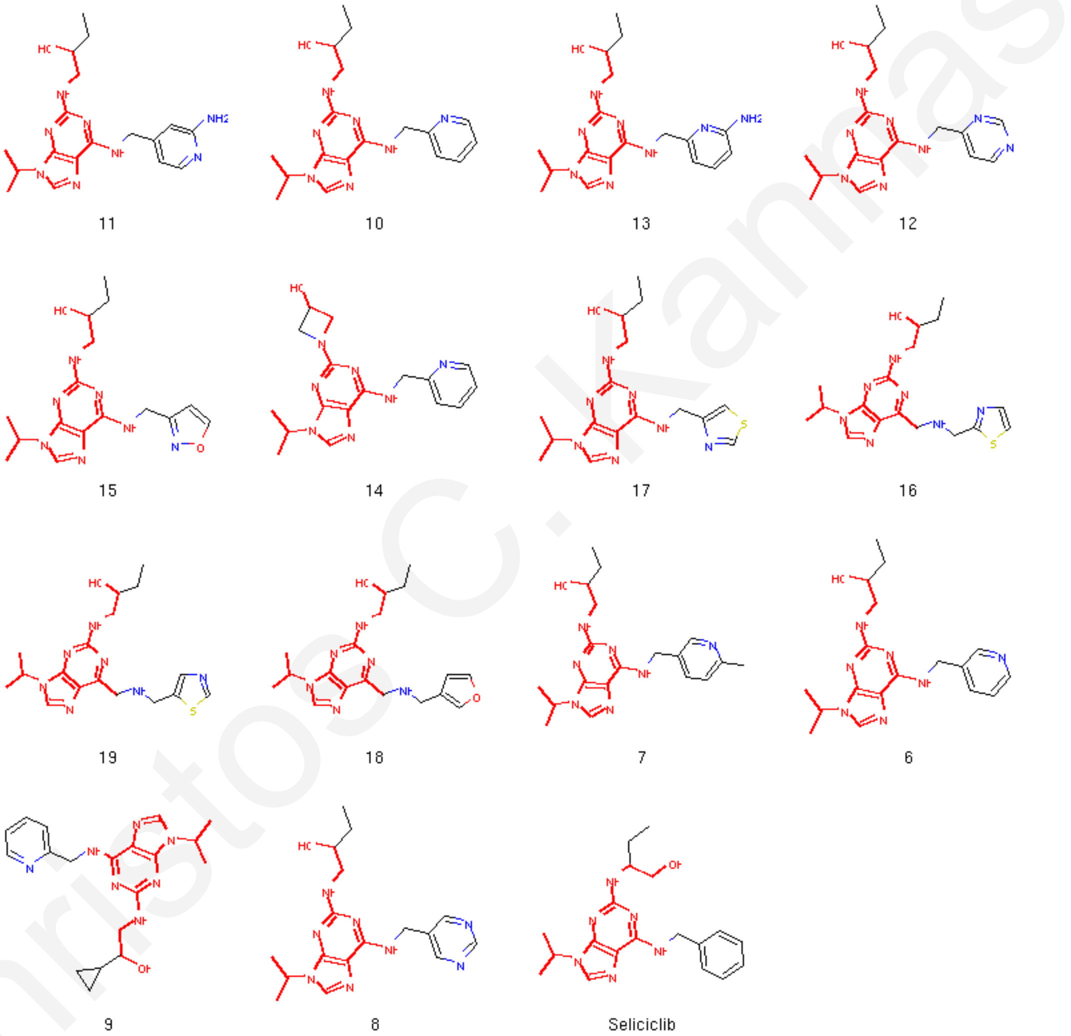


Figure 38: Compare Multi-Objective Algorithm for Replacement of Fragments (MOARF)'s results with Seliciclib.

6.2 Use Case 1: Design ER-α inhibitors based on similarity to Tamoxifen and similarity to Ibuproxam

6.2.1 Methodology

The objective of the experiment was to design molecules that have structural similarity to Tamoxifen[5] (Figure 39) and structural similarity to Ibuproxam[6] (Figure 40).

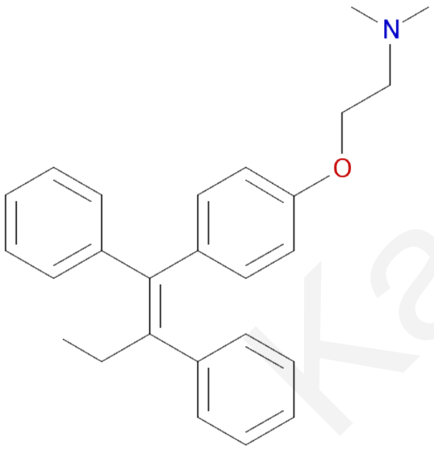


Figure 39: Tamoxifen.

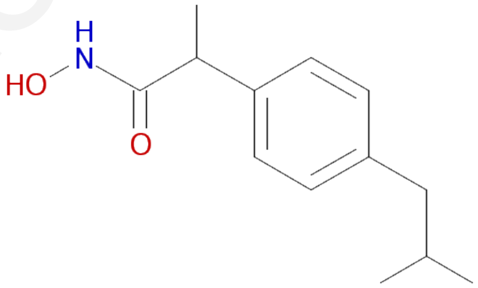


Figure 40: Ibuproxam.

The objectives for the molecular design algorithm were, structural similarity based on Soergel distance [272] to Tamoxifen Citerate (Figure 39), and structural similarity based on Soergel distance

[5] `https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL83`

[6] `https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL292707`

to Ibuproxam (Figure 40). The optimization process was aimed to minimize both of those objectives to 0.

Structural similarity is an objective fitness function that calculates a fitness score for a molecule to the target molecule, computed as the graph distance based on their MCS[7] , an in-house implementation based on RASCAL [269].

For input we used a collection of molecules retrieved from the latest version of ZINC database, ZINC15[8] . We selected molecules using the filters clean (Substances with ”clean” reactivity), in-vitro (Substances reported or inferred active at 10 uM or better in direct binding assays) and now (Immediate delivery, includes in-stock and agent). The collection contains 7035 molecules.

The Self-Adaptive MOEA works on a population size of 50 that are the settings for the second level eMEGAs. Self-Adaptive MOEA's chromosome is shown in Figure 20. Self-Adaptive MOEA operates with a mutation probability set to 15% while the crossover probability was set to 80%. Parent selection was set to roulette. For the elitist generation selection Self-Adaptive MOEA was set to use phenotype diversity. The second level eMEGAs operate on a population size of 250. eMEGAs operate on mutations probability in the range of 0 to 0.2 and crossover probabilities in the range of 0.8 to 1.0. Results were assessed after 500 iterations. A synopsis of Self-Adaptive MOEA settings can be found in Table 10.

Table 10: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) experimental design settings

Self-Adaptive MOEA				
Dataset	Objectives	Population	Iterations	Evolutionary Operations
ZINC15 clean, in-vitro, now	Non Dominate Solutions Percentage Pareto Front Hypervolume	50	500	Mutation Probability: **15%** Crossover Probability: **80%** Selection Type: **Roulette** Diversity Type: **Phenotype**
eMEGAs				
ZINC15 clean, in-vitro, now	Structural Similarity (Tamoxifen) Structural Similarity (Ibuproxam)	250	1	Defined during run time. Based on Self-Adaptive MOEA's chromosomes.

[7] https://en.wikipedia.org/wiki/Maximum_common_subgraph

[8] http://zinc15.docking.org/

The final solutions were filtered and we selected the ones with objective fitness score lower than 0.5 for each objective. As the objectives used, describe the distance to specific target molecules, we would like to have solutions that are closer (in case of similarity) from the defined target.

In order to validate the effectiveness of the proposed solutions, as a last step we perform a docking experiment to ER-α using the AutoDock Vina [232] tools provided by **Li**fe **S**ciences **I**nformatic**s** (LiSIs) [238]. The selected solutions were docked to ER-α 3ERT.

6.2.2 Results

In Figure 41 all the proposed solutions in objective space, are shown and in Figure 42 the non dominated proposed solutions in objective space, are shown. All solutions within the range of 0 to 0.5 for both objectives are considered good solutions, though we decided to use the non dominated solutions for the docking experiment just for sake of simplicity. The solutions fill the space in an arc between the range of 0 to 0.6 for both objectives. In total there are 29 non dominated solutions, represented by 22 unique points. As pointed above we are interested for the solutions between 0 and 0.5 range for both objectives. Which are the 10 non dominated solutions shown in Figure 43.

Table 11 shows the docking experiments results, sorted in ascending order at Docking Affinity column. Figures 44, 45, 46, 47, 48, 49, 50, 51, 52 and 53 show the selected molecules at their docked position to ER-α docking site.

Table 11: AutoDock Vina docking to ER-α results

Molecule Id	Docking Affinity (kcal/mol)
Tamoxifen Citrate	**-8.2**
DnD_6_SP_20_4_X_13a	-7.9
DnD_31_SP_150_37_M_19	-7.9
DnD_8_SP_9_2_M_13	-7.8
DnD_4_SP_199_49_X_46b	-7.7
DnD_12_SP_75_18_M_13	-7.6
DnD_31_SP_6_1_M_16	-7.2
DnD_15_SP_168_41_M_0	-7.2
Ibuproxam	**-7.2**
DnD_11_SP_74_18_M_4	-7.1
DnD_31_SP_193_48_X_76b	-6.9
DnD_1_SP_78_19_X_84a	-6.8

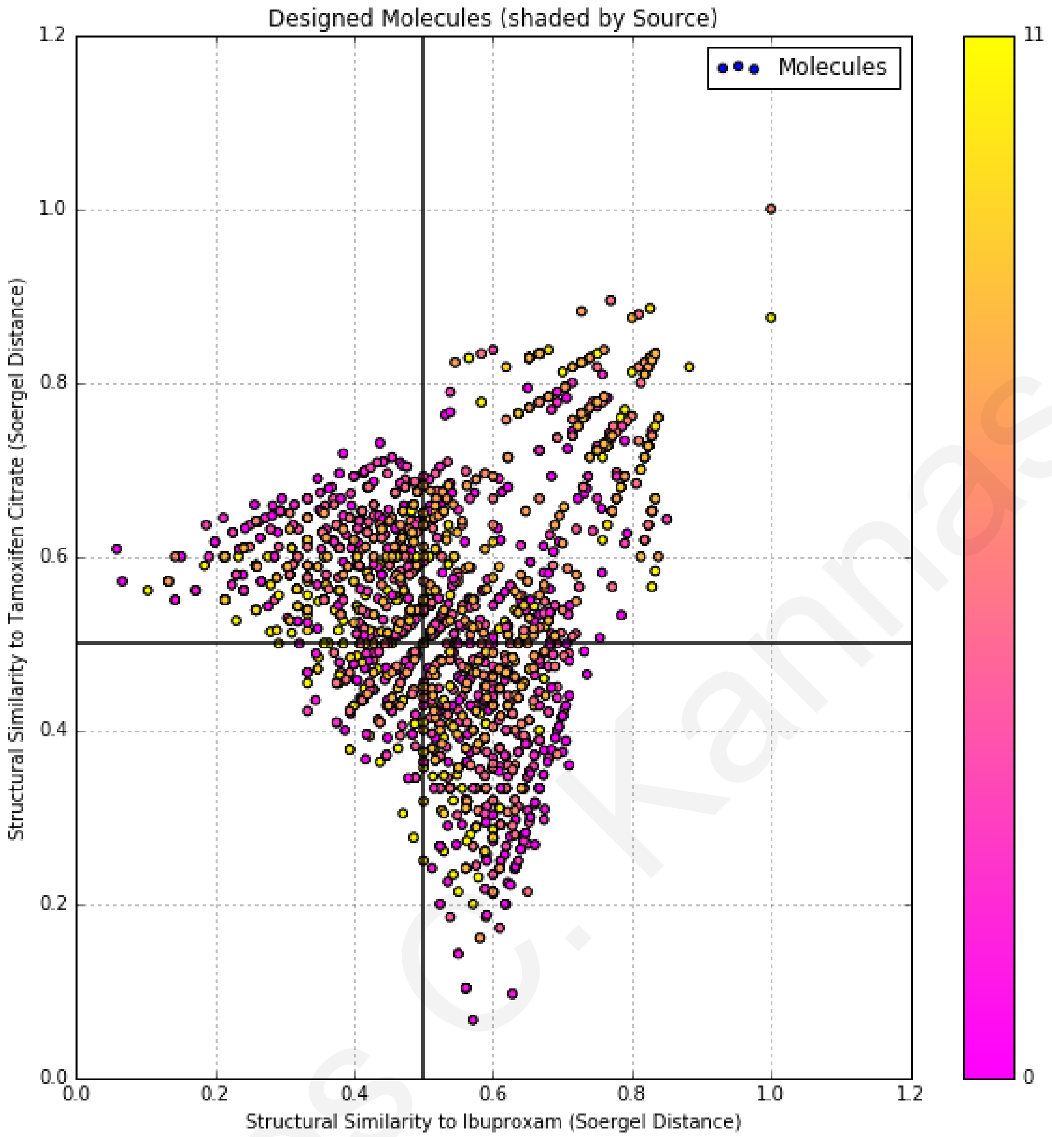


Figure 41: Designed molecules in objective space.

The secondary output of Self-Adaptive MOEA are the proposed settings for eMEGA for the problem. In Table 12 there are the non dominated proposed settings.

6.2.3 Discussion

As shown from the docking experiment results in Table 11, the selected solutions have good docking affinity to ER-α, which is between -6.8 and -7.9 kcal/mol. From the visualisation of the docking conformations of the solutions in Figures 44, 45, 46, 47, 48, 49, 50, 51, 52 and 53, we see that the solutions fit well in the docking site of the protein. The proposed molecules are small in size, with two exceptions (Figures 51 and 53).

From Table 12 we understand that the preferred eMEGA parameter settings are:

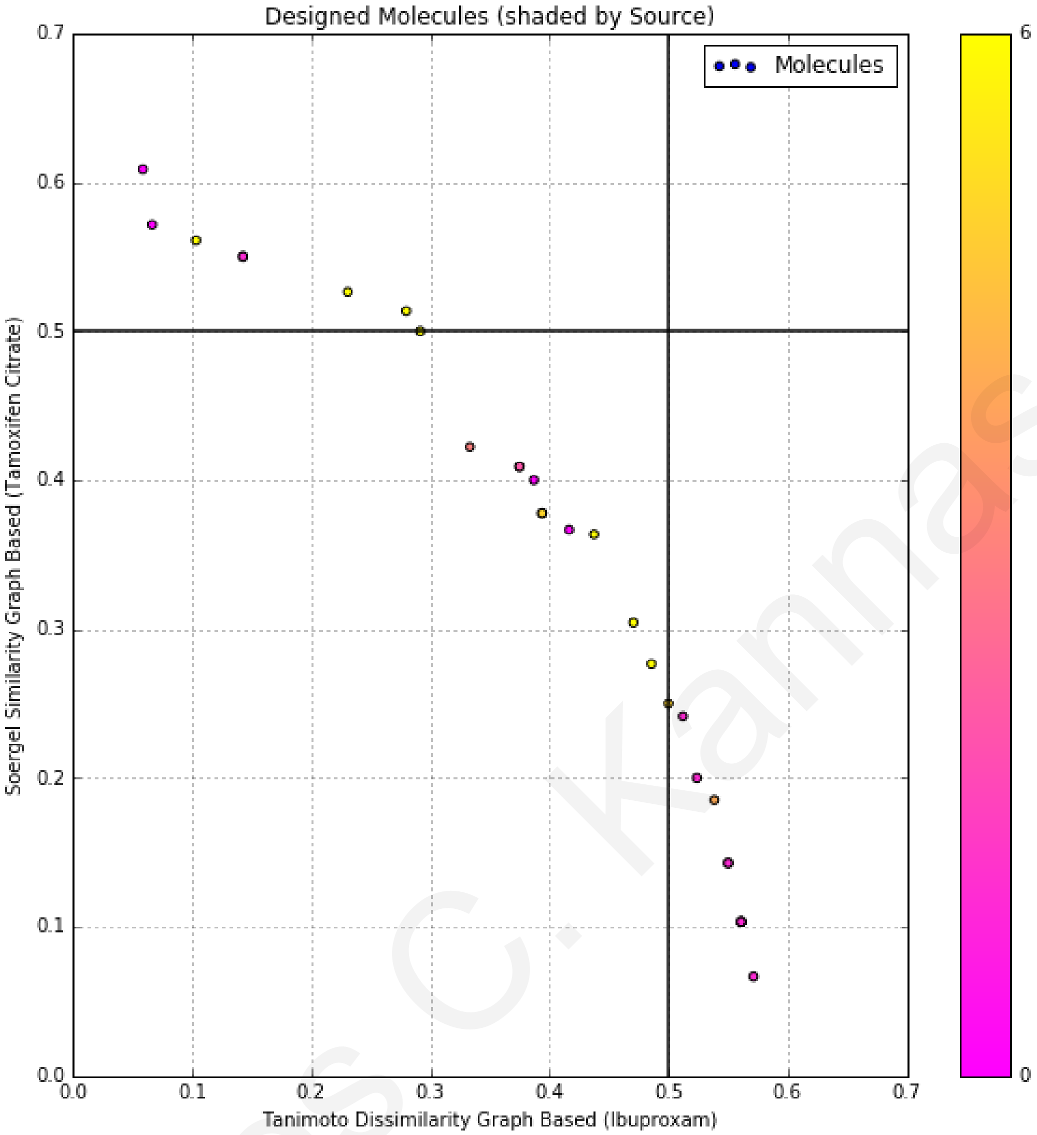


Figure 42: Non dominated designed molecules in objective space.

- *Mutation Probability* of *15 - 16%*,
- *Crossover Probability* of *88%*,
- *Parent Selection* based on *Tournament Selection*, and
- *Next Generation (Population) Selection* based on *Genotype Diversity*.

This experiment required 106 hours (106:16:43) to complete the 500 iterations. The execution was split up into 10 batches of 50 iterations and on average each batch required on average 9 hours and 19 minutes.

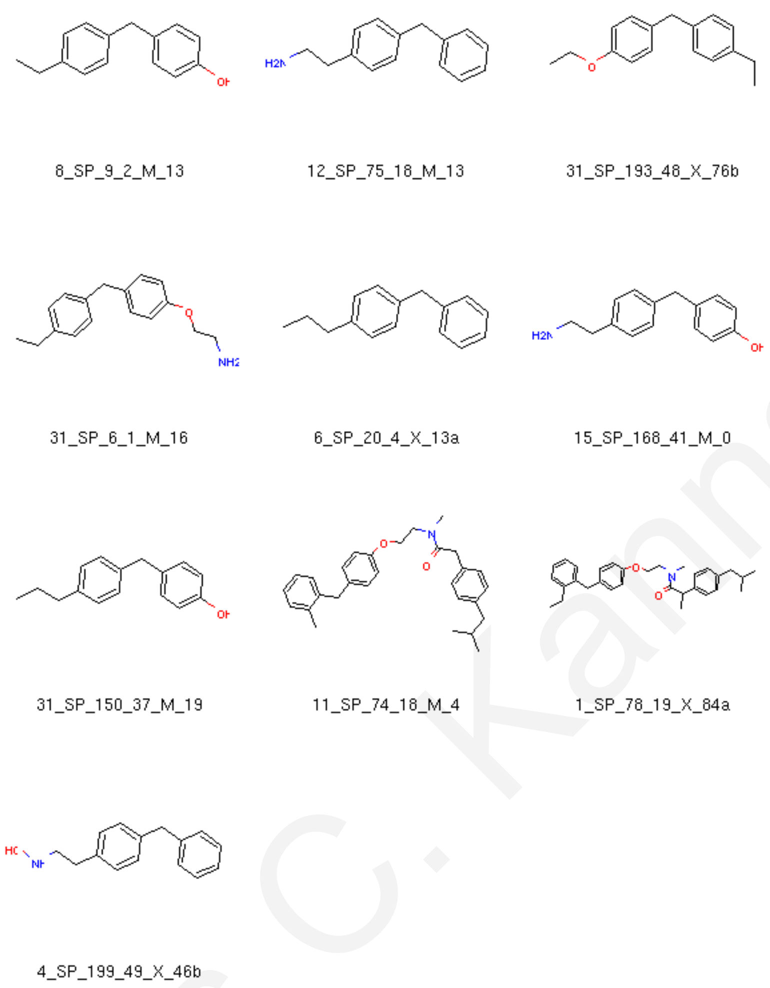


Figure 43: Designed molecules 2D depictions.

From discovery informatics point of view the selected designed compounds look promising. Further investigation is required to investigate the behaviour of the protein-ligand complex, which requires *in-vitro* experiments.

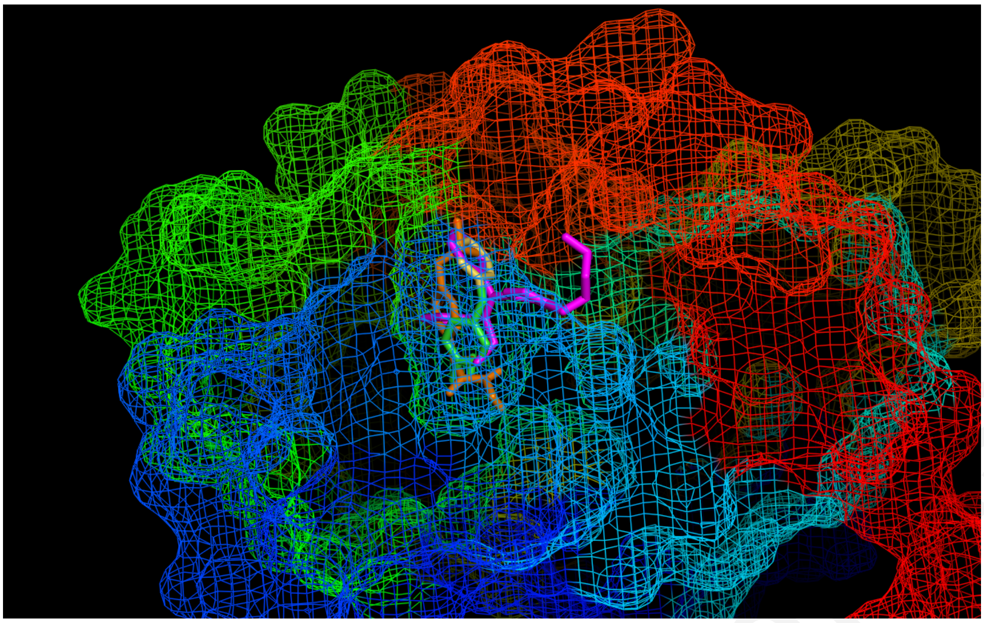


Figure 44: Designed molecule DnD_6_SP_20_4_X_13a docked to ER-α, in reference with Tamoxifen Citrate (magenta) and Ibuproxam (orange).

Table 12: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) non dominated settings for elite Multi-Objective Evolutionary Graph Algorithm (eMEGA)

Mutation Probability	Crossover Probability	Selection Type	Diversity Type	Non Dominated %	Pareto Hypervolume	Rank
0.16	0.80	tournament	genotype	0.63	0.34	1
0.16	0.88	tournament	genotype	0.63	0.34	1
0.16	0.89	tournament	genotype	0.63	0.34	1
0.16	0.89	roulette	genotype	0.65	0.34	1
0.01	0.94	best	genotype	0.62	0.43	1

Note: The numbers for 'Non Dominated %' are 1 minus the actual %. The smaller the number listed here the better. 'Rank' is their non dominance rank.

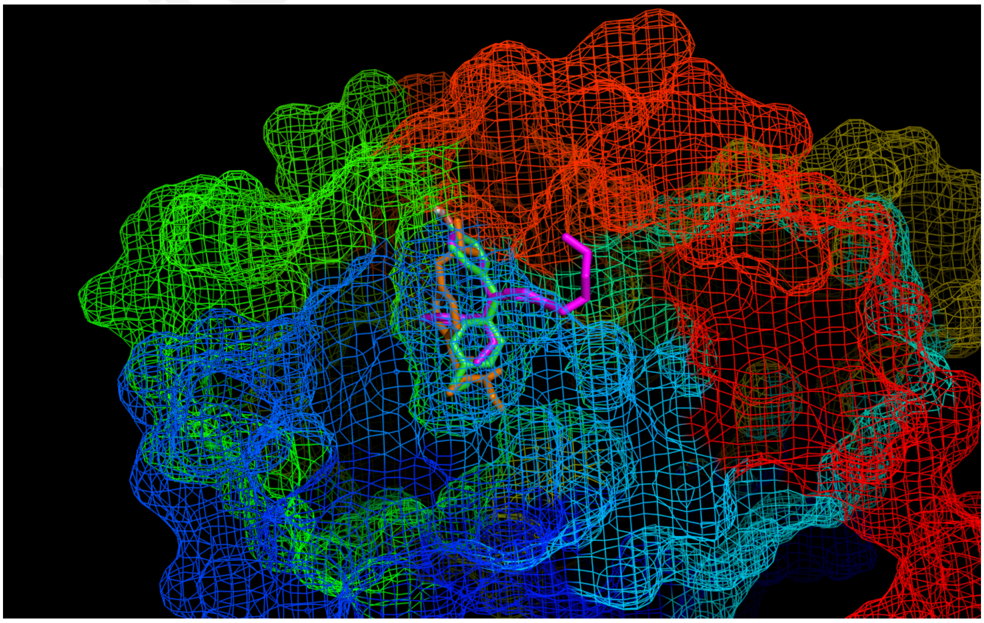


Figure 45: Designed molecule DnD_31_SP_150_37_M_19 docked to ER-α, in reference with Tamoxifen Citrate (magenta) and Ibuproxam (orange).

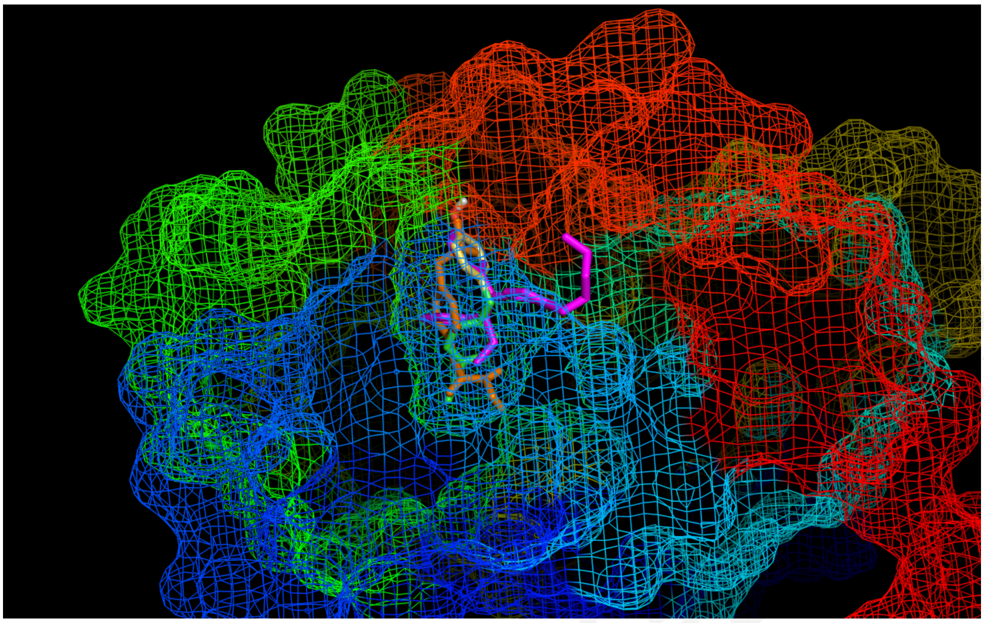


Figure 46: Designed molecule DnD_8_SP_9_2_M_13 docked to ER-α, in reference with Tamoxifen Citrate (magenta) and Ibuproxam (orange).

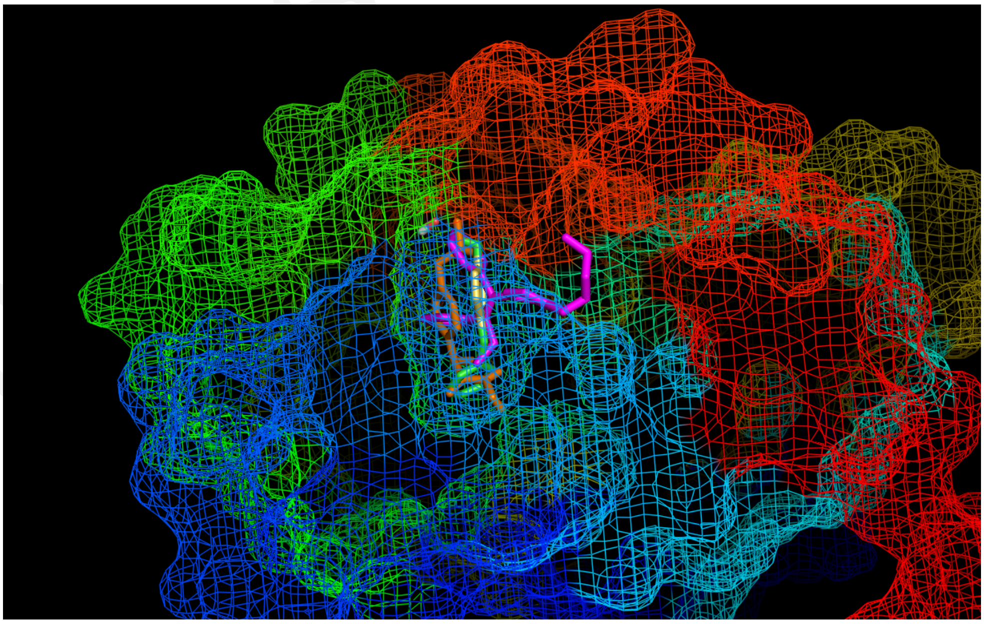


Figure 47: Designed molecule DnD_4_SP_199_49_X_46b docked to ER-α, in reference with Tamoxifen Citrate (magenta) and Ibuproxam (orange).

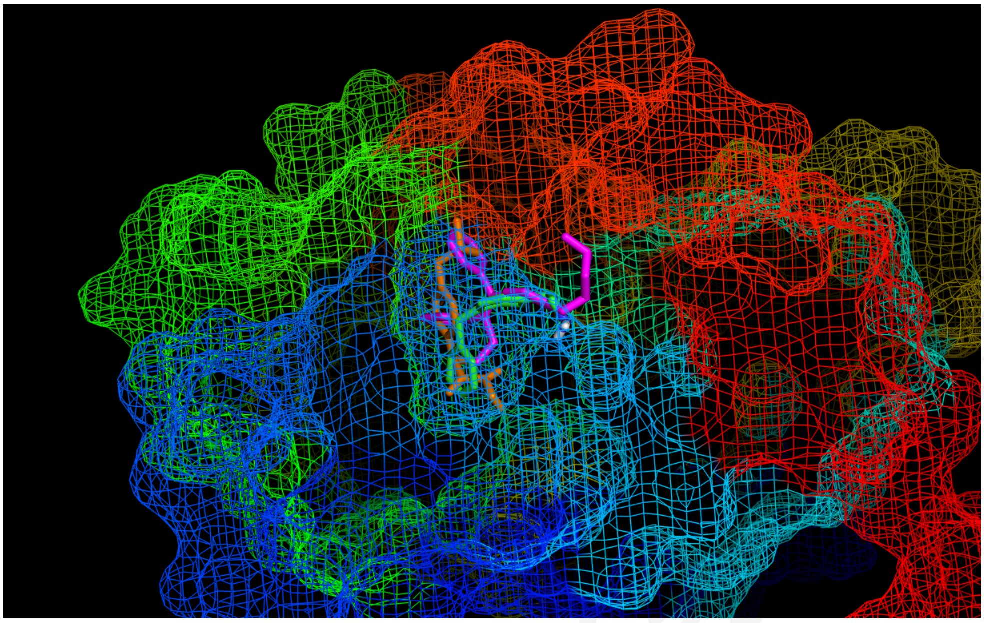


Figure 48: Designed molecule DnD_12_SP_75_18_M_13 docked to ER-α, in reference with Tamoxifen Citrate (magenta) and Ibuproxam (orange).

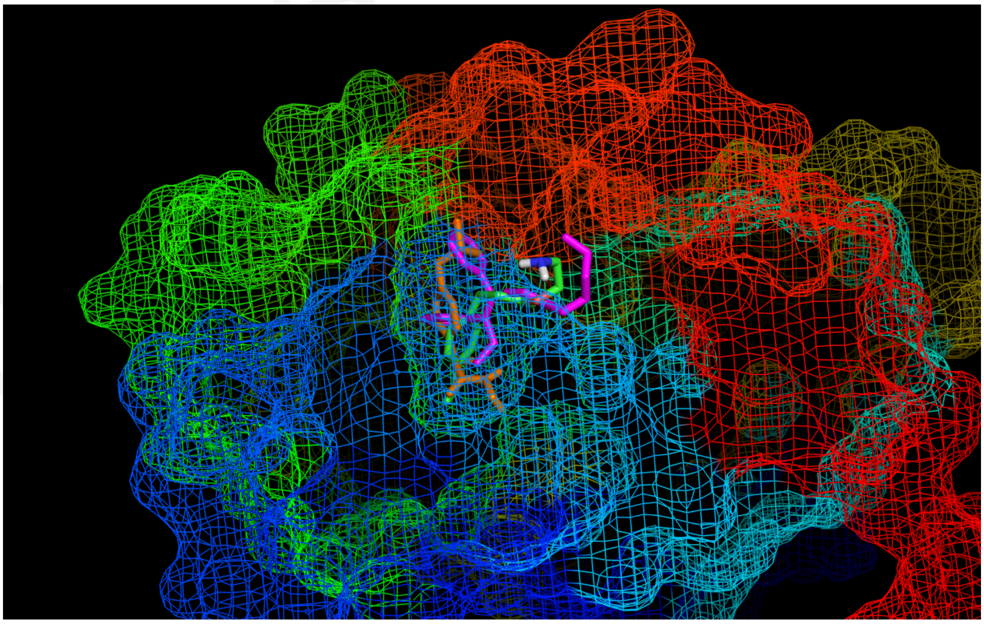


Figure 49: Designed molecule DnD_31_SP_6_1_M_16 docked to ER-α, in reference with Tamoxifen Citrate (magenta) and Ibuproxam (orange).

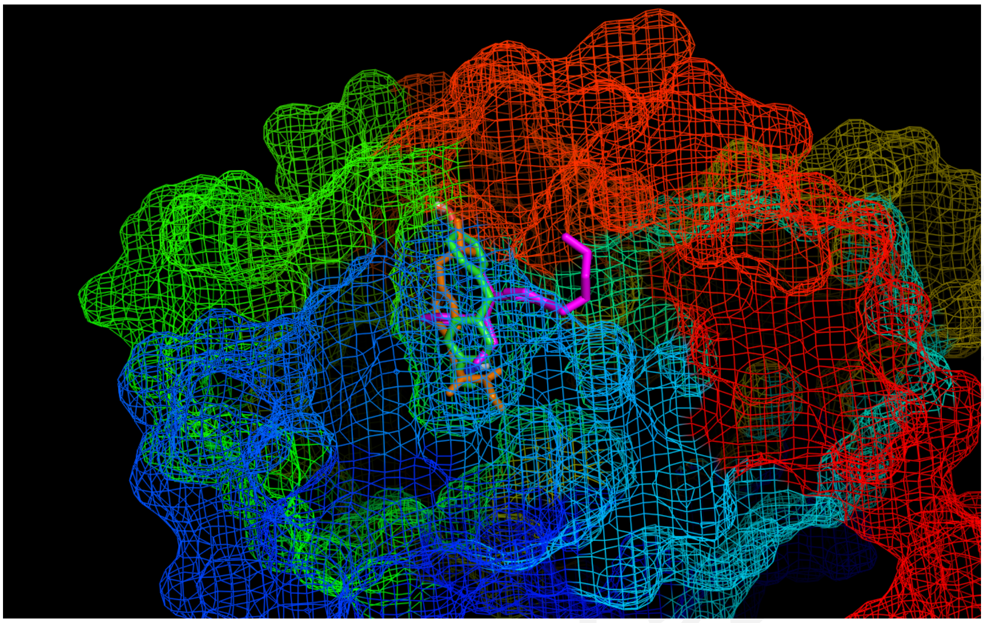


Figure 50: Designed molecule DnD_15_SP_168_41_M_0 docked to ER-α, in reference with Tamoxifen Citrate (magenta) and Ibuprofen (orange).

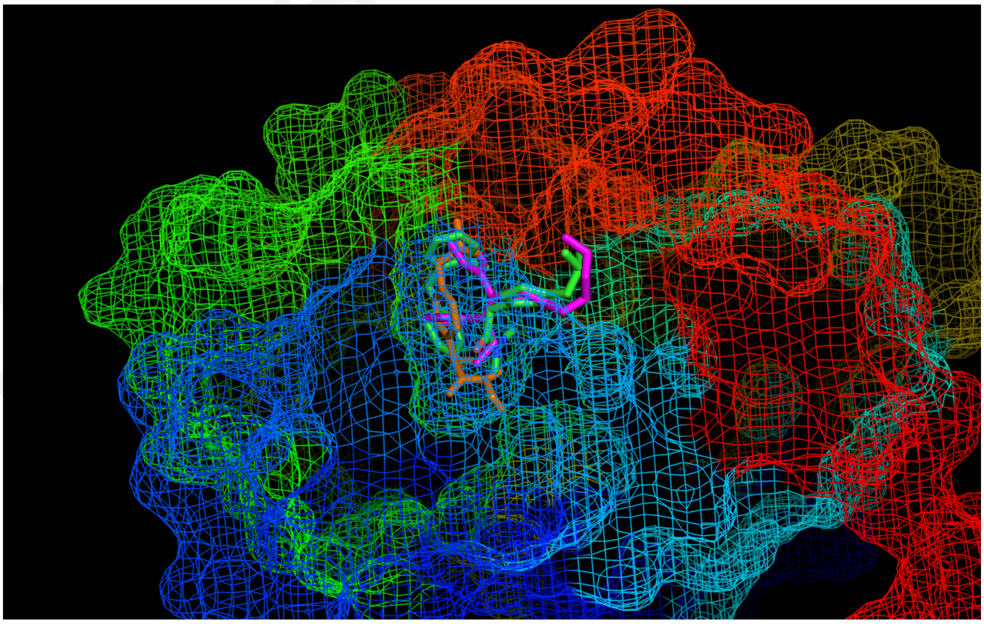


Figure 51: Designed molecule DnD_11_SP_74_18_M_4 docked to ER-α, in reference with Tamoxifen Citrate (magenta) and Ibuprofen (orange).

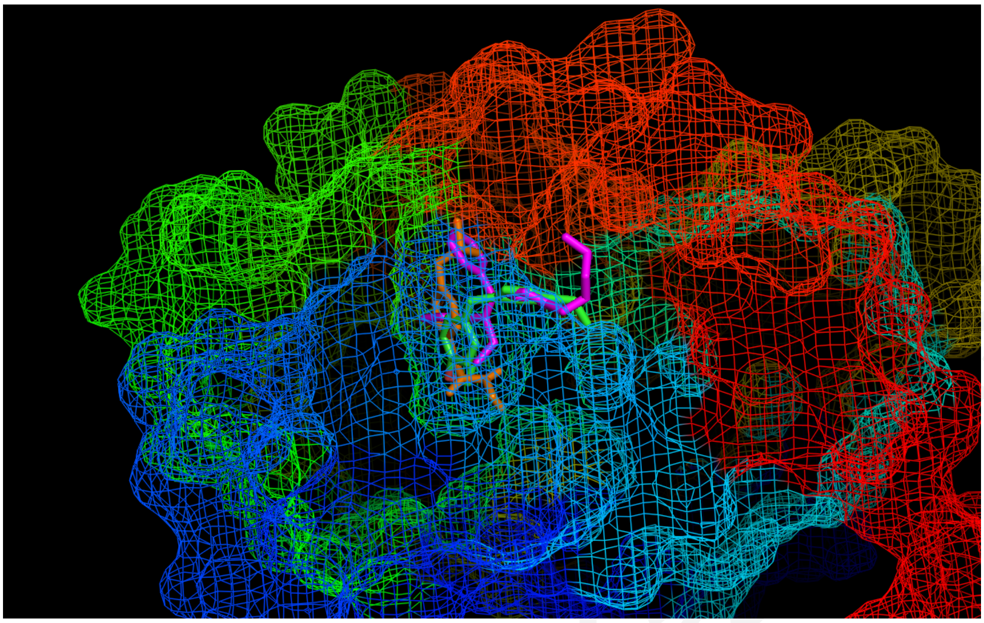


Figure 52: Designed molecule DnD_31_SP_193_48_X_76b docked to ER-α, in reference with Tamoxifen Citrate (magenta) and Ibuproxam (orange).

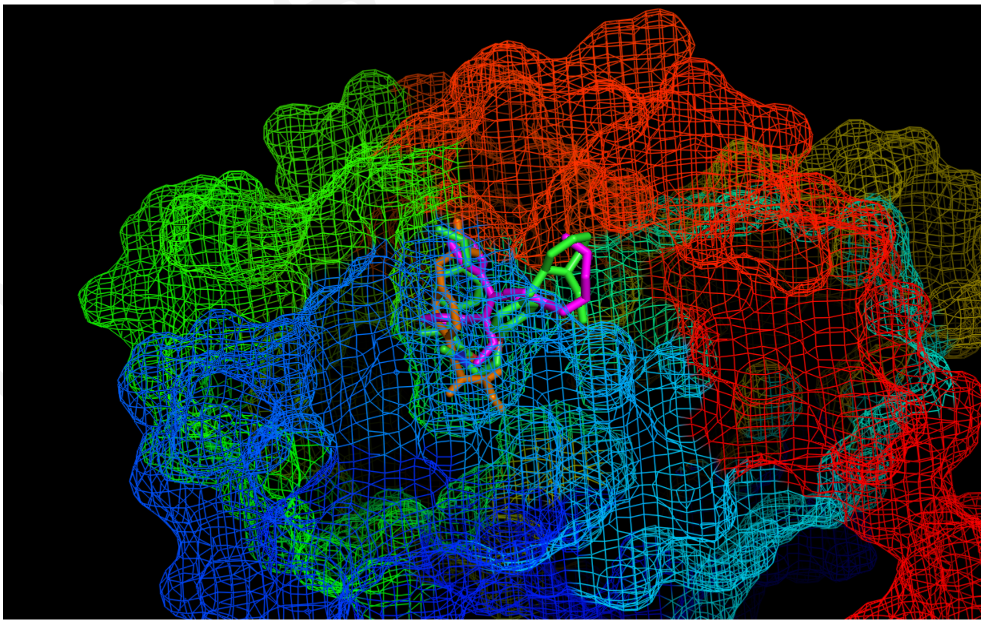


Figure 53: Designed molecule DnD_1_SP_78_19_X_84a docked to ER-α, in reference with Tamoxifen Citrate (magenta) and Ibuproxam (orange).

6.3 Use Case 2: Design ER-α inhibitors based on similarity to Tamoxifen

6.3.1 Methodology

The objective of the experiment was to design molecules that have structural and chemical descriptors similarity to Tamoxifen[9] (Figure 39).

The objectives for the molecular design algorithm were, structural similarity based on Soergel distance [272] and chemical descriptors similarity based on Euclidean distance [272] to Tamoxifen Citerate (Figure 39). The optimization process was aimed to minimize both of those objectives to 0.

Structural similarity is an objective fitness function that calculates a fitness score for a molecule to the target molecule, computed as the graph distance based on their MCS[10] , an in-house implementation based on RASCAL [269].

Chemical descriptors similarity is an objective fitness function that calculates a fitness score for a molecule to the target molecule, computed as the distance of their chemical descriptors vector. The chemical descriptors are based on an in-house implementation of atom pairs and topological torsions calculations for each molecule [270] and [271].

For input we used a collection of molecules retrieved from the latest version of ZINC database, ZINC15[11] . We selected molecules using the filters clean (Substances with ”clean” reactivity), in-vitro (Substances reported or inferred active at 10 uM or better in direct binding assays) and now (Immediate delivery, includes in-stock and agent). The collection contains 7035 molecules.

The Self-Adaptive MOEA works on a population size of 50 that are the settings for the second level eMEGAs. Self-Adaptive MOEA’s chromosome is shown in Figure 20. Self-Adaptive MOEA operates with a mutation probability set to 15% while the crossover probability was set to 80%. Parent selection was set to roulette. For the elitist generation selection Self-Adaptive MOEA was set to use phenotype diversity. The second level eMEGAs operate on a population size of 250. eMEGAs operate on mutations probability in the range of 0 to 0.2 and crossover probabilities in the range of

[9] https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL83

[10] https://en.wikipedia.org/wiki/Maximum_common_subgraph

[11] http://zinc15.docking.org/

0.8 to 1.0. Results were assessed after 100 iterations. A synopsis of Self-Adaptive MOEA settings can be found in Table 13.

Table 13: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) experimental design settings

Self-Adaptive MOEA				
Dataset	Objectives	Population	Iterations	Evolutionary Operations
ZINC15 clean, in-vitro, now	Non Dominated Solutions Percentage Pareto Front Hypervolume	50	100	Mutation Probability: **15%** Crossover Probability: **80%** Selection Type: **Roulette** Diversity Type: **Phenotype**
eMEGAs				
ZINC15 clean, in-vitro, now	Structural Similarity Chemical Descriptors Similarity	250	1	Defined during run time. Based on Self-Adaptive MOEA's chromosomes.

The final solutions were filtered and we selected the ones with objective fitness score lower than 0.5 for each objective. As the objectives used, describe the distance to specific target molecules, we would like to have solutions that are closer (in case of similarity) from the defined target.

In order to validate the effectiveness of the proposed solutions, as a last step we perform a docking experiment to ER-α using the AutoDock Vina [232] tools provided by LiSIs [238]. The selected solutions were docked to ER-α 3ERT.

6.3.2 Results

In Figure 54 all the proposed solutions in objective space, are shown and in Figure 55 the non dominated proposed solutions in objective space, are shown. All solutions within the range of 0 to 0.5 for both objectives are considered good solutions, though we decided to use the non dominated solutions for the docking experiment just for sake of simplicity. The solutions fill the space in an arc between the range of 0 to 0.1 in Y-axis and 0.1 to 0.2 in X-axis. In total there are 4 non dominated solutions, represented by 3 unique points. As pointed above we are interested for the solutions between 0 and 0.5 range for both objectives. Which are the 4 non dominated solutions shown in Figure 56.

Table 14 shows the docking experiments results, sorted in ascending order at Docking Affinity column. Figures 57, 58, 59 and 60 show the selected molecules at their docked position to ER-α docking site.

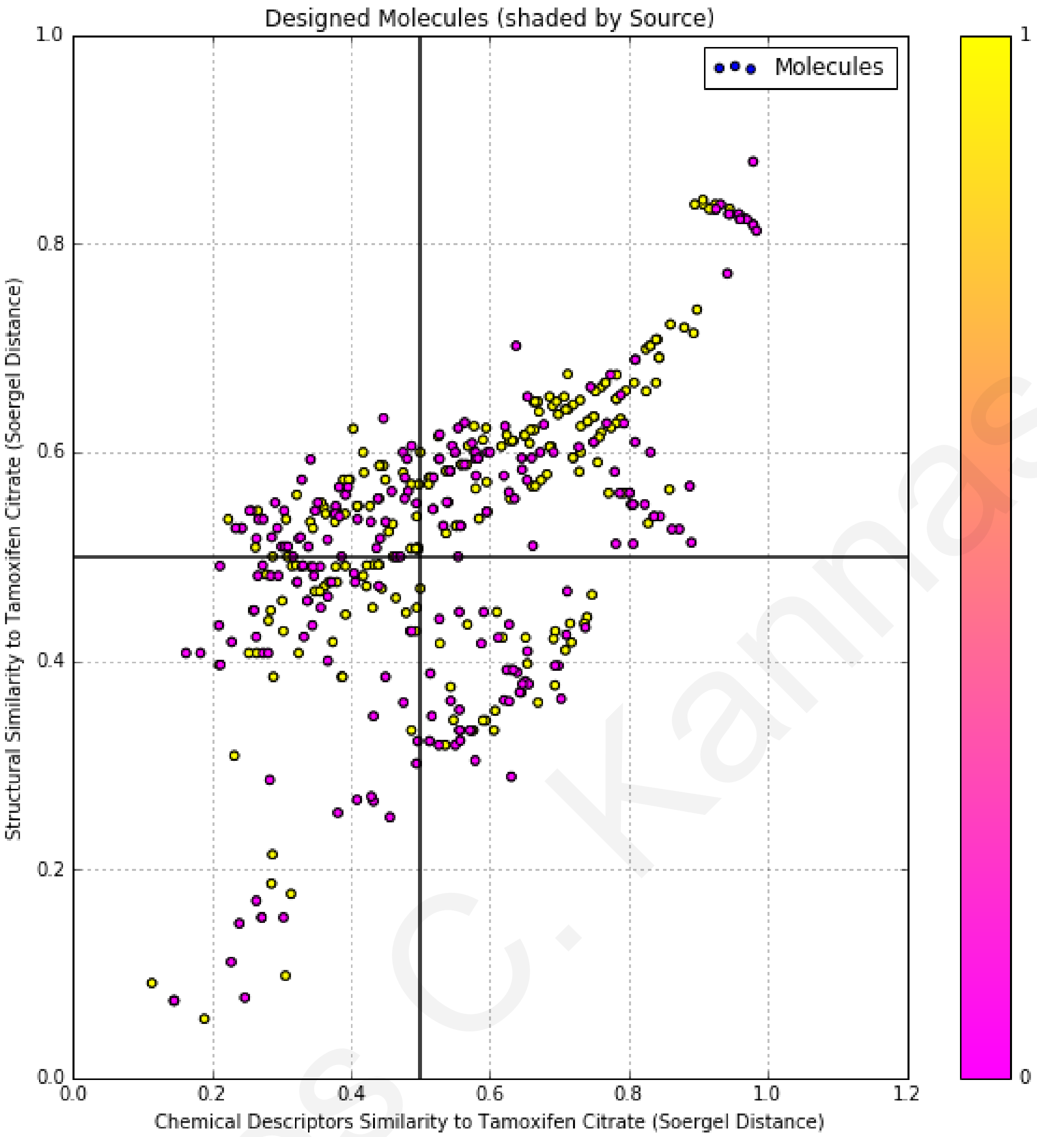


Figure 54: Designed molecules in objective space.

The secondary output of Self-Adaptive MOEA are the proposed settings for eMEGA for the problem. In Table 15 there are the non dominated proposed settings.

6.3.3 Discussion

As shown from the docking experiment results in Table 14, the selected solutions have very good docking affinity to ER-α, which is between -9.6 and -10.1 kcal/mol. From the visualisation of the docking conformations of the solutions in Figures 57, 58, 59 and 60, we see that the solutions fit well in the docking site of the protein. The proposed molecules are small in size.

From Table 15 we understand that the preferred eMEGA parameter settings are:

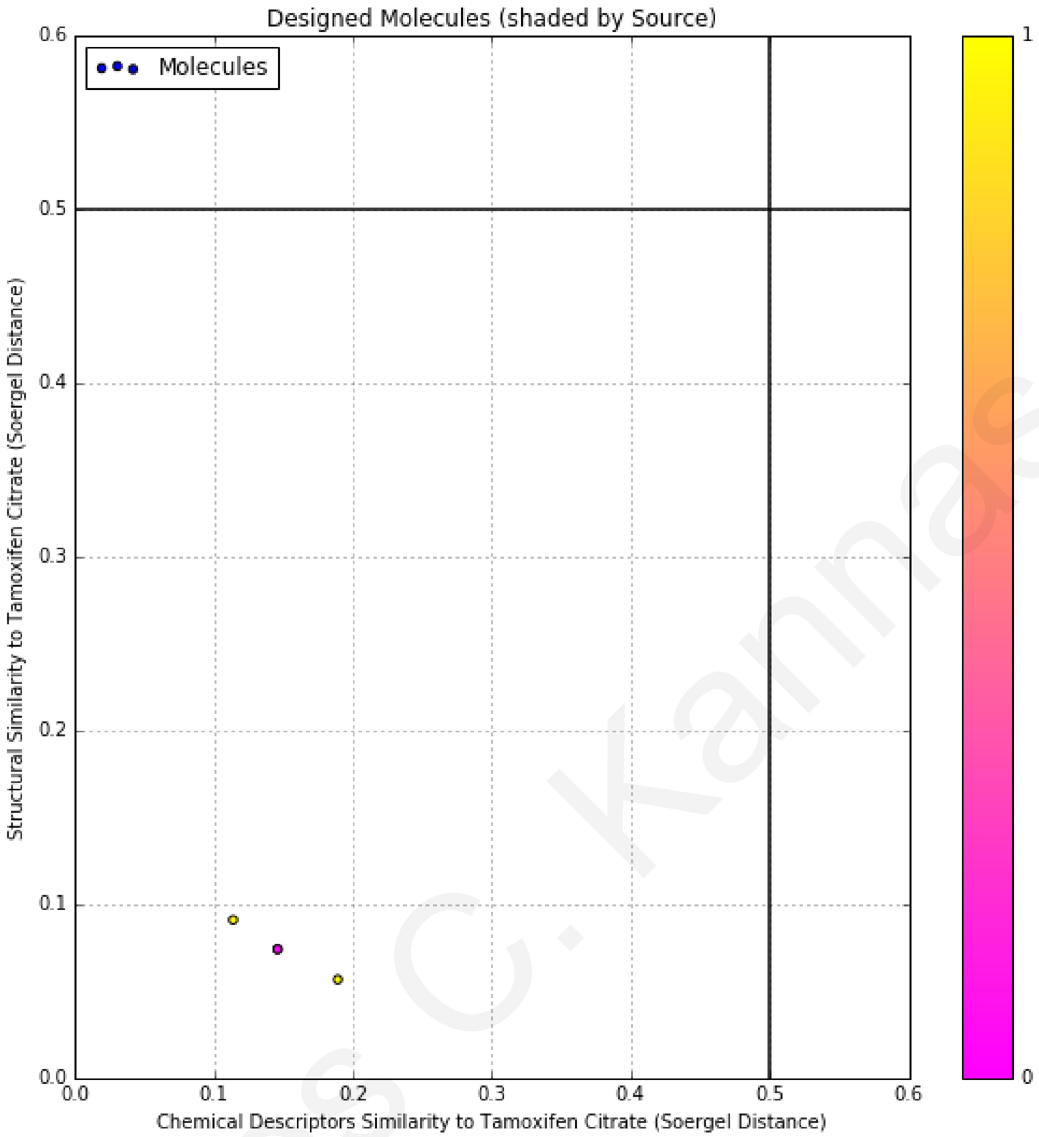


Figure 55: Non dominated designed molecules in objective space.

- *Mutation Probability* of *3%*,
- *Crossover Probability* of *98%*,
- *Parent Selection* based on *Tournament Selection*, and
- *Next Generation (Population) Selection* based on *Genotype Diversity* or *Phenotype Diversity*.

This experiment required 29 hours (29:35:36) to complete the 100 iterations. The execution was split up into 2 batches of 50 iterations and on average each batch required on average 14 hours and 48 minutes.

Table 14: AutoDock Vina docking to ER-α results

Molecule Id	Docking Affinity (kcal/mol)
Tamoxifen Citrate	**-8.2**
DnD_42_SP_194_48_X_96b	-10.1
DnD_17_SP_199_49_M_4	-10
DnD_33_SP_189_47_X_66b	-9.9
DnD_48_SP_193_48_M_5	-9.6

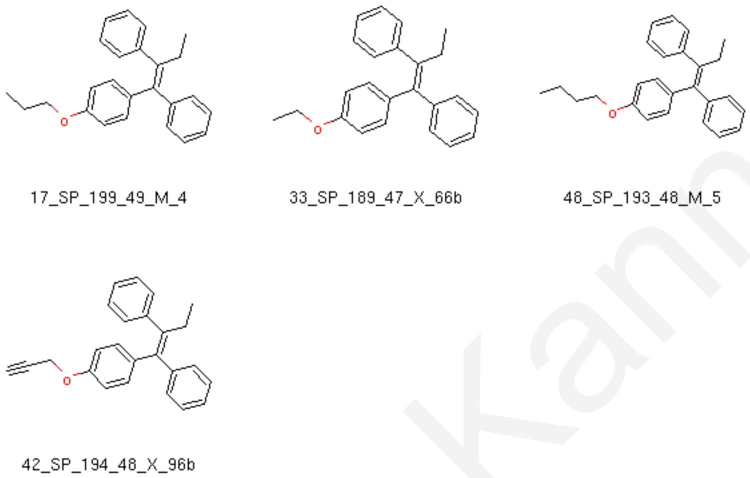


Figure 56: Designed molecules 2D depictions.

From discovery informatics point of view the selected designed compounds look promising. Further investigation is required to investigate the behaviour of the protein-ligand complex, which requires *in-vitro* experiments.

Table 15: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) non dominated settings for elite Multi-Objective Evolutionary Graph Algorithm (eMEGA)

Mutation Probability	Crossover Probability	Selection Type	Diversity Type	Non Dominated %	Pareto Hypervolume	Rank
0.03	0.98	tournament	genotype	0.98	0.15	1
0.03	0.98	tournament	phenotype	0.98	0.15	1

Note: The numbers for 'Non Dominated %' are 1 minus the actual %. The smaller the number listed here the better. 'Rank' is their non dominance rank.

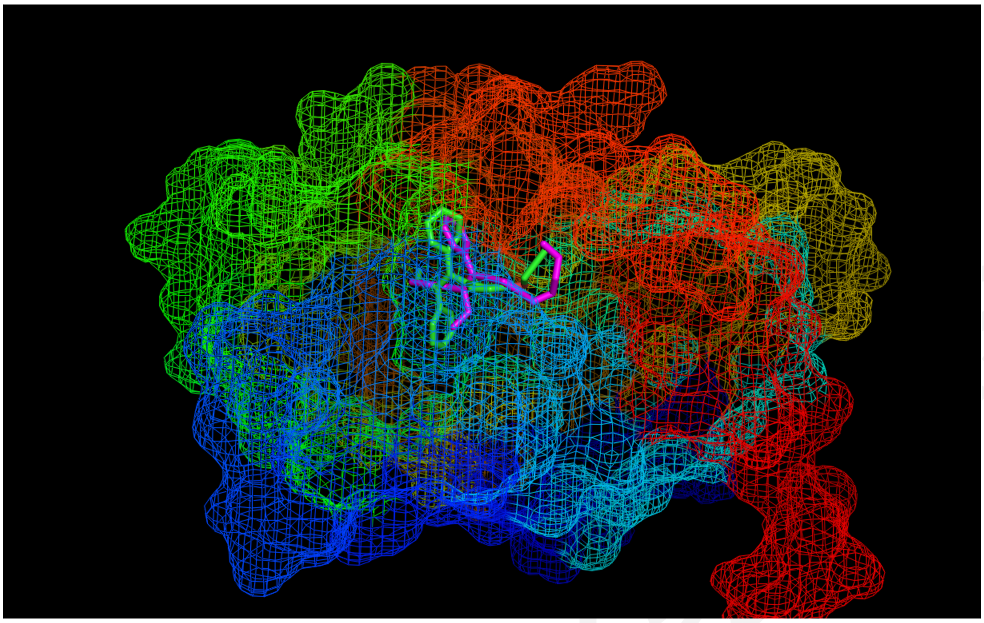


Figure 57: Designed molecule DnD_42_SP_194_48_X_96b docked to ER-α, in reference with Tamoxifen Citrate (magenta).

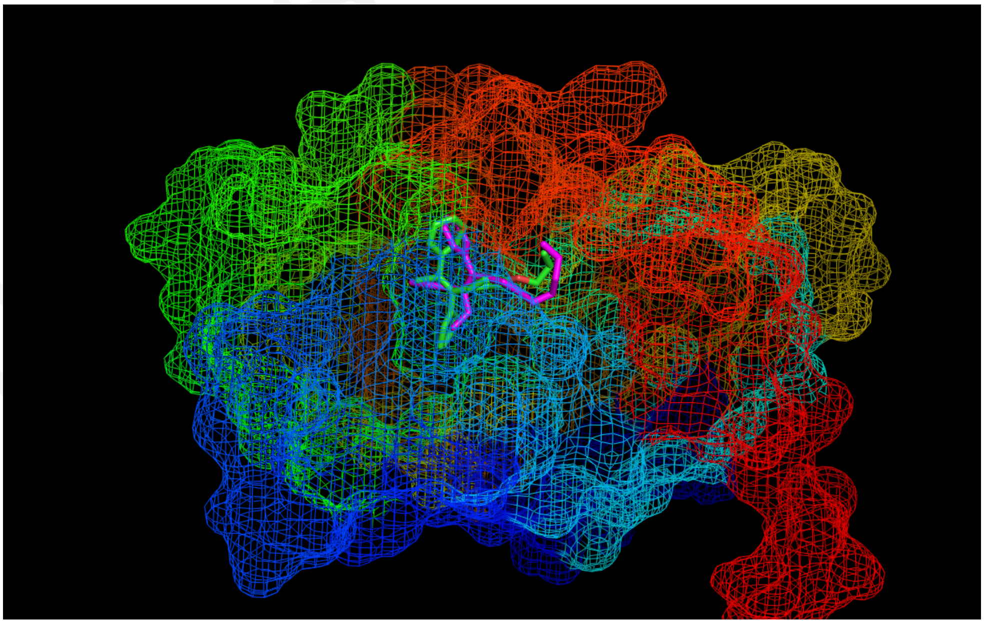


Figure 58: Designed molecule DnD_17_SP_199_49_M_4 docked to ER-α, in reference with Tamoxifen Citrate (magenta).

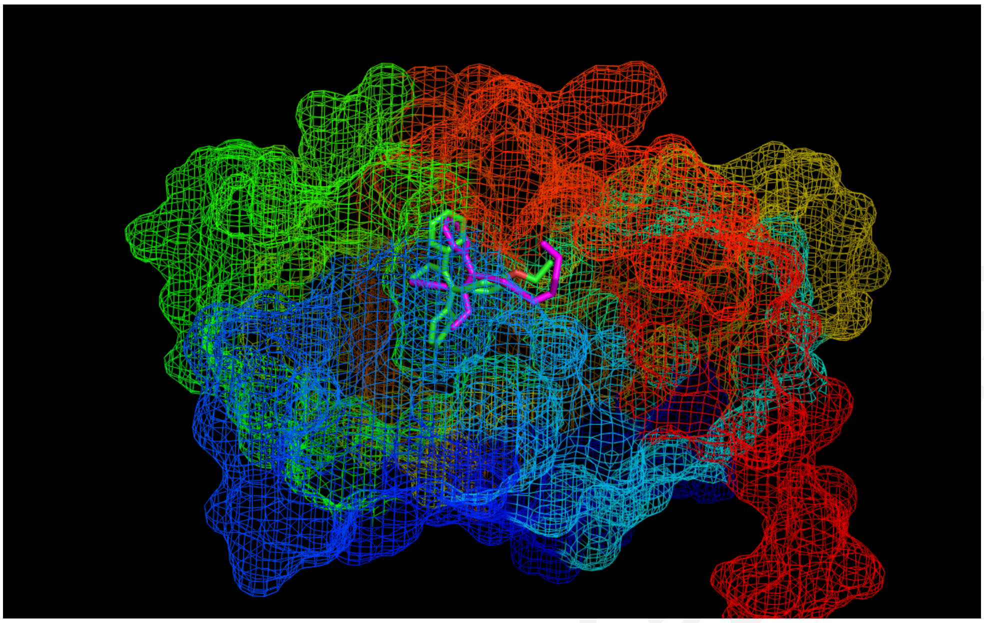


Figure 59: Designed molecule DnD_33_SP_189_47_X_66b docked to ER-α, in reference with Tamoxifen Citrate (magenta).

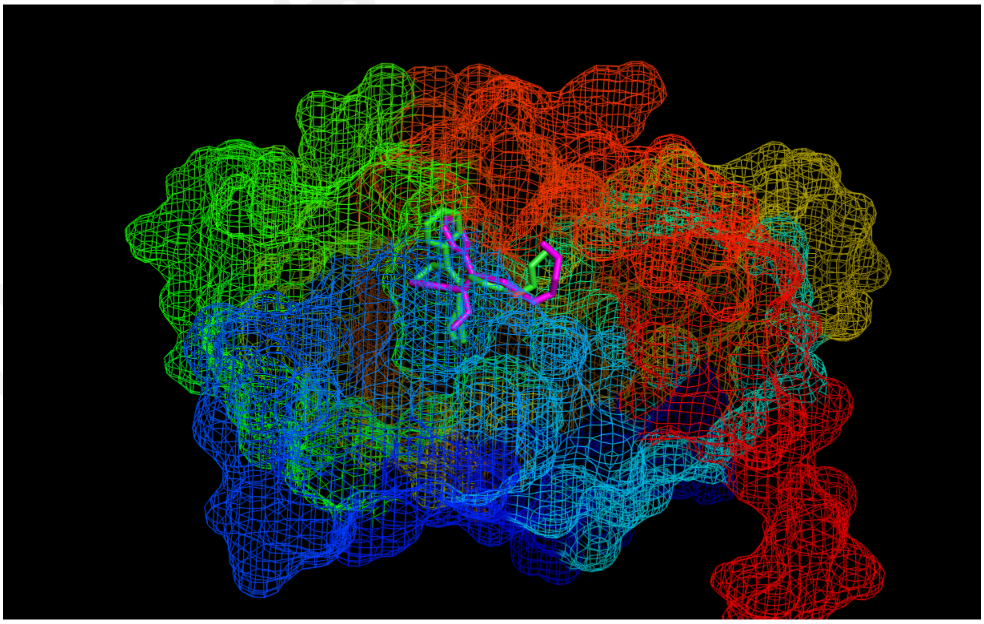


Figure 60: Designed molecule DnD_48_SP_193_48_M_5 docked to ER-α, in reference with Tamoxifen Citrate (magenta).

6.4 Use Case 3: Design ER-α inhibitors based on similarity to Raloxifene

6.4.1 Methodology

The objective of the experiment was to design molecules that have structural and chemical descriptors similarity to Raloxifene[12] (Figure 61).

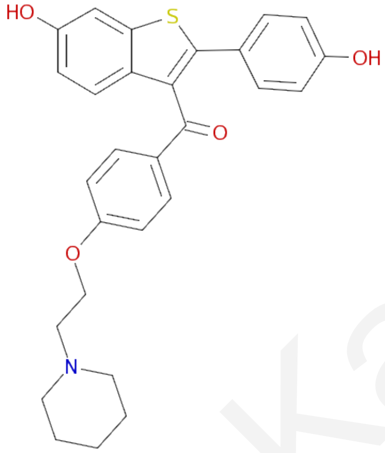


Figure 61: Raloxifene.

The objectives for the molecular design algorithm were, structural similarity based on Soergel distance [272] and chemical descriptors similarity based on Euclidean distance [272] to Raloxifene (Figure 61). The optimization process was aimed to minimize both of those objectives to 0.

Structural similarity is an objective fitness function that calculates a fitness score for a molecule to the target molecule, computed as the graph distance based on their MCS[13] , an in-house implementation based on RASCAL [269].

Chemical descriptors similarity is an objective fitness function that calculates a fitness score for a molecule to the target molecule, computed as the distance of their chemical descriptors vector. The chemical descriptors are based on an in-house implementation of atom pairs and topological torsions calculations for each molecule [270] and [271].

[12] https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL81

[13] https://en.wikipedia.org/wiki/Maximum_common_subgraph

For input we used a collection of molecules retrieved from the latest version of ZINC database, ZINC15[14] . We selected molecules using the filters clean (Substances with ”clean” reactivity), in-vitro (Substances reported or inferred active at 10 uM or better in direct binding assays) and now (Immediate delivery, includes in-stock and agent). The collection contains 7035 molecules.

The Self-Adaptive MOEA works on a population size of 50 that are the settings for the second level eMEGAs. Self-Adaptive MOEA's chromosome is shown in Figure 20. Self-Adaptive MOEA operates with a mutation probability set to 15% while the crossover probability was set to 80%. Parent selection was set to roulette. For the elitist generation selection Self-Adaptive MOEA was set to use phenotype diversity. The second level eMEGAs operate on a population size of 250. eMEGAs operate on mutations probability in the range of 0 to 0.2 and crossover probabilities in the range of 0.8 to 1.0. Results were assessed after 50 iterations. A synopsis of Self-Adaptive MOEA settings can be found in Table 16.

Table 16: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) experimental design settings

Self-Adaptive MOEA				
Dataset	Objectives	Population	Iterations	Evolutionary Operations
ZINC15 clean, in-vitro, now	Non Dominated Solutions Percentage Pareto Front Hypervolume	50	50	Mutation Probability: **15%** Crossover Probability: **80%** Selection Type: **Roulette** Diversity Type: **Phenotype**
eMEGAs				
ZINC15 clean, in-vitro, now	Structural Similarity Chemical Descriptors Similarity	250	1	Defined during run time. Based on Self-Adaptive MOEA's chromosomes.

The final solutions were filtered and we selected the ones with objective fitness score lower than 0.5 for each objective. As the objectives used, describe the distance to specific target molecules, we would like to have solutions that are closer (in case of similarity) from the defined target.

In order to validate the effectiveness of the proposed solutions, as a last step we perform a docking experiment to ER-α using the AutoDock Vina [232] tools provided by LiSIs [238]. The selected solutions were docked to ER-α 3ERT.

[14] http://zinc15.docking.org/

6.4.2 Results

In Figure 62 all the proposed solutions in objective space, are shown and in Figure 63 the non dominated proposed solutions in objective space, are shown. All solutions within the range of 0 to 0.5 for both objectives are considered good solutions, though we decided to use the non dominated solutions for the docking experiment just for sake of simplicity. The solutions fill the space in an arc between the range of 0 to 0.2 in Y-axis and 0.2 to 0.3 in X-axis. In total there are 2 non dominated solutions. As pointed above we are interested for the solutions between 0 and 0.5 range for both objectives. Which are the 2 non dominated solutions shown in Figure 64.

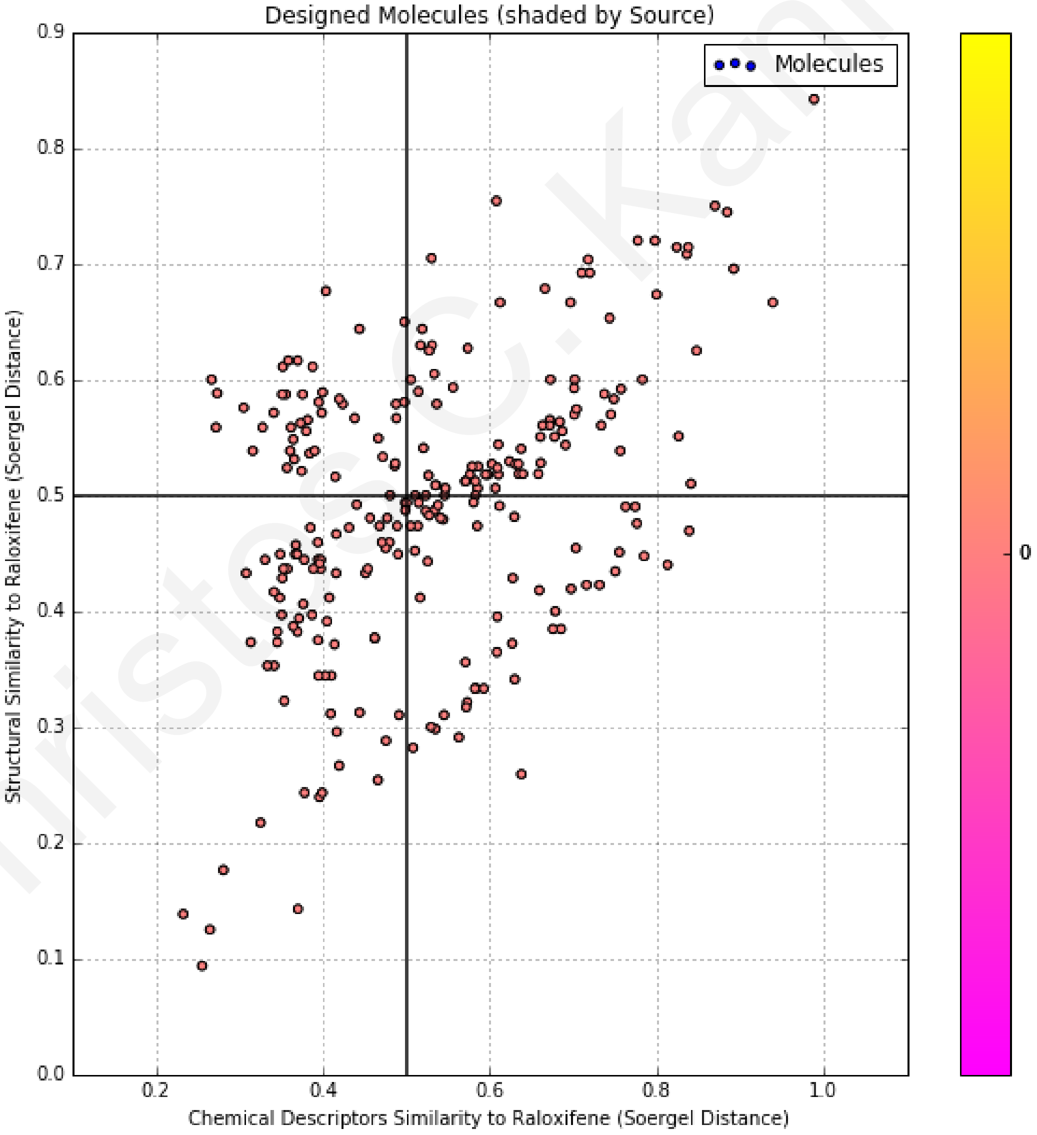


Figure 62: Designed molecules in objective space.

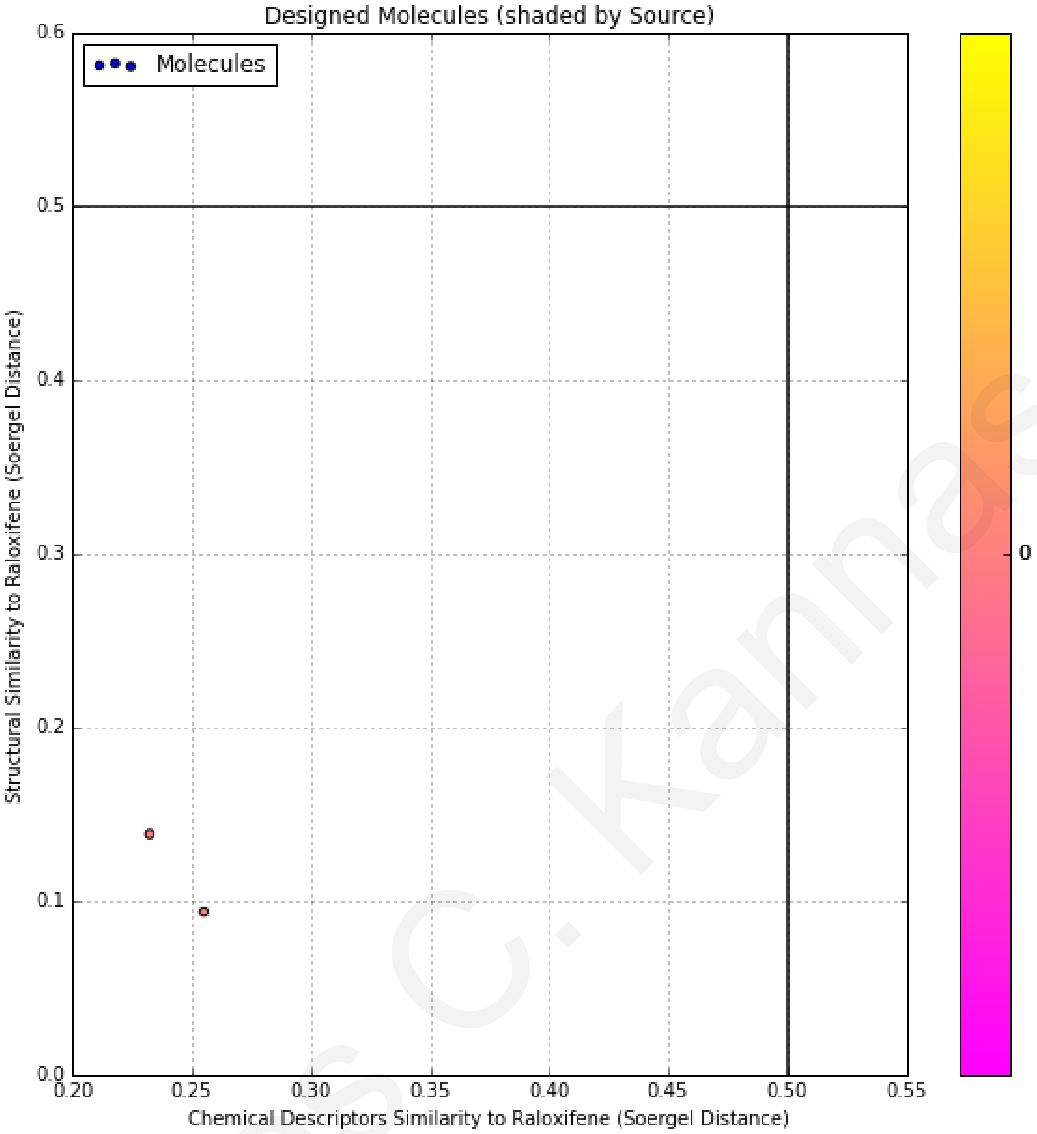


Figure 63: Non dominated designed molecules in objective space.

Table 17 shows the docking experiments results, sorted in ascending order at Docking Affinity column. Figures 65 and 66 show the selected molecules at their docked position to ER-α docking site.

Table 17: AutoDock Vina docking to ER-α results

Molecule Id	Docking Affinity (kcal/mol)
Raloxifene	**-2.2 (-11.7 PubChem)**
DnD_31_SP_194_48_M_49	-8.2
DnD_34_SP_197_49_X_13a	-5.9

The secondary output of Self-Adaptive MOEA are the proposed settings for eMEGA for the problem. In Table 18 there are the non dominated proposed settings.

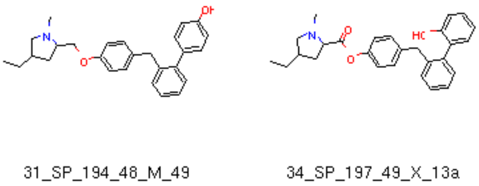


Figure 64: Designed molecules 2D depictions.

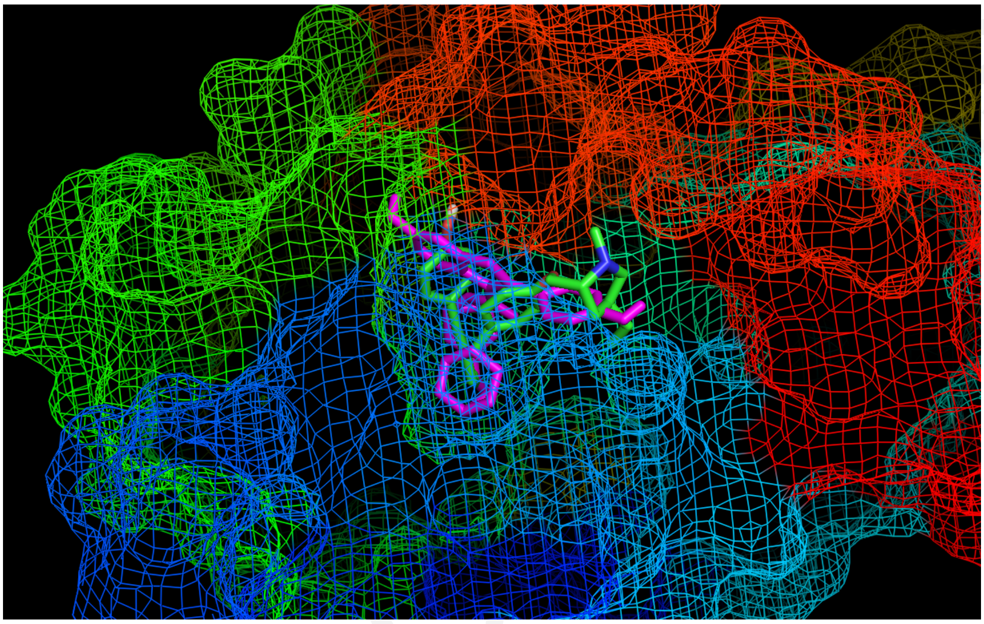


Figure 65: Designed molecule DnD_31_SP_194_48_M_49 docked to ER-α, in reference with Raloxifene (magenta).

6.4.3 Discussion

As shown from the docking experiment results in Table 17, the selected solutions have good docking affinity to ER-α, which is between -5.9 and -8.2 kcal/mol. From the visualisation of the docking conformations of the solutions in Figures 65 and 66, we see that the solutions fit well in the docking site of the protein.

From Table 18 we understand that the preferred eMEGA parameter settings are:

- *Mutation Probability* of *13%*,
- *Crossover Probability* of *98%*,
- *Parent Selection* based on *Roulette Selection*, and
- *Next Generation (Population) Selection* based on *Genotype Diversity*.

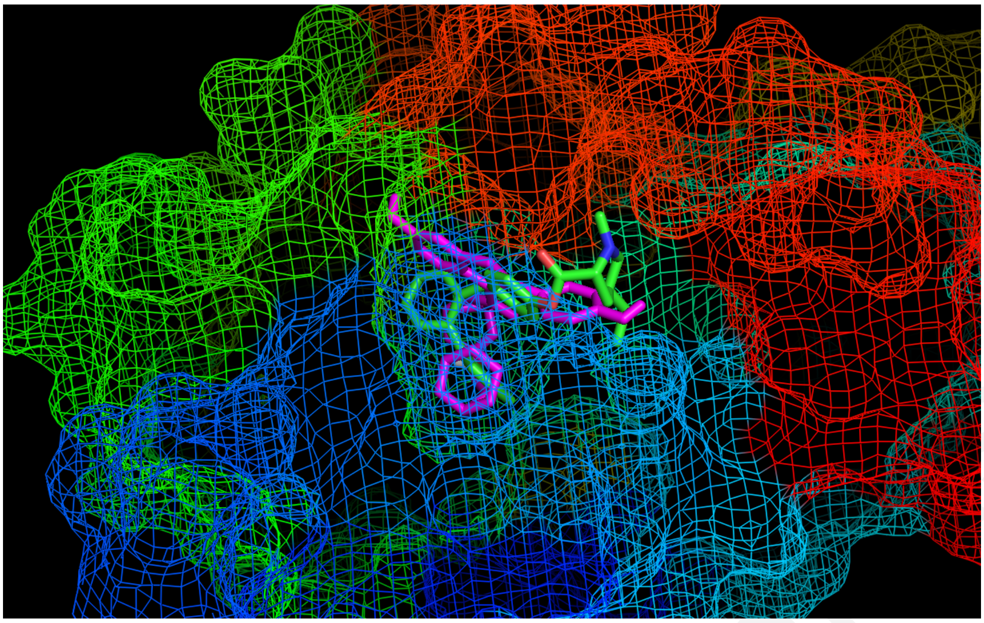


Figure 66: Designed molecule DnD_34_SP_197_49_X_13a docked to ER-α, in reference with Raloxifene (magenta).

Table 18: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) non dominated settings for elite Multi-Objective Evolutionary Graph Algorithm (eMEGA)

Mutation Probability	Crossover Probability	Selection Type	Diversity Type	Non Dominated %	Pareto Hypervolume	Rank
0.13	0.99	roulette	genotype	1.0	0.27	1
0.13	0.99	roulette	genotype	1.0	0.27	1
0.13	0.99	roulette	genotype	1.0	0.27	1
0.13	0.99	roulette	genotype	1.0	0.27	1
0.13	0.99	roulette	genotype	1.0	0.27	1
0.13	0.99	tournament	genotype	1.0	0.27	1

Note: The numbers for 'Non Dominated %' are 1 minus the actual %. The smaller the number listed here the better. 'Rank' is their non dominance rank.

This experiment required 5 hours (05:28:27) to complete the 50 iterations.

From discovery informatics point of view the selected designed compounds look promising, despite the low docking affinity of the molecule in Figure 66. It is known that Raloxifene (Figure 61) binds to ER-α by different mechanism than Tamoxifen (Figure 39) [273]. Further investigation is required to investigate the behaviour of the protein-ligand complex, which requires *in-vitro* experiments.

6.5 Use Case 4: Design Proteasome B5 inhibitors based on similarity to Ixazomib

Proteasome subunit beta type-5 as known as 20S proteasome subunit beta-5 is a protein that in humans is encoded by the PSMB5 gene. This protein is one of the 17 essential subunits (alpha subunits 17, constitutive beta subunits 17, and inducible subunits including beta1i, beta2i, beta5i) that contributes to the complete assembly of 20S proteasome complex. In particular, proteasome subunit beta type-5, along with other beta subunits, assemble into two heptameric rings and subsequently a proteolytic chamber for substrate degradation. This protein contains ”chymotrypsin-like” activity and is capable of cleaving after large hydrophobic residues of peptide. The eukaryotic proteasome recognized degradable proteins, including damaged proteins for protein quality control purpose or key regulatory protein components for dynamic biological processes. An essential function of a modified proteasome, the immunoproteasome, is the processing of class I Major Histocompatibility Complex peptides.

6.5.1 Methodology

The objective of the experiment was to design molecules that have structural and chemical descriptors similarity to Ixazomib[15] (Figure 67) which is a known Proteasome B5 inhibitor (PSMB5).

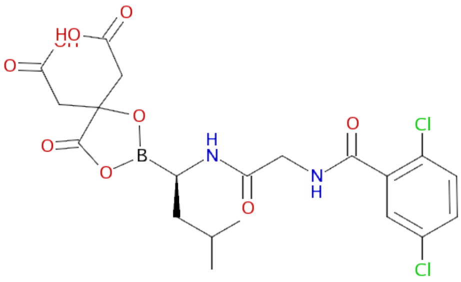


Figure 67: Ixazomib.

[15] https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL3545432

The objectives for the molecular design algorithm were, structural similarity based on Soergel distance [272] and chemical descriptors similarity based on Euclidean distance [272] to Ixazomib (Figure 67). The optimization process was aimed to minimize both of those objectives to 0.

Structural similarity is an objective fitness function that calculates a fitness score for a molecule to the target molecule, computed as the graph distance based on their MCS[16] , an in-house implementation based on RASCAL [269].

Chemical descriptors similarity is an objective fitness function that calculates a fitness score for a molecule to the target molecule, computed as the distance of their chemical descriptors vector. The chemical descriptors are based on an in-house implementation of atom pairs and topological torsions calculations for each molecule [270] and [271].

For input we used a collection of molecules retrieved from the latest version of ZINC database, ZINC15[17] . We selected molecules using the filters clean (Substances with ”clean” reactivity), in-vitro (Substances reported or inferred active at 10 uM or better in direct binding assays) and now (Immediate delivery, includes in-stock and agent). The collection contains 7035 molecules.

The Self-Adaptive MOEA works on a population size of 50 that are the settings for the second level eMEGAs. Self-Adaptive MOEA's chromosome is shown in Figure 20. Self-Adaptive MOEA operates with a mutation probability set to 15% while the crossover probability was set to 80%. Parent selection was set to roulette. For the elitist generation selection Self-Adaptive MOEA was set to use phenotype diversity. The second level eMEGAs operate on a population size of 250. eMEGAs operate on mutations probability in the range of 0 to 0.2 and crossover probabilities in the range of 0.8 to 1.0. Results were assessed after 50 iterations. A synopsis of Self-Adaptive MOEA settings can be found in Table 19.

The final solutions were filtered and we selected the ones with objective fitness score lower than 0.5 for each objective. As the objectives used, describe the distance to specific target molecules, we would like to have solutions that are closer (in case of similarity) from the defined target.

[16] https://en.wikipedia.org/wiki/Maximum_common_subgraph

[17] http://zinc15.docking.org/

Table 19: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) experimental design settings

Self-Adaptive MOEA				
Dataset	Objectives	Population	Iterations	Evolutionary Operations
ZINC15 clean, in-vitro, now	Non Dominated Solutions Percentage Pareto Front Hypervolume	50	50	Mutation Probability: **15%** Crossover Probability: **80%** Selection Type: **Roulette** Diversity Type: **Phenotype**
eMEGAs				
ZINC15 clean, in-vitro, now	Structural Similarity Chemical Descriptors Similarity	250	1	Defined during run time. Based on Self-Adaptive MOEA's chromosomes.

In order to validate the effectiveness of the proposed solutions, as a last step we perform a docking experiment to Proteasome B5 using the AutoDock 4 [274]. The docking experiments were performed by Dr. Erika Loizidou.

6.5.2 Results

In Figure 68 all the proposed solutions in objective space, are shown and in Figure 69 the non dominated proposed solutions in objective space, are shown. All solutions within the range of 0 to 0.5 for both objectives are considered good solutions, though we decided to use the non dominated solutions for the docking experiment just for sake of simplicity. The solutions fill the space in an arc between the range of 0.1 to 0.3 in Y-axis and 0.3 to 0.5 in X-axis. In total there are 3 non dominated solutions. As pointed above we are interested for the solutions between 0 and 0.5 range for both objectives. Which are the 3 non dominated solutions shown in Figure 70.

Table 20 shows the docking experiments results, sorted in ascending order at Docking Affinity column. Figures 71, 72 and 73 show the selected molecules at their docked position to Proteasome B5 docking sites.

Table 20: AutoDock 4 docking to Proteasome B5 results

Molecule Id	Docking Affinity (kcal/mol)
DnD_19_SP_196_48_X_59b	-7.19
DnD_49_SP_193_48_X_123b	-6.68
DnD_1_SP_196_48_X_67a	-6.08

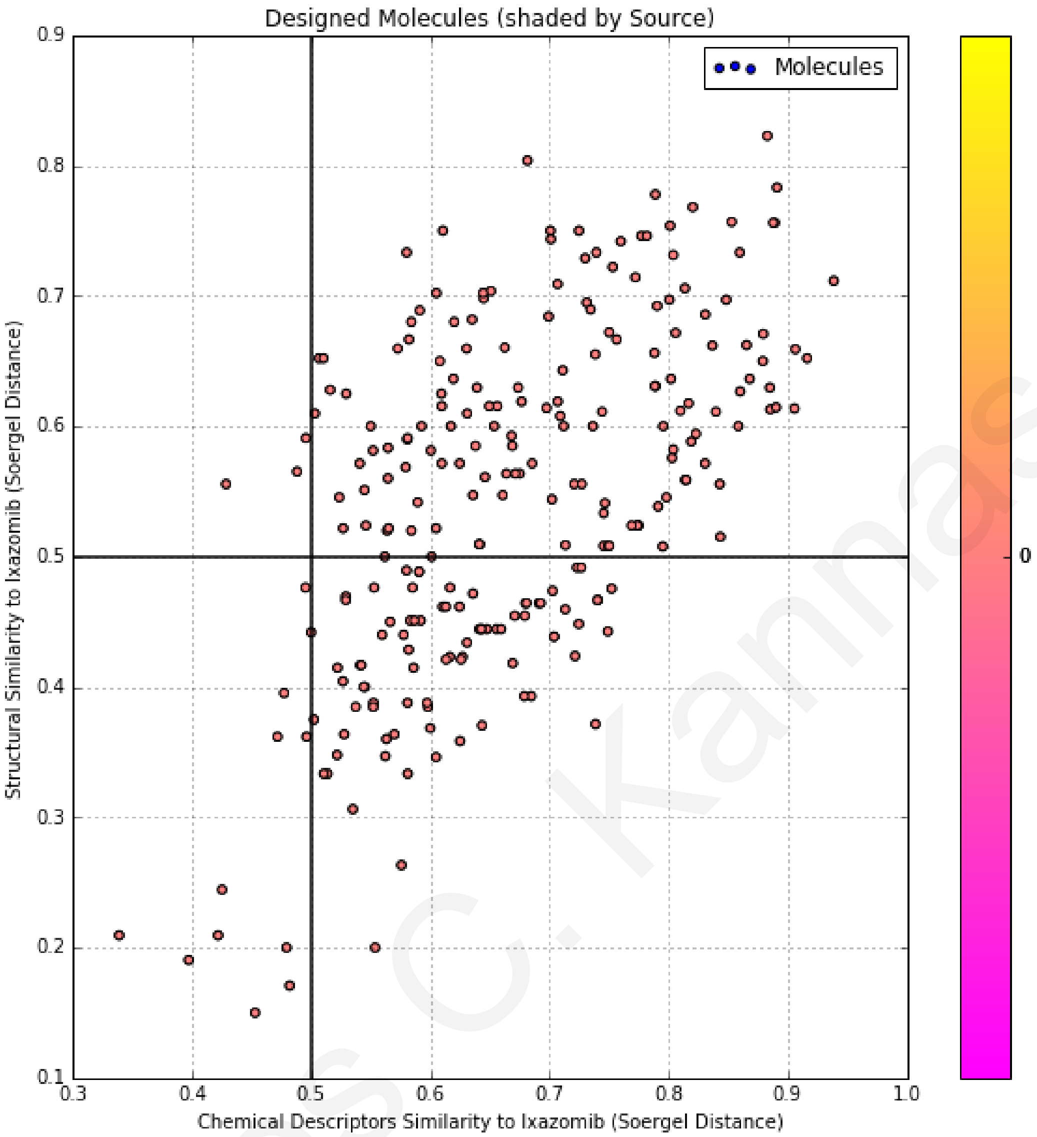


Figure 68: Designed molecules in objective space.

The secondary output of Self-Adaptive MOEA are the proposed settings for eMEGA for the problem. In Table 21 there are the non dominated proposed settings.

6.5.3 Discussion

As shown from the docking experiment results in Table 20, the selected solutions have good docking affinity to Proteasome B5, which is between -6.08 and -7.19 kcal/mol. From the visualisation of the docking conformations of the solutions in Figures 71, 72 and 73, we see that the solutions fit well in the docking sites of the protein.

From Table 21 we understand that the preferred eMEGA parameter settings are:

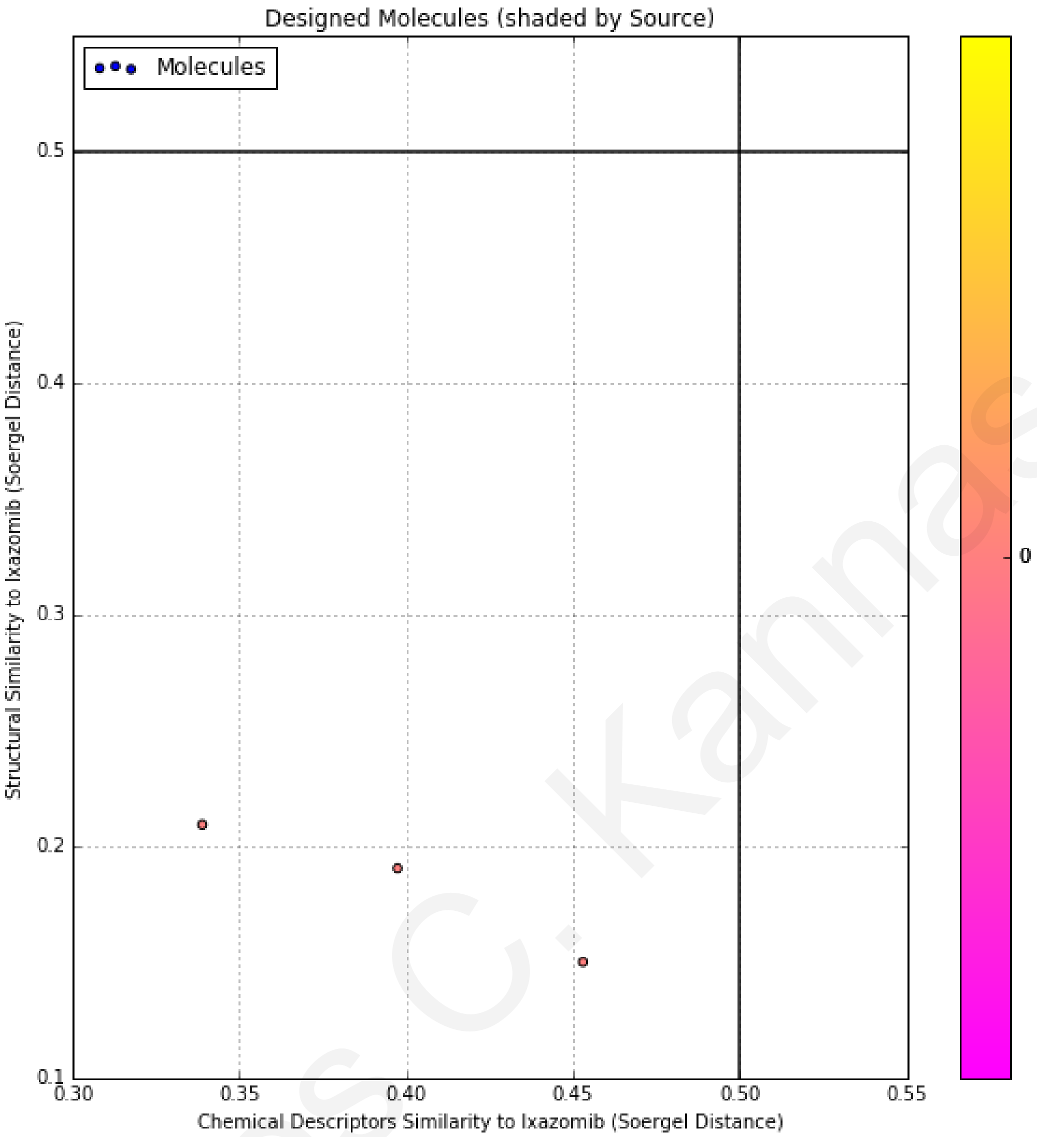
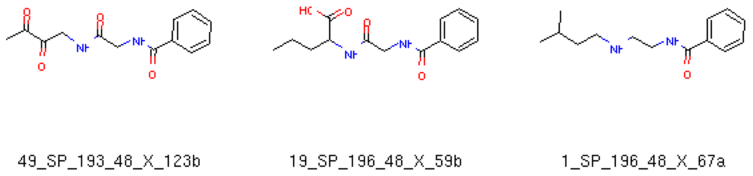


Figure 69: Non dominated designed molecules in objective space.

- *Mutation Probability* of *9%*,
- *Crossover Probability* of *98%*,
- *Parent Selection* based on *Roulette Selection*, and
- *Next Generation (Population) Selection* based on *Genotype Diversity* or *Phenotype Diversity*.

This experiment required 23 hours (23:09:38) to complete the 50 iterations.



49_SP_193_48_X_123b

19_SP_196_48_X_59b

1_SP_196_48_X_67a

Figure 70: Designed molecules 2D depictions.

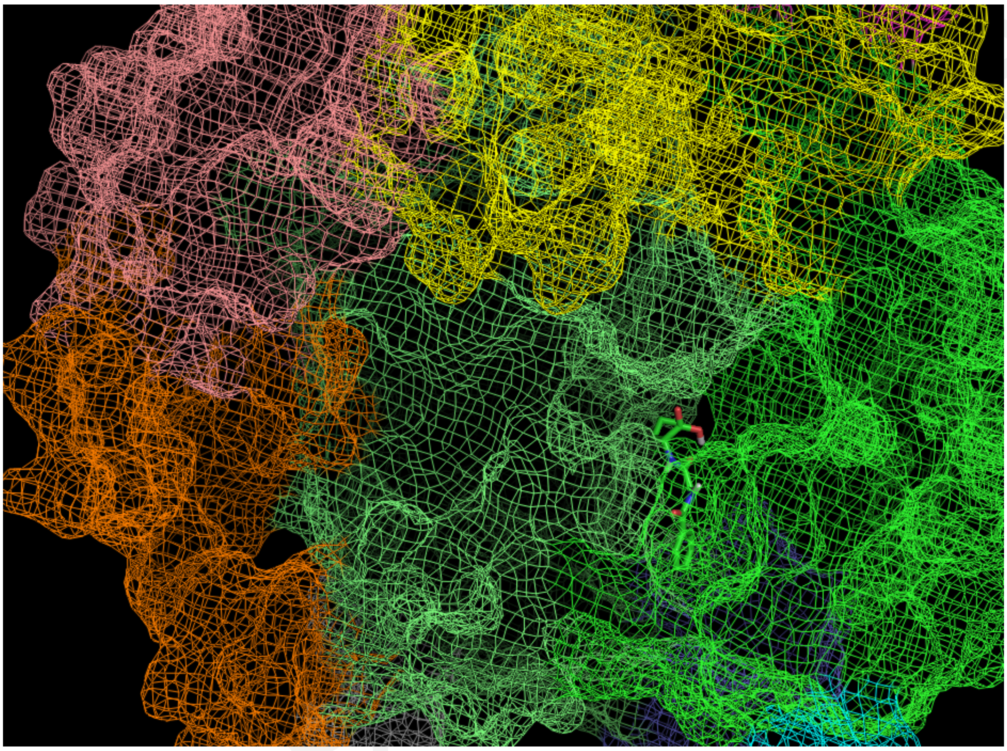


Figure 71: Designed molecule DnD_19_SP_196_48_X_59b docked to Proteasome B5.

From discovery informatics point of view the selected designed compounds look promising, but unfortunately we can not compare them to Ixazomib's docking affinity as AutoDock 4 does not identify the Boron atom present in Ixazomib (Figure 67) and as such it can not perform a docking experiment to compare it. Further investigation is required to investigate the behaviour of the protein-ligand complex, which requires *in-vitro* experiments.

6.6 Overall Discussion

The execution time of Self-Adaptive MOEA depends on a multitude of parameters, i.e. the maximum iterations, population size, the contents of the starting population and the evolutionary search parameters (mutation and crossover probabilities).

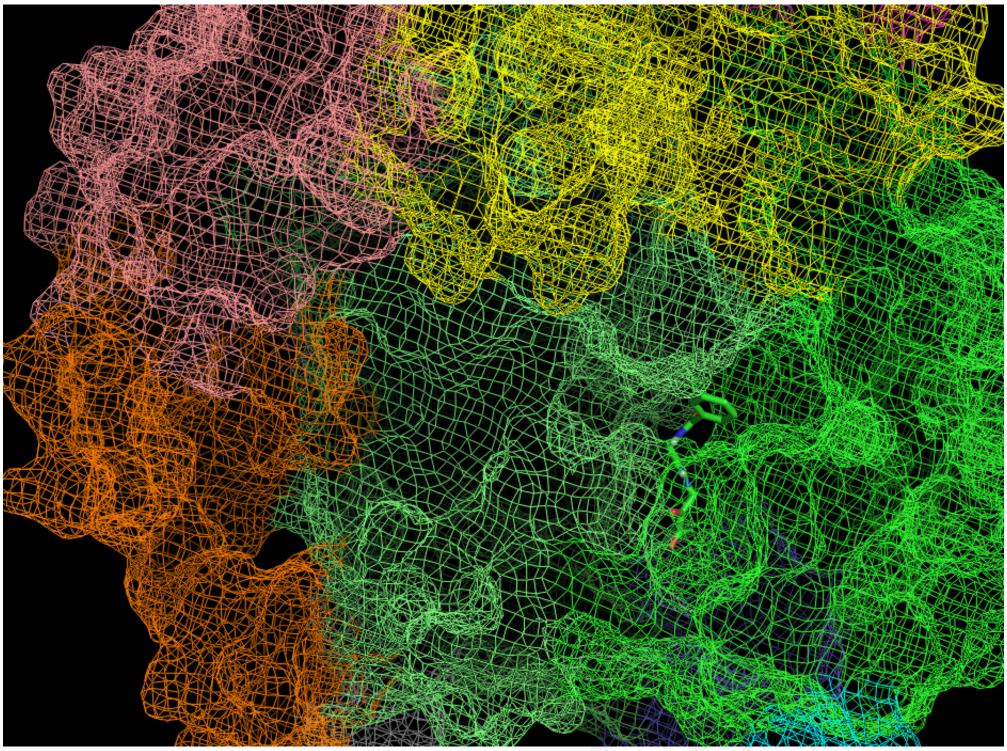


Figure 72: Designed molecule DnD_49_SP_193_48_X_123b docked to Proteasome B5.

Table 21: Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA) non dominated settings for elite Multi-Objective Evolutionary Graph Algorithm (eMEGA)

Mutation Probability	Crossover Probability	Selection Type	Diversity Type	Non Dominated %	Pareto Hypervolume	Rank
0.10	0.98	tournament	phenotype	1.0	0.44	1
0.10	0.98	roulette	phenotype	1.0	0.44	1
0.10	0.98	roulette	genotype	1.0	0.43	1
0.10	0.98	roulette	phenotype	1.0	0.44	1
0.09	0.98	roulette	genotype	1.0	0.44	1

Note: The numbers for 'Non Dominated %' are 1 minus the actual %. The smaller the number listed here the better. 'Rank' is their non dominance rank.

Use cases 1 to 4 use the same settings for Self-Adaptive MOEA except for the iteration number and the molecular design objectives, which are different for every use case. As such we can compare their execution times and derive some conclusions for Self-Adaptive MOEA's efficiency. As seen previously on the dedicated use cases sections the algorithm requires a few hours to a few days to complete a given task. With the help of Table 22 we are investigating the average time that is required for Self-Adaptive MOEA to complete an iteration. To do so we make an estimation by dividing the total time, in hours, by the number of iterations. From the results we see that the average time per iteration is not constant. Use cases 1 and 2 require 13 and 18 minutes per iteration while use case 3 requires only 7 minutes per iteration, and lastly use case 4 requires an astonishing 29 minutes per

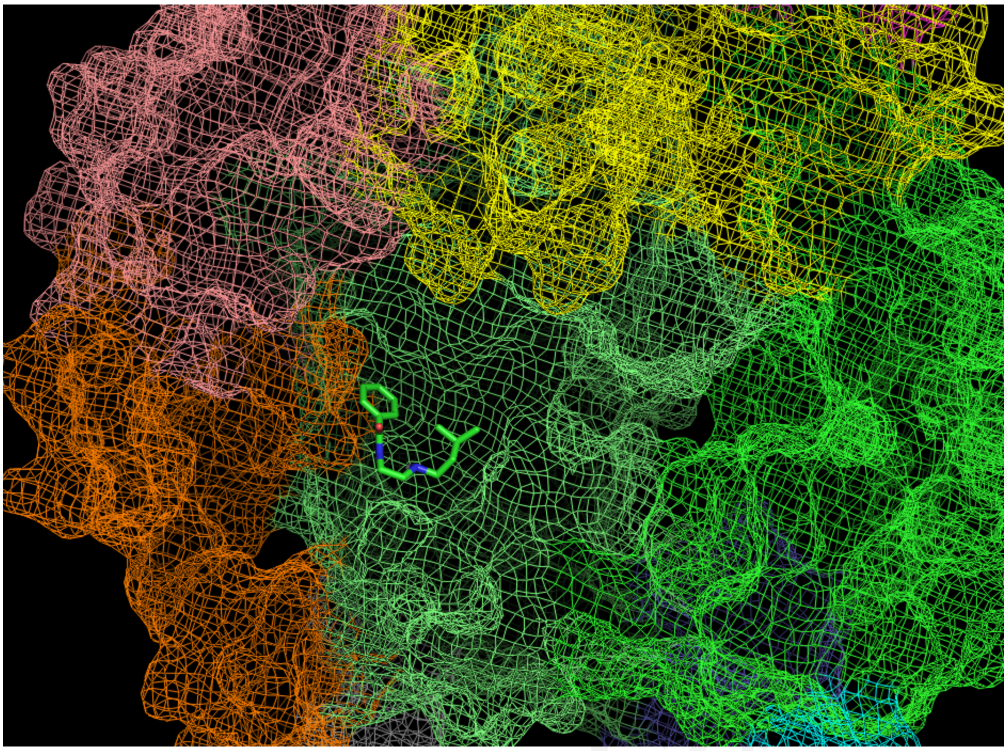


Figure 73: Designed molecule DnD_1_SP_196_48_X_67a docked to Proteasome B5.

iteration. These discrepancies can be attributed to the difficulty of the problem and the load of the system.

The number of iterations was heuristically derived to guarantee that convergence has been achieved. Numerous runs were carried out, starting from a small number of iterations and increasing it stepwise. The generated solutions were investigated qualitatively.

It is important to remember that Self-Adaptive MOEA executes a number of eMEGAs per iteration, which in use cases 1 to 4 are 50 eMEGAs for the first half and an additional number of eMEGAs for the second half, based on the evolutionary process stochastic nature. Each eMEGA proposes 250 solutions so Self-Adaptive MOEA has 12500 (50 x 250) solutions to look into at the first half of the iteration, though many of them are identical solutions, because the different eMEGAs start from the same population and eventually will make the same decisions (select identical mutations and crossover pairs), so the actual number of unique solutions is much lower.

As mentioned in the use cases discussion sections the proposed solutions look promising, but further investigation is required to investigate the behaviour of their protein-ligand complex, which requires *in-vitro* experiments.

Table 22: Self-Adaptive MOEA's total time and the estimated average time per iteration for use cases 1 to 4.

Use Case	Execution Time (Hours:Minutes:Seconds)	Iterations	Average Time per Iteration (Minutes)
1	106:16:43	500	13 (107/500)
2	29:35:36	100	18 (30/100)
3	05:28:27	50	7 (6/50)
4	23:09:38	50	29 (24/50)

Looking at the Self-Adaptive MOEA proposed final settings for eMEGA for each experiment we noticed that different search parameters were proposed. This occurs due to the difference in the nature of the respective problem as the algorithm has to search in different regions of the molecular search space of candidate solutions. With regards to the preferred crossover rate, in all use cases Self-Adaptive MOEA prefers high crossover rate, as it enables eMEGA to perform a better global search. With regards to the preferred mutation rate, in use cases 1 (Section 6.2), 3 (Section 6.4) and 4 (Section 6.5) a high mutation rate is also preferred, as this enables eMEGA to achieve a better local search. In contrast to use case 2 (Section 6.3) where the preferred mutation rate is low (3%), which shows that for the specific problem the local search does not produce good solutions. With regards to the preferred parent selection mechanism, in use cases 1 (Section 6.2) and 2 (Section 6.3) tournament selection is the preferred selection mechanism, which means that eMEGA selects the fittest individuals as parents. In contrast, in use cases 3 (Section 6.4) and 4 (Section 6.5) roulette selection is the preferred selection mechanism, which means that eMEGA selects individuals as parents in a stochastic approach based on the efficiency of each individual. With regards to the preferred diversity selection mechanism that is a clustering of the population based on chromosome or objective fitness scores, in use cases 1 (Section 6.2) and 3 (Section 6.4) genotype diversity is preferred as it enables eMEGA to select individuals for the next generation based on the diversity of their chromosomes. In contrast, in use cases 2 (Section 6.3) and 4 (Section 6.5) genotype and phenotype diversity are equally preferred as the first enables eMEGA to select individuals for the next generation based on their chromosome diversity and the latter enables eMEGA to select individuals for the next generation based on the diversity of their objective fitness scores.

6.7 Concluding Remarks

In the validation experiment Self-Adaptive MOEA performed reasonably well considering that it run for less iterations (100) than eMEGA (500 iterations). The proposed solutions covered larger region in the objective space. Similarly in the use cases presented, Self-Adaptive MOEA proposes interesting solutions in relative low iteration numbers, but with the need of more execution time. This is due to the architecture of the algorithm that involves executing several instances of the inner algorithm (eMEGA in this case). Due to this feature Self-Adaptive MOEA generates and evaluates a large number of solutions per iteration.

In every experiment Self-Adaptive MOEA was used for, preferred a different set of search parameters for its inner algorithm (eMEGA). This should have been expected to some extend as each problem has to search in different region of the vast chemical space, using a different starting point featured by the starting population.

Self-Adaptive MOEA has been built with adaptability in mind that is to be able to be used with different inner MOEAs adapted for other problems. To aid this decision the objective fitness functions for the self-adaptive part of Self-Adaptive MOEA (outer loop) are designed to evaluate the effectiveness and the progression of any MOEA used in the inner loop.

The chosen chromosome structure enables Self-Adaptive MOEA to be expandable with additional search parameters for optimisation and for future implementations of additional self-adaptive techniques, i.e. select applied evolutionary operators.

Self-Adaptive MOEA has been designed to leverage multi-core parallelism, where possible, by running a number of eMEGAs concurrently. This number is defined by the number of cores the system has minus 1, which is reserved for the controlling parent process that also runs the self-adaptive techniques.

Chapter 7

Concluding Remarks and Future Work

This chapter discusses our concluding remarks for **Li**fe **S**ciences **I**nformatic**s** (LiSIs) and Self-Adaptive Multi-Objective Evolutionary Algorithm (Self-Adaptive MOEA), and provides our thoughts for future work.

7.1 Concluding Remarks

7.1.1 Concluding Remarks for Life Sciences Informatics platform

To the best of our knowledge LiSIs was the first free web based Scientific Workflow Management System (SWMS) for the community of Life Sciences Informatics, although there were at least two other SWMS desktop based platforms (Taverna [49], [55], [56] and KNIME [60], [61]), though it was the first to utilise a web based interface. At its current stage (the objective of GRANATUM project[1]) it features a Web based Virtual Screening platform, focused for Cancer Chemoprevention Research. LiSIs as an integrated web based Virtual Screening (VS) framework achieved its goal to the fullest. We managed to implement a platform that is adaptable, expandable and scalable (on HPC cluster or on cloud based servers), features that are inherited from Galaxy. With some extra work on the available tools LiSIs, noted in Section 7.2.1, will be able to achieve the scalability of Galaxy.

[1] www.granatum.org

LiSIs was successfully used to support our expert partners in Cancer Chemoprevention with their hypotheses. We implemented a number of Scientific Workflows (SWs) that were: (a) preparing docking models, (b) preparing predictive models, (c) performing docking experiments, (d) using predictive models to predict biochemical properties and behaviour, and (e) performing VS workflows. As shown in Sections 4.9 and 4.11 LiSIs was successful in identifying novel cancer chemopreventive agents from molecules retrieved from Indofine's datasets.

LiSIs was build with in mind to be expanded in the future with tools featuring the algorithms from Multi-Objective Evolutionary Graph Algorithm (MEGA) framework.

A general concluding remark about the bottlenecks of SWMS are: (a) the need of dedicated hardware and/or software for specific tasks, i.e. docking, molecular dynamics, and (b) the heterogeneity in Input/Output (I/O) capabilities, information flow and scalability among the tools comprising the SWMS requires deep knowledge of the weaker players performance since it will affect the whole performance of the pipeline.

7.1.2 Concluding Remarks for Self-Adaptive Multi-Objective Evolutionary Algorithm

Self-Adaptive MOEA was used to design molecules that bear similarity to Seliciclib[2] (Section 6.1), Tamoxifen[3] (Sections 6.2 and 6.3), Raloxifen[4] (Section 6.4) and Ixazomib[5] (Section 6.5), across different experiments.

In general Self-Adaptive MOEA compared to elite Multi-Objective Evolutionary Graph Algorithm (eMEGA): (a) searches a larger space, (b) generates far more solutions per iteration, (c) proposes solutions in wider range, (d) requires more time for the same iterations (due to the fact that runs multiple eMEGAs per iteration), (e) evaluates different sets of parameter options for eMEGA for the given problem, and (f) proposes the fittest parameter sets that should be used from eMEGA for the given problem. The last two points are important specifically when we need to fine tune our

[2] https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL14762

[3] https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL83

[4] https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL177798

[5] https://www.ebi.ac.uk/chembl/compound/inspect/CHEMBL3545432

Multi-Objective Evolutionary Algorithm (MOEA), because the general approach of finding what parameter settings yield better results in any given MOEA requires to perform multiple experiments with different settings before focusing to the set of parameters that yields better solutions.

Someone would expect that Self-Adaptive MOEA would have outperformed eMEGA, because in theory changing the settings that drive the evolutionary process should make the algorithm be more efficient and intelligent. In practice we noticed that it performs slightly worse. The reasons are: (a) how an algorithm performs is not dictated only by its parameters, (b) input data and data generated during the iterative process have a significant role, and (c) Self-Adaptive MOEA's true use case is to provide us with information of how the underlying algorithm, eMEGA in our case, behaves with the data given to it within the problem to solve, in order to configure eMEGA in a way to tackle with the problem as good as it can.

Self-Adaptive MOEA is a useful tool for fine tuning the underlying MOEA to approximate a given problem. In the hands of an experienced user it can prove very powerful, as the expert can guide Self-Adaptive MOEA to the range of settings and the algorithm will propose the ones that tackle the problem better.

Our proposed Self-Adaptive MOEA has been built with adaptability in mind to be able to be used with different MOEAs in the inner loop. So we decided to implement objective fitness functions (Section 5.3) that would be able to be used to evaluate the effectiveness and the progression of any MOEA. This decision also helps to use Self-Adaptive MOEA for other applications different from molecular De Novo Design (DND). This also applies, given that some prior information about the model of the problem under investigation is known.

By choosing to use the specific chromosome (Figure 20) in our Self-Adaptive MOEA we made the algorithm expandable with additional search parameters for optimisation and for future implementations of additional self-adaptive techniques.

It is highlighted that the main Self-Adaptive MOEA's features are: (a) objective fitness functions that can evaluate the effectiveness and the progression of MOEAs, and (b) an expandable chromosome for MOEA's search parameters. These enable the Self-Adaptive MOEA framework to be configurable so that is able to use different MOEAs or Multi-Objective Optimization (MOO) algorithms in the inner loop, and as such is applicable in problems in various disciplines. This is possible because Self-Adaptive MOEA's domain problem solving functionality is the responsibility of the inner loop algorithm. Though, for this to be possible the following two modifications are required: (a) define the MOO algorithm that is used in Self-Adaptive MOEA's inner loop, and (b) formulate a model of the domain problem to solve.

From the first iterations of the design phase we realised that Self-Adaptive MOEA would benefit greatly if we leveraged multi-core parallelism. Because Self-Adaptive MOEA relies on running multiple eMEGAs (with different search parameters) per iteration. The decision was to implement Self-Adaptive MOEA to use all cores of the system, by assigning one eMEGA process per core. This approach helps to reduce Self-Adaptive MOEA's execution time. Though this decision has a caveat, running multiple eMEGAs requires also to have sufficient memory for each active eMEGA. From our experience a single eMEGA requires 6GB to 10GB of RAM, on large scale experiments. As such we can deploy Self-Adaptive MOEA on a cloud based machine, i.e. Amazon Web Services[6] and Google Cloud Platform[7] , with sufficient amount of memory per core to perform large experiments.

7.2 Future Work

A number of different research directions have already been initiated to expand on the work presented in this thesis. These initiatives can be grouped into two general categories; the first involves research on algorithmic enhancements and improvements of the computational performance of LiSIs while the second focuses on research on algorithmic enhancements, improvements of the computational performance and on problem specific applications of the Self-Adaptive MOEA. The potential directions of future work are outlined below.

[6] https://aws.amazon.com/

[7] https://cloud.google.com/

7.2.1 Future Work for Life Sciences Informatics platform

As stated above LiSIs has so much potential and was developed with future upgrade-ability and expandability.

The first step would be to develop LiSIs 2.0, based on the updated Galaxy[8] , and changing the tools to a form that will enable them to be deployed via Galaxy's specialised Tool Shed[9] ,[10] . This will make LiSIs up to date and re-enable its scalability functionality.

The second step would be to update LiSIs with a feature to visualise intermediate results from various tools, as there is the need for the users to be able to see these results. In the current version, tools that generate these intermediate results, store them in a binary format as there are data that should not be altered.

The third step would be to expand LiSIs tools with tools featuring the MEGA line-up of algorithms and Self-Adaptive MOEA. This work will expand LiSIs tools in the region of molecular DND.

An interesting research route would be to explore resource management in SWMSs, with a goal to suggest SW optimisation approaches. This research could have two branches: (i) **Novel Scientific Workflow Multi-Objective Optimization approaches:** Identify and implement novel MOO approaches for optimising the design of SWs on a SWMS, (ii) **Novel Multi-Objective Optimization scheduling approaches of Scientific Workflows:** Identify and implement novel MOO approaches for scheduling and optimising the execution of SWs on a SWMS.

7.2.2 Future Work for Self-Adaptive Multi-Objective Evolutionary Algorithm

The proposed Self-Adaptive MOEA has been designed to be adaptable, expandable and scalable (utilising multi-core parallelism). We should exploit these features to transform it to a robust Multi-Objective Optimization Problem (MOOP) and Many-Objective Optimization Problem (MaOOP) framework for searching near optimal solutions in a wide range of problems.

[8] https://galaxyproject.org/

[9] https://new.galaxyproject.org/toolshed/

[10] https://toolshed.g2.bx.psu.edu/

We noticed that Self-Adaptive MOEA proposed interesting solutions in all problems that has been applied to. As mentioned in Sections 6.1.3, 6.2.3, 6.3.3, 6.4.3 and 6.5.3 further *in-vitro* investigation is required to understand the behaviour of the proposed designed molecules in real environment.

As stated previously in Section 6.1.3 Self-Adaptive MOEA and its underling eMEGA are resource hungry algorithms specifically in large experiments. This should be addressed in the near future, as many optimisations can be implemented primarily in the MEGA framework. This will also benefit all the variations of MEGAs. This work can be the core of a future Ph.D. as it tackle the problem of algorithm optimisation via memory management and parallelism.

At the current state of Self-Adaptive MOEA the self-adaptive technique is applied only on the parameters of the underlying MOEA. As shown in Section 3.5, MOEA's behaviour is not driven only by its parameters, but also by the genetic operators that is using. When there are multiple mutation and crossover operators to choose from, in practice we leave it to chance or we enable the ones we think might work better with our problem and data. There has been substantial research in self-adaptive techniques that try to tackle the problem of choosing the operators the underlying MOEA works on. Self-Adaptive MOEA should be updated to have a self-adaptive technique for the genetic operators eMEGA can use.

The MEGA framework provides access to different variations of MEGA: (a) that use a niching mechanism (see eMEGA [245]), and (b) that use a local search algorithm to enhance MEGA's behaviour (see STagnation Identification and Resolution (STIR) in [245]). This is an interesting work as these different versions of MEGA operate on different settings and data structures.

An interesting implementation would be to apply Self-Adaptive MOEA in completely different problems by adapting Self-Adaptive MOEA to use other MOEAs and implementing models for other problems. For this to be possible, the following must be implemented: (a) do minor modifications to Self-Adaptive MOEA in order to be able to use any MOO algorithm with a simple parameter change, (b) define the MOO algorithm that is used in Self-Adaptive MOEA's inner loop, and (c) formulate a model of the domain problem to solve. An example of a generic problem, used in Computer Science to evaluate search algorithms, is the Travelling Salesman Problem[11] which can be expanded to be a MOOP.

7.3 Closing Paragraph

The research presented in this thesis introduces frameworks and algorithmic approaches applied in the domain of Life Sciences for Virtual Screening and Multi-Objective Molecular De Novo Design. The knowledge and expertise gained alongside with the platform and frameworks developed should be used towards exploring solutions for precision medicine.

[11] `http://www.math.uwaterloo.ca/tsp/`

References

[1] M. Hartenfeller and G. Schneider, "De novo drug design," *Methods in Molecular Biology (Clifton, N.J.)*, vol. 672, pp. 299–323, 2011. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20838974

[2] G. Schneider, M. Hartenfeller, M. Reutlinger, Y. Tanrikulu, E. Proschak, and P. Schneider, "Voyages to the (un)known: adaptive design of bioactive compounds," *Trends in Biotechnology*, vol. 27, no. 1, pp. 18–26, Jan. 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19004513

[3] J. Xu and A. Hagler, "Chemoinformatics and Drug Discovery," *Molecules*, vol. 7, no. 8, pp. 566–600, Aug. 2002. [Online]. Available: http://www.mdpi.com/1420-3049/7/8/566

[4] J. Drews, "Strategic trends in the drug industry," *Drug Discovery Today*, vol. 8, no. 9, pp. 411–420, May 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359644603026904

[5] J. K. Wegner, A. Sterling, R. Guha, A. Bender, J.-L. Faulon, J. Hastings, N. O'Boyle, J. Overington, H. Van Vlijmen, and E. Willighagen, "Cheminformatics," *Commun. ACM*, vol. 55, no. 11, pp. 65–75, Nov. 2012. [Online]. Available: http://doi.acm.org/10.1145/2366316.2366334

[6] A. S. Reddy, S. P. Pati, P. P. Kumar, H. N. Pradeep, and G. N. Sastry, "Virtual screening in drug discovery – a computational perspective," *Current protein & peptide science*, vol. 8, no. 4, pp. 329–351, Aug. 2007. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17696867

[7] C. McInnes, "Virtual screening strategies in drug discovery," *Current Opinion in Chemical Biology*, vol. 11, no. 5, pp. 494–502, Oct. 2007. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17936059

[8] I. Muegge and S. Oloff, "Advances in virtual screening," *Drug Discovery Today: Technologies*, vol. 3, no. 4, pp. 405–411, Dec. 2006. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1740674906000758

[9] S. Ekins, J. Mestres, and B. Testa, "*In silico* pharmacology for drug discovery: methods for virtual ligand screening and profiling," *British Journal of Pharmacology*, vol. 152, no. 1, pp. 9–20, Sep. 2007. [Online]. Available: http://doi.wiley.com/10.1038/sj.bjp.0707305

[10] V. Vyas, A. Jain, A. Jain, and A. Gupta, "Virtual Screening: A Fast Tool for Drug Design," *Scientia Pharmaceutica*, vol. 76, no. 3, pp. 333–360, 2008. [Online]. Available: http://www.scipharm.at/2008/3/333

[11] C. Wegscheid-Gerlach, *Chemoinformatics approaches to virtual screening*, A. Varnek and A. Tropsha, Eds. Cambridge, UK: Royal Society of Chemistry, 2008. [Online]. Available: http://dx.doi.org/10.1039/9781847558879

[12] W. L. Jorgensen, “The Many Roles of Computation in Drug Discovery,” *Science*, vol. 303, no. 5665, pp. 1813–1818, Mar. 2004. [Online]. Available: http://www.sciencemag.org/content/303/5665/1813

[13] A. Rusinko, M. W. Farmen, C. G. Lambert, P. L. Brown, and S. S. Young, “Analysis of a Large Structure/Biological Activity Data Set Using Recursive Partitioning1,” *Journal of Chemical Information and Computer Sciences*, vol. 39, no. 6, pp. 1017–1026, Nov. 1999. [Online]. Available: http://dx.doi.org/10.1021/ci9903049

[14] X. Chen, A. Rusinko, and S. Young, “Recursive Partitioning Analysis of a Large Structure-Activity Data Set Using Three-Dimensional Descriptors1,” *Journal of Chemical Information and Modeling*, vol. 38, no. 6, pp. 1054–1062, Nov. 1998. [Online]. Available: http://pubs.acs.org/cgi-bin/doilookup/?10.1021/ci980089g

[15] E. M. Krovat, T. Steindl, and T. Langer, “Recent Advances in Docking and Scoring,” *Current Computer - Aided Drug Design*, vol. 1, no. 1, pp. 93–102, Jan. 2005. [Online]. Available: http://www.ingentaselect.com/rpsv/cgi-bin/cgi?ini=xref&body=linker&reqdoi=10.2174/1573409052952314

[16] C. Bissantz, G. Folkers, and D. Rognan, “Protein-Based Virtual Screening of Chemical Databases. 1. Evaluation of Different Docking/Scoring Combinations,” *Journal of Medicinal Chemistry*, vol. 43, no. 25, pp. 4759–4767, Dec. 2000. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/jm001044l

[17] M. Stahl and M. Rarey, “Detailed analysis of scoring functions for virtual screening.” *Journal of Medicinal Chemistry*, vol. 44, no. 7, pp. 1035–1042, Mar. 2001. [Online]. Available: http://view.ncbi.nlm.nih.gov/pubmed/11297450

[18] B. Waszkowycz, T. D. J. Perkins, R. A. Sykes, and J. Li, “Large-scale virtual screening for discovering leads in the postgenomic era,” *IBM Systems Journal*, vol. 40, no. 2, pp. 360–376, 2001. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5386989

[19] P. Lyne, “Structure-based virtual screening: an overview,” *Drug Discovery Today*, vol. 7, no. 20, pp. 1047–1055, Oct. 2002. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1359644602024832

[20] C. A. Lipinski, F. Lombardo, B. W. Dominy, and P. J. Feeney, “Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings,” *Advanced Drug Delivery Reviews*, vol. 46, no. 13, pp. 3–26, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169409X00001290

[21] A. C. Anderson and D. L. Wright, “The Design and Docking of Virtual Compound Libraries to Structures of Drug Targets,” *Current Computer - Aided Drug Design*, vol. 1, no. 1, pp. 103–127, Jan. 2005. [Online]. Available: http://www.ingentaselect.com/rpsv/cgi-bin/cgi?ini=xref&body=linker&reqdoi=10.2174/1573409052952279

[22] A. Barker and J. V. Hemert, “Scientific Workflow: A Survey and Research Directions,” in *Proceedings of the 7th international conference on Parallel processing and applied mathematics*, ser. PPAM’07. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 746–753. [Online]. Available: http://dl.acm.org/citation.cfm?id=1786194.1786281

[23] T. McPhillips, S. Bowers, D. Zinn, and B. Ludscher, “Scientific workflow design for mere mortals,” *Future Generation Computer Systems*, vol. 25, no. 5, pp. 541–551, May 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167739X08000873

[24] Y. L. Simmhan, B. Plale, and D. Gannon, “A Survey of Data Provenance in E-Science,” *ACM SIGMOD Record*, vol. 34, no. 3, pp. 31–36, Sep. 2005. [Online]. Available: http://doi.acm.org/10.1145/1084805.1084812

[25] B. Ludscher, M. Weske, T. Mcphillips, and S. Bowers, "Scientific Workflows: Business as Usual?" in *Proceedings of the 7th International Conference on Business Process Management*, ser. BPM '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 31–47. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03848-8_4

[26] B. Ludscher, I. Altintas, S. Bowers, J. Cummings, T. Critchlow, E. Deelman, D. D. Roure, J. Freire, C. Goble, M. Jones, S. Klasky, T. McPhillips, N. Podhorszki, C. Silva, I. Taylor, and M. Vouk, "Scientific Process Automation and Workflow Management," in *Scientific Data Management*, ser. Computational Science Series, A. Shoshani and D. Rotem, Eds. Chapman & Hall, 2009. [Online]. Available: http://www.bibsonomy.org/bibtex/2764c2f395494f81ff2d6dd772840aea9/ludaesch

[27] K. Achilleos, C. Kannas, C. Nicolaou, C. Pattichis, and V. Promponas, "Open source workflow systems in life sciences informatics," in *2012 IEEE 12th International Conference on Bioinformatics Bioengineering (BIBE)*, 2012, pp. 552–558.

[28] J. Reed, R. Toombs, and N. A. Barricelli, "Simulation of biological evolution and machine learning," *Journal of Theoretical Biology*, vol. 17, no. 3, pp. 319–342, Dec. 1967. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0022519367900975

[29] R. S. Rosenberg, "Simulation of genetic populations with biochemical properties. I. The model." *Mathematical Biosciences*, vol. 7, pp. 233–57, 1970.

[30] R. Weinberg and M. Berkus, "Computer simulation of a living cell: Part II," *International Journal of Bio-Medical Computing*, vol. 2, no. 3, pp. 167–188, Jul. 1971. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0020710171900109

[31] I. Rechenberg, *Evolutionsstrategie; Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Stuttgart-Bad Cannstatt: Frommann-Holzboog, 1973, oCLC: 9020616.

[32] R. Mercer and J. Sampson, "ADAPTIVE SEARCH USING A REPRODUCTIVE METAPLAN," *Kybernetes*, vol. 7, no. 3, pp. 215–228, Mar. 1978. [Online]. Available: http://www.emeraldinsight.com/doi/abs/10.1108/eb005486

[33] J. Grefenstette, "Optimization of Control Parameters for Genetic Algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 16, no. 1, pp. 122–128, Jan. 1986.

[34] A. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 124–141, Jul. 1999. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=771166

[35] V. M. Spears and K. A. D. Jong, "On the virtues of parameterized uniform crossover," in *In Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1991, pp. 230–236. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.161.4101

[36] N. Saravanan, D. B. Fogel, and K. M. Nelson, "A comparison of methods for self-adaptation in evolutionary algorithms," *Biosystems*, vol. 36, no. 2, pp. 157–166, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/030326479501534R

[37] L. Batista, F. Campelo, F. Guimaraes, and J. Ramirez, "A new self-adaptive approach for evolutionary multiobjective optimization," in *2010 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2010, pp. 1–8.

[38] H. Jain and K. Deb, "An Improved Adaptive Approach for Elitist Nondominated Sorting Genetic Algorithm for Many-Objective Optimization," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, Eds. Springer Berlin Heidelberg, Jan. 2013,

no. 7811, pp. 307–321. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-37140-0_25

[39] J. M. Oliver, T. Kipouros, and A. M. Savill, "A Self-adaptive Genetic Algorithm Applied to Multi-Objective Optimization of an Airfoil," in *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, ser. Advances in Intelligent Systems and Computing, M. Emmerich, A. Deutz, O. Schuetze, T. Bck, E. Tantar, A.-A. Tantar, P. D. Moral, P. Legrand, P. Bouvry, and C. A. Coello, Eds. Springer International Publishing, 2013, no. 227, pp. 261–276, dOI: 10.1007/978-3-319-01128-8_17. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-01128-8_17

[40] A. Shahsavar, A. A. Najafi, and S. T. A. Niaki, "Three self-adaptive multi-objective evolutionary algorithms for a triple-objective project scheduling problem," *Computers & Industrial Engineering*, vol. 87, pp. 4–15, Sep. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360835215002053

[41] E. Zitzler, D. Brockhoff, and L. Thiele, "The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Springer Berlin Heidelberg, Mar. 2007, no. 4403, pp. 862–876, dOI: 10.1007/978-3-540-70928-2_64. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-70928-2_64

[42] M. Abouelhoda, S. Alaa, and M. Ghanem, "Meta-workflows: pattern-based interoperability between Galaxy and Taverna," in *Proceedings of the 1st International Workshop on Workflow Approaches to New Data-centric Science*, ser. Wands '10. New York, NY, USA: ACM, 2010, pp. 2:1–2:8. [Online]. Available: http://doi.acm.org/10.1145/1833398.1833400

[43] "KNIME | Konstanz Information Miner." [Online]. Available: http://www.knime.org/

[44] D. Hollingsworth, "The Workflow Reference Model," Tech. Rep., Jan. 1995. [Online]. Available: http://www.wfmc.org/standards/docs/tc003v11.pdf

[45] T. Fleuren, J. Gotze, and P. Muller, "Workflow Skeletons: Increasing Scalability of Scientific Workflows by Combining Orchestration and Choreography," in *2011 Ninth IEEE European Conference on Web Services (ECOWS)*, Sep. 2011, pp. 99 –106.

[46] J. Yu and R. Buyya, "A Taxonomy of Scientific Workflow Systems for Grid Computing," *ACM SIGMOD Record*, vol. 34, no. 3, pp. 44–49, 2005.

[47] C. Goble, P. Missier, and D. De Roure, "Scientific workflows," in *McGraw Hill 2009 Yearbook of Science & Technology*. McGraw-Hill Professional, 2008, vol. 11, no. 3. [Online]. Available: http://eprints.soton.ac.uk/266628/

[48] K. Grlach, M. Sonntag, D. Karastoyanova, F. Leymann, and M. Reiter, "Conventional Workflow Technology for Scientific Simulation," in *Guide to e-Science*, X. Yang, L. Wang, and W. Jie, Eds. London: Springer London, 2011, pp. 323–352. [Online]. Available: http://www.springerlink.com/index/10.1007/978-0-85729-439-5_12

[49] P. Missier, S. Soiland-Reyes, S. Owen, W. Tan, A. Nenadic, I. Dunlop, A. Williams, T. Oinn, and C. Goble, "Taverna, Reloaded," in *Scientific and Statistical Database Management*, ser. Lecture Notes in Computer Science, M. Gertz and B. Ludscher, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 6187, pp. 471–481. [Online]. Available: http://www.springerlink.com/content/n22t5r3272372190/abstract/

[50] E. Deelman, D. Gannon, M. Shields, and I. Taylor, "Workflows and e-Science: An overview of workflow system features and capabilities," *Future Generation Computer Systems*, vol. 25, no. 5, pp. 528–540, May 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X08000861

[51] I. J. Taylor, *Workflows for e-science scientific workflows for grids*. London: Springer, 2007. [Online]. Available: http://dx.doi.org/10.1007/978-1-84628-757-2

[52] A. Tiwari and A. K. Sekhar, "Workflow based framework for life science informatics," *Computational Biology and Chemistry*, vol. 31, no. 56, pp. 305–319, Oct. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1476927107001107

[53] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers, "Examining the Challenges of Scientific Workflows," *Computer*, vol. 40, no. 12, pp. 24 –32, Dec. 2007.

[54] V. Curcin and M. Ghanem, "Scientific workflow systems - can one size fit all?" in *Biomedical Engineering Conference, 2008. CIBEC 2008. Cairo International*. Cairo: IEEE, Dec. 2008, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp%3Farnumber%3D4786077

[55] D. Hull, K. Wolstencroft, R. Stevens, C. A. Goble, M. R. Pocock, P. Li, and T. Oinn, "Taverna: A tool for building and running workflows of services," *Nucleic Acids Research*, vol. 34, no. Web Server issue, pp. W729–W732, 2006.

[56] T. Oinn, M. Greenwood, M. Addis, M. N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin, P. Li, P. Lord, M. R. Pocock, M. Senger, R. Stevens, A. Wipat, and C. Wroe, "Taverna: Lessons in creating a workflow environment for the life sciences," *Concurrency and Computation - Practice and Experience*, vol. 18, no. 10, pp. 1067–1100, 2006. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/cpe.993/abstract

[57] I. Taylor, M. Shields, and I. Wang, "Distributed P2p computing within Triana: a galaxy visualization test case," in *Parallel and Distributed Processing Symposium, 2003. Proceedings. International*, Apr. 2003, p. 8 pp.

[58] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock, "Kepler: an extensible system for design and execution of scientific workflows," in *16th International Conference on Scientific and Statistical Database Management, 2004. Proceedings*, Jun. 2004, pp. 423 – 424.

[59] E. Deelman, G. Singh, M.-h. Su, J. Blythe, A. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz, "Pegasus: a framework for mapping complex scientific workflows onto distributed systems," *Scientific Programming Journal*, vol. 13, pp. 219–237, 2005.

[60] M. Berthold, N. Cebron, F. Dill, T. Gabriel, T. Ktter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel, "KNIME: The Konstanz Information Miner," in *Data Analysis, Machine Learning and Applications*. Springer Berlin Heidelberg, 2008, pp. 319–326. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-78246-9_38

[61] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Ktter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel, "KNIME - the Konstanz information miner: version 2.0 and beyond," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 26–31, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656280

[62] B. Giardine, C. Riemer, R. C. Hardison, R. Burhans, L. Elnitski, P. Shah, Y. Zhang, D. Blankenberg, I. Albert, J. Taylor, W. Miller, W. J. Kent, and A. Nekrutenko, "Galaxy: A Platform for Interactive Large-Scale Genome Analysis," *Genome Research*, vol. 15, no. 10, pp. 1451–1455, Oct. 2005. [Online]. Available: http://genome.cshlp.org/content/15/10/1451

[63] D. Blankenberg, G. V. Kuster, N. Coraor, G. Ananda, R. Lazarus, M. Mangan, A. Nekrutenko, and J. Taylor, "Galaxy: A Web-Based Genome Analysis Tool for Experimentalists," in *Current Protocols in Molecular Biology*. John Wiley & Sons, Inc., 2010. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/0471142727.mb1910s89/abstract

[64] J. Goecks, A. Nekrutenko, J. Taylor, and T. Galaxy Team, “Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences,” *Genome Biology*, vol. 11, no. 8, p. R86, 2010. [Online]. Available: http://genomebiology.com/2010/11/8/R86

[65] “Pipeline Pilot is Accelrys’ scientific informatics platform.” [Online]. Available: http://accelrys.com/products/pipeline-pilot/

[66] “IDBS - InforSense Suite - analytical data management.” [Online]. Available: http://www.idbs.com/products-and-services/inforsense-suite/

[67] R. Barga, J. Jackson, N. Araujo, D. Guo, N. Gautam, K. Grochow, and E. Lazowska, “Trident: Scientific Workflow Workbench for Oceanography,” in *Services, IEEE Congress on.* Los Alamitos, CA, USA: IEEE Computer Society, 2008, pp. 465–466.

[68] A. C. Siepel, A. N. Tolopko, A. D. Farmer, P. A. Steadman, F. D. Schilkey, B. D. Perry, and W. D. Beavis, “An integration platform for heterogeneous bioinformatics software components,” *IBM Syst. J.*, vol. 40, no. 2, pp. 570–591, Feb. 2001. [Online]. Available: http://dx.doi.org/10.1147/sj.402.0570

[69] R. Buyya and S. Venugopal, “The Gridbus Toolkit for Service Oriented Grid and Utility Computing: An Overview and Status Report,” in *Grid Economics and Business Models, 2004. GECON 2004. 1st IEEE International Workshop on.* IEEE, 2004, pp. 19–66, iEEE Xplore:. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=1317583&queryText%3DThe+Gridbus+Toolkit+for+Service+Oriented+Grid+and+Utility+Computing%3A+An+Overview+and+Status+Report

[70] N. Furmento, A. Mayer, S. McGough, S. Newhouse, T. Field, and J. Darlington, “ICENI: Optimisation of component applications within a Grid environment,” *Parallel Computing*, vol. 28, no. 12, pp. 1753–1772, Dec. 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167819102001874

[71] C. Walton, a. C. D. Walton, and A. D. Barker, “An Agent-based e-Science Experiment Builder,” in *In Proceedings of the 1st International Workshop on Semantic Intelligent Middleware for the Web and the Grid*, 2004.

[72] J. Brown, C. Ferner, T. Hudson, A. Stapleton, R. Vetter, T. Carland, A. Martin, J. Martin, A. Rawls, W. Shipman, and M. Wood, “GridNexus: A Grid Services Scientific Workflow System,” *International Journal of Computer and Information Science (IJCIS)*, vol. 6, no. 2, 2005.

[73] T. Fahringer, R. Prodan, R. Duan, F. Nerieri, S. Podlipnig, J. Qin, M. Siddiqui, H.-L. Truong, A. Villazon, and M. Wieczorek, “ASKALON: A Grid Application Development and Computing Environment,” in *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, ser. GRID ’05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 122–131. [Online]. Available: http://dx.doi.org/10.1109/GRID.2005.1542733

[74] W. A. Warr, “Scientific workflow systems: Pipeline Pilot and KNIME,” *Journal of Computer-Aided Molecular Design*, vol. 26, no. 7, pp. 801–804, Jul. 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3414708/

[75] Y. Collette and P. Siarry, *Multiobjective optimization : principles and case studies*. Berlin; New York: Springer, 2003. [Online]. Available: http://www.springer.com/gp/book/9783540401827

[76] J. Handl, D. B. Kell, and J. Knowles, “Multiobjective Optimization in Bioinformatics and Computational Biology,” *IEEEACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 2, pp. 279–292, Apr. 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4196538

[77] F. M. Gimnez, P. M. M. Collazos, I. A. Moralejo, and C. A. Gimnez, “Multiobjective evolutionary algorithms: Pareto rankings,” in *VII Jornadas Zaragoza-Pau de Matemtica Aplicada y estadstica : Jaca (Huesca), 17-18 de septiembre de 2001, 2003, ISBN 84-96214-04-4, pgs. 27-36*. Prensas Universitarias de Zaragoza, 2003, pp. 27–36. [Online]. Available: https://documat.unirioja.es/servlet/articulo?codigo=867106&info=resumen&idioma=SPA

[78] C. A. Nicolaou, J. Apostolakis, and C. S. Pattichis, “De Novo Drug Design Using Multiobjective Evolutionary Graphs,” *Journal of Chemical Information and Modeling*, vol. 49, no. 2, pp. 295–307, Feb. 2009. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci800308h

[79] C. A. Coello, G. B. Lamont, and D. A. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed., ser. Genetic and Evolutionary Computation. New York: Springer, 2007. [Online]. Available: http://www.springer.com/computer/theoretical+computer+science/book/978-0-387-33254-3?cm_mmc=Google-_-Book%20Search-_-Springer-_-0

[80] J. D. Schaffer, “Multiple Objective Optimization with Vector Evaluated Genetic Algorithms,” in *Proceedings of the 1st International Conference on Genetic Algorithms*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1985, pp. 93–100. [Online]. Available: http://dl.acm.org/citation.cfm?id=645511.657079

[81] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, “Multiobjective evolutionary algorithms: A survey of the state of the art,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, Mar. 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210650211000058

[82] C. M. Fonseca and P. J. Fleming, “Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization,” in *Proceedings of the 5th International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA 1993, 1993, pp. 416 – 423.

[83] N. Srinivas and K. Deb, “Muiltiobjective optimization using nondominated sorting in genetic algorithms,” *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Sep. 1994. [Online]. Available: http://dx.doi.org/10.1162/evco.1994.2.3.221

[84] J. Horn, N. Nafpliotis, and D. Goldberg, “A niched Pareto genetic algorithm for multiobjective optimization,” in *, Proceedings of the First IEEE Conference on Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence*, 1994, pp. 82–87 vol.1.

[85] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, 2000.

[86] Q. Zhang and H. Li, “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[87] H. Li and Q. Zhang, “Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[88] Q. Zhang, W. Liu, and H. Li, “The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances,” in *IEEE Congress on Evolutionary Computation, 2009. CEC ’09*, 2009, pp. 203–208.

[89] A. J. Nebro and J. J. Durillo, “A Study of the Parallelization of the Multi-Objective Metaheuristic MOEA/D,” in *Learning and Intelligent Optimization*, ser. Lecture Notes in Computer Science, C. Blum and R. Battiti, Eds. Springer Berlin Heidelberg, Jan.

2010, no. 6073, pp. 303–317. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-13800-3_32

[90] P. Palmers, T. McConnaghy, M. Steyaert, and G. Gielen, "Massively multi-topology sizing of analog integrated circuits," in *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, 2009, pp. 706–711.

[91] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, ser. GECCO '10. New York, NY, USA: ACM, 2010, pp. 519–526. [Online]. Available: http://doi.acm.org/10.1145/1830483.1830577

[92] K. Miettinen, "Some Methods for Nonlinear Multi-objective Optimization," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, E. Zitzler, L. Thiele, K. Deb, C. A. C. Coello, and D. Corne, Eds. Springer Berlin Heidelberg, Jan. 2001, no. 1993, pp. 1–20. [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-44719-9_1

[93] T. Tanino, M. Tanaka, and C. Hojo, "An interactive multicriteria decision making method by using a genetic algorithm," in *Proceedings of the Second International Conference on Systems Science and Systems Engineering (ICSSSE93)*, 1993, pp. 381–386.

[94] G. W. Greenwood, X. Hu, and J. G. D'Ambrosio, "Fitness Functions for Multiple Objective Optimization Problems: Combining Preferences with Pareto Rankings," in *FOGA'96*, 1996, pp. 437–455.

[95] M. Sakawa and K. Kato, "An interactive fuzzy satisficing method for general multiobjective 01 programming problems through genetic algorithms with double strings based on a reference solution," *Fuzzy Sets and Systems*, vol. 125, no. 3, pp. 289–300, Feb. 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016501140100029X

[96] S. P. Phelps and M. Kksalan, "An Interactive Evolutionary Metaheuristic for Multiobjective Combinatorial Optimization," *Manage. Sci.*, vol. 49, no. 12, pp. 1726–1738, Dec. 2003. [Online]. Available: http://dx.doi.org/10.1287/mnsc.49.12.1726.25117

[97] J. Branke and K. Deb, "Integrating User Preferences into Evolutionary Multi-Objective Optimization," in *Knowledge Incorporation in Evolutionary Computation*, ser. Studies in Fuzziness and Soft Computing, D. Y. Jin, Ed. Springer Berlin Heidelberg, Jan. 2005, no. 167, pp. 461–477. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-44511-1_21

[98] K. Deb, J. Sundar, N. Udaya Bhaskara Rao, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 273–286, 2006. [Online]. Available: http://www.softcomputing.net/ijcir/vol2-issu3-paper4.pdf

[99] K. Deb and A. Kumar, "Interactive evolutionary multi-objective optimization and decision-making using reference direction method," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, ser. GECCO '07. New York, NY, USA: ACM, 2007, pp. 781–788. [Online]. Available: http://doi.acm.org/10.1145/1276958.1277116

[100] K. Deb and S. Chaudhuri, "I-EMO: An Interactive Evolutionary Multi-objective Optimization Tool," in *Pattern Recognition and Machine Intelligence*, ser. Lecture Notes in Computer Science, S. K. Pal, S. Bandyopadhyay, and S. Biswas, Eds. Springer Berlin Heidelberg, Jan. 2005, no. 3776, pp. 690–695. [Online]. Available: http://link.springer.com/chapter/10.1007/11590316_111

[101] H. Li and D. Landa-Silva, "Evolutionary Multi-objective Simulated Annealing with adaptive and competitive search direction," in *IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*, 2008, pp. 3311–3318.

[102] J. Sanchis, M. A. Martnez, and X. Blasco, “Integrated multiobjective optimization and a priori preferences using genetic algorithms,” *Information Sciences*, vol. 178, no. 4, pp. 931–951, Feb. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025507004513

[103] K. Deb, A. Sinha, P. Korhonen, and J. Wallenius, “An Interactive Evolutionary Multiobjective Optimization Method Based on Progressively Approximated Value Functions,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 723–739, 2010.

[104] L. Rachmawati and D. Srinivasan, “Multiobjective Evolutionary Algorithm With Controllable Focus on the Knees of the Pareto Front,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 810–824, 2009.

[105] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, “A Preference-Based Evolutionary Algorithm for Multi-Objective Optimization,” *Evolutionary Computation*, vol. 17, no. 3, pp. 411–436, Sep. 2009. [Online]. Available: http://dx.doi.org/10.1162/evco.2009.17.3.411

[106] E. Zitzler and S. Knzli, “Indicator-Based Selection in Multiobjective Search,” in *Parallel Problem Solving from Nature - PPSN VIII*, ser. Lecture Notes in Computer Science, X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervs, J. A. Bullinaria, J. E. Rowe, P. Tio, A. Kabn, and H.-P. Schwefel, Eds. Springer Berlin Heidelberg, Jan. 2004, no. 3242, pp. 832–842. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-30217-9_84

[107] M. Basseur and E. Zitzler, “A Preliminary Study on Handling Uncertainty in Indicator-Based Multiobjective Optimization,” in *Applications of Evolutionary Computing: EvoWorkshops 2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, and EvoSTOC, Budapest, Hungary, April 10-12, 2006. Proceedings*, F. Rothlauf, J. Branke, S. Cagnoni, E. Costa, C. Cotta, R. Drechsler, E. Lutton, P. Machado, J. H. Moore, J. Romero, G. D. Smith, G. Squillero, and H. Takagi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 727–739, dOI: 10.1007/11732242_71. [Online]. Available: http://dx.doi.org/10.1007/11732242_71

[108] D. Brockhoff and E. Zitzler, “Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods,” in *IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, 2007, pp. 2086–2093.

[109] J. Bader and E. Zitzler, “Robustness in Hypervolume-based Multiobjective Search,” Computer Engineering and Networks Laboratory, ETH Zurich, Technical TIK 317, 2010.

[110] ——, “HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization,” *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1162/EVCO_a_00009

[111] A. Lara, G. Sanchez, C. A. Coello Coello, and O. Schutze, “HCS: A New Local Search Strategy for Memetic Multiobjective Evolutionary Algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 112–132, 2010.

[112] H. Ishibuchi and T. Murata, “A multi-objective genetic local search algorithm and its application to flowshop scheduling,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 28, no. 3, pp. 392–403, 1998.

[113] A. Jaszkiewicz, “Do multiple-objective metaheuristics deliver on their promises? A computational experiment on the set-covering problem,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 133–143, 2003.

[114] B. Qian, L. Wang, D.-X. Huang, and X. Wang, “Multi-objective no-wait flowshop scheduling with a memetic algorithm based on differential evolution,” *Soft Computing*, vol. 13, no. 8-9, pp. 847–869, Jul. 2009. [Online]. Available: http://link.springer.com/article/10.1007/s00500-008-0350-8

[115] J. Chen, Q. Lin, and Z. Ji, "A hybrid immune multiobjective optimization algorithm," *European Journal of Operational Research*, vol. 204, no. 2, pp. 294–302, Jul. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221709007541

[116] W.-F. Leong and G. Yen, "PSO-Based Multiobjective Optimization With Dynamic Population Size and Adaptive Local Archives," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 5, pp. 1270–1293, 2008.

[117] R. Caballero, M. Gonzlez, F. M. Guerrero, J. Molina, and C. Paralera, "Solving a multiobjective location routing problem with a metaheuristic based on tabu search. Application to a real case in Andalusia," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1751–1763, Mar. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221705006429

[118] E. F. Wanner, F. G. Guimares, R. H. C. Takahashi, and P. J. Fleming, "Local search with quadratic approximations into memetic algorithms for optimization with multiple criteria," *Evol. Comput.*, vol. 16, no. 2, pp. 185–224, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1162/evco.2008.16.2.185

[119] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Use of biased neighborhood structures in multiobjective memetic algorithms," *Soft Comput.*, vol. 13, no. 8-9, pp. 795–810, Mar. 2009. [Online]. Available: http://dx.doi.org/10.1007/s00500-008-0352-6

[120] S. F. Adra, T. J. Dodd, I. A. Griffin, and P. J. Fleming, "Convergence Acceleration Operator for Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 825–847, Aug. 2009. [Online]. Available: http://eprints.whiterose.ac.uk/9817/

[121] Y. Wang, Z. Cai, G. Guo, and Y. Zhou, "Multiobjective Optimization and Hybrid Evolutionary Algorithm to Solve Constrained Optimization Problems," *Trans. Sys. Man Cyber. Part B*, vol. 37, no. 3, pp. 560–575, Jun. 2007. [Online]. Available: http://dx.doi.org/10.1109/TSMCB.2006.886164

[122] M. Delgado, M. P. Cuellar, and M. C. Pegalajar, "Multiobjective Hybrid Optimization and Training of Recurrent Neural Networks," *Trans. Sys. Man Cyber. Part B*, vol. 38, no. 2, pp. 381–403, Apr. 2008. [Online]. Available: http://dx.doi.org/10.1109/TSMCB.2007.912937

[123] P. Koduru, Z. Dong, S. Das, S. Welch, J. Roe, and E. Charbit, "A Multiobjective Evolutionary-Simplex Hybrid Approach for the Optimization of Differential Equation Models of Gene Networks," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 5, pp. 572–590, 2008.

[124] J. Knowles and D. Corne, "M-PAES: a memetic algorithm for multiobjective optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation, 2000*, vol. 1, 2000, pp. 325–332 vol.1.

[125] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 402–412, 2002.

[126] A. Caponio and F. Neri, "Integrating Cross-Dominance Adaptation in Multi-Objective Memetic Algorithms," in *Multi-Objective Memetic Algorithms*, ser. Studies in Computational Intelligence, C.-K. Goh, Y.-S. Ong, and K. C. Tan, Eds. Springer Berlin Heidelberg, Jan. 2009, no. 171, pp. 325–351. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-88051-6_15

[127] O. Soliman, L. T. Bui, and H. Abbass, "A Memetic Coevolutionary Multi-Objective Differential Evolution Algorithm," in *Multi-Objective Memetic Algorithms*, ser. Studies in Computational Intelligence, C.-K. Goh, Y.-S. Ong, and K. C. Tan, Eds. Springer Berlin Heidelberg, Jan. 2009, no. 171, pp. 369–388. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-88051-6_17

[128] H. Li and D. Landa-Silva, “An Elitist GRASP Metaheuristic for the Multi-objective Quadratic Assignment Problem,” in *Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization*, ser. EMO ’09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 481–494. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01020-0_38

[129] K. Deb, M. Mohan, and S. Mishra, “Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions,” *Evolutionary computation*, vol. 13, no. 4, pp. 501–525, 2005.

[130] K. Tan, Y. J. Yang, and C. Goh, “A distributed Cooperative coevolutionary algorithm for multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 527–549, 2006.

[131] C.-K. Goh and K. Chen Tan, “A Competitive-Cooperative Coevolutionary Paradigm for Dynamic Multiobjective Optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009.

[132] C. Goh, K. Tan, D. Liu, and S. Chiam, “A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design,” *European Journal of Operational Research*, vol. 202, no. 1, pp. 42–54, Apr. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221709003166

[133] E. Zitzler and L. Thiele, *Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*, 1999.

[134] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm,” Tech. Rep., 2001.

[135] B. Y. Qu and P. N. Suganthan, “Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection,” *Inf. Sci.*, vol. 180, no. 17, pp. 3170–3181, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2010.05.013

[136] J. D. Knowles and D. W. Corne, “Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy,” *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, Jun. 2000. [Online]. Available: http://dx.doi.org/10.1162/106365600568167

[137] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, “Combining convergence and diversity in evolutionary multiobjective optimization,” *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, Sep. 2002. [Online]. Available: http://dx.doi.org/10.1162/106365602760234108

[138] G. Yen and H. Lu, “Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 253–274, 2003.

[139] C. Coello, G. Pulido, and M. Lechuga, “Handling multiple objectives with particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

[140] S.-Z. Zhao and P. N. Suganthan, “Multi-objective evolutionary algorithm with ensemble of external archives,” *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 1, pp. 1713–1726, 2010.

[141] M. Gong, L. Jiao, H. Du, and L. Bo, “Multiobjective immune algorithm with nondominated neighbor-based selection,” *Evol. Comput.*, vol. 16, no. 2, pp. 225–255, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1162/evco.2008.16.2.225

[142] B. Soylu and M. Kksalan, “A Favorable Weight-Based Evolutionary Algorithm for Multiple Criteria Problems,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 2, pp. 191–205, 2010.

[143] Y.-N. Wang, L.-H. Wu, and X.-F. Yuan, “Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure,” *Soft Computing*, vol. 14, no. 3, pp. 193–209, Feb. 2010. [Online]. Available: http://link.springer.com/article/10.1007/s00500-008-0394-9

[144] B. Panigrahi, V. Ravikumar Pandi, S. Das, and S. Das, “Multiobjective fuzzy dominance based bacterial foraging algorithm to solve economic emission dispatch problem,” *Energy*, vol. 35, no. 12, pp. 4761–4770, Dec. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360544210004937

[145] D. Kundu, K. Suresh, S. Ghosh, S. Das, B. Panigrahi, and S. Das, “Multi-objective optimization with artificial weed colonies,” *Information Sciences*, vol. 181, no. 12, pp. 2441–2454, Jun. 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025510004809

[146] H. Fang, Q. Wang, Y.-C. Tu, and M. F. Horstemeyer, “An efficient non-dominated sorting method for evolutionary algorithms,” *Evol. Comput.*, vol. 16, no. 3, pp. 355–384, Sep. 2008. [Online]. Available: http://dx.doi.org/10.1162/evco.2008.16.3.355

[147] C. Shi, Z. Yan, Z. Shi, and L. Zhang, “A fast multi-objective evolutionary algorithm based on a tree structure,” *Appl. Soft Comput.*, vol. 10, no. 2, pp. 468–480, Mar. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2009.08.018

[148] M. Fleischer and M. Fleischer, “The Measure of Pareto Optima. Applications to Multi-objective Metaheuristics,” in *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003.* Springer, 2003, pp. 519–533.

[149] S. Huband, P. Hingston, and e. al, “An Evolution Strategy with Probabilistic Mutation for Multi-Objective Optimisation,” in *IN CEC 03.* IEEE Press, Piscataway NJ, 2003, pp. 2284–2291.

[150] B. Naujoks, N. Beume, and M. Emmerich, “Multi-objective optimisation using S-metric selection: application to three-dimensional solution spaces,” in *The 2005 IEEE Congress on Evolutionary Computation, 2005*, vol. 2, 2005, pp. 1282–1289 Vol. 2.

[151] C. Igel, N. Hansen, and S. Roth, “Covariance Matrix Adaptation for Multi-objective Optimization,” *Evol. Comput.*, vol. 15, no. 1, pp. 1–28, Mar. 2007. [Online]. Available: http://dx.doi.org/10.1162/evco.2007.15.1.1

[152] A. Iorio and X. Li, *Rotationally Invariant Crossover Operators in Evolutionary Multi-objective Optimization*, 2008.

[153] S. Y. Zeng, L. S. Kang, and L. X. Ding, “An orthogonal multi-objective evolutionary algorithm for multi-objective optimization problems with constraints,” *Evolutionary computation*, vol. 12, no. 1, pp. 77–98, 2004.

[154] K. Weinert, A. Zabel, P. Kersting, T. Michelitsch, and T. Wagner, “On the use of problem-specific candidate generators for the hybrid optimization of multi-objective production engineering problems,” *Evol. Comput.*, vol. 17, no. 4, pp. 527–544, Dec. 2009. [Online]. Available: http://dx.doi.org/10.1162/evco.2009.17.4.17405

[155] Q. Zhang, A. Zhou, and Y. Jin, *RM-MEDA: A Regularity Model Based Multiobjective Estimation of Distribution Algorithm*, 2008.

[156] A. Zhou, Q. Zhang, and Y. Jin, “Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm,” *Trans. Evol. Comp*, vol. 13, no. 5, pp. 1167–1189, Oct. 2009. [Online]. Available: http://dx.doi.org/10.1109/TEVC.2009.2021467

[157] R. Storn and K. Price, *Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces*, 1995.

[158] K. V. Price, “An introduction to differential evolution,” in *New ideas in optimization*, D. Corne, M. Dorigo, F. Glover, D. Dasgupta, P. Moscato, R. Poli, and K. V. Price, Eds. Maidenhead, UK, England: McGraw-Hill Ltd., UK, 1999, pp. 79–108. [Online]. Available: http://dl.acm.org/citation.cfm?id=329055.329069

[159] R. Sarker and H. A. Abbass, “Differential Evolution for Solving Multi-Objective Optimization Problems,” in *no.2, World Scientific*, 2004, pp. 225–240.

[160] W. Gong and Z. Cai, “An improved multiobjective differential evolution based on Pareto-adaptive [epsilon]-dominance and orthogonal design,” *European Journal of Operational Research*, vol. 198, no. 2, pp. 576–601, 2009. [Online]. Available: http://ideas.repec.org/a/eee/ejores/v198y2009i2p576-601.html

[161] B. Alatas, E. Akin, and A. Karci, “MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules,” *Applied Soft Computing*, vol. 8, no. 1, pp. 646–656, Jan. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S156849460700049X

[162] T. Fukuda, K. Mori, and M. Tsukiyama, “Immune networks using genetic algorithm for adaptive production scheduling,” *15th IFAC World Congress*, vol. 3, 1993. [Online]. Available: http://portal.acm.org/citation.cfm?id=903795&dl=GUIDE&coll=GUIDE

[163] C. A. Coello and N. C. Corts, “Solving Multiobjective Optimization Problems Using an Artificial Immune System,” *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, Jun. 2005. [Online]. Available: http://dx.doi.org/10.1007/s10710-005-6164-x

[164] R. Tavakkoli-Moghaddam, A. Rahimi-Vahed, and A. H. Mirzaei, “A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: Weighted mean completion time and weighted mean tardiness,” *Information Sciences*, vol. 177, no. 22, pp. 5072–5090, Nov. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025507002861

[165] Z.-H. Hu, “A multiobjective immune algorithm based on a multiple-affinity model,” *European Journal of Operational Research*, vol. 202, no. 1, pp. 60–72, Apr. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221709003464

[166] Z. Zhang, “Immune optimization algorithm for constrained nonlinear multiobjective optimization problems,” *Appl. Soft Comput.*, vol. 7, no. 3, pp. 840–857, Jun. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2006.02.008

[167] X. Zuo, H. Mo, and J. Wu, “A robust scheduling method based on a multi-objective immune algorithm,” *Inf. Sci.*, vol. 179, no. 19, pp. 3359–3369, Sep. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2009.06.003

[168] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995. MHS ’95*, 1995, pp. 39–43.

[169] J. F. Kennedy, J. Kennedy, and R. C. Eberhart, *Swarm Intelligence*. Morgan Kaufmann, 2001.

[170] J. Moore and R. Chapman, “Application Of Particle Swarm To Multiobjective Optimization,” 1999. [Online]. Available: http://natcomp.liacs.nl/SWI/papers/particle.swarm.optimization/moore99application.pdf

[171] S. Janson, D. Merkle, and M. Middendorf, "Molecular docking with multi-objective Particle Swarm Optimization," *Applied Soft Computing*, vol. 8, no. 1, pp. 666–675, Jan. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494607000518

[172] D. S. Liu, K. C. Tan, S. Y. Huang, C. K. Goh, and W. K. Ho, "On solving multiobjective bin packing problems using evolutionary particle swarm optimization," *European Journal of Operational Research*, vol. 190, no. 2, pp. 357–382, Oct. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037722170700567X

[173] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, "Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients," *Information Sciences*, vol. 177, no. 22, pp. 5033–5049, Nov. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025507003155

[174] Y. Wang and Y. Yang, "Particle swarm optimization with preference order ranking for multi-objective optimization," *Information Sciences*, vol. 179, no. 12, pp. 1944–1959, May 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025509000176

[175] A. Elhossini, S. Areibi, and R. Dony, "Strength pareto particle swarm optimization and hybrid ea-pso for multi-objective optimization," *Evol. Comput.*, vol. 18, no. 1, pp. 127–156, Mar. 2010. [Online]. Available: http://dx.doi.org/10.1162/evco.2010.18.1.18105

[176] A. R. Rahimi-Vahed, S. M. Mirghorbani, and M. Rabbani, "A new particle swarm algorithm for a multi-objective mixed-model assembly line sequencing problem," *Soft Computing*, vol. 11, no. 10, pp. 997–1012, Feb. 2007. [Online]. Available: http://link.springer.com/article/10.1007/s00500-007-0149-z

[177] S. Agrawal, B. K. Panigrahi, and M. K. Tiwari, "Multiobjective Particle Swarm Algorithm With Fuzzy Clustering for Electrical Power Dispatch," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 5, pp. 529–541, Oct. 2008.

[178] V. Huang, P. Suganthan, and J. Liang, "Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems," *International Journal of Intelligent Systems*, vol. 21, no. 2, pp. 209–226, Feb. 2006. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/int.20128/abstract

[179] S. Z. Zhao and P. N. Suganthan, "Two- lbests based multi-objective particle swarm optimizer," *Engineering Optimization*, vol. 43, no. 1, pp. 1–17, Jan. 2011. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/03052151003686716

[180] N. A. Moubayed, A. Petrovski, and J. McCall, "A Novel Smart Multi-Objective Particle Swarm Optimisation Using Decomposition," in *Parallel Problem Solving from Nature, PPSN XI*, ser. Lecture Notes in Computer Science, R. Schaefer, C. Cotta, J. Koodziej, and G. Rudolph, Eds. Springer Berlin Heidelberg, Sep. 2010, no. 6239, pp. 1–10, dOI: 10.1007/978-3-642-15871-1_1. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-15871-1_1

[181] M. Reyes-Sierra and C. Coello, "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, 2006.

[182] M. Dorigo and T. Sttzle, *Ant Colony Optimization*. Scituate, MA, USA: Bradford Company, 2004.

[183] D. Angus, "Crowding Population-based Ant Colony Optimisation for the Multi-objective Travelling Salesman Problem," in *IEEE Symposium on Computational Intelligence in Multicriteria Decision Making*, Apr. 2007, pp. 333–340.

[184] C. Garca-Martnez, O. Cordn, and F. Herrera, “A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP,” *European Journal of Operational Research*, vol. 180, no. 1, pp. 116–148, Jul. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221706002451

[185] D. M. Chitty and M. L. Hernandez, “A Hybrid Ant Colony Optimisation Technique for Dynamic Vehicle Routing,” in *Genetic and Evolutionary Computation GECCO 2004*, ser. Lecture Notes in Computer Science, K. Deb, Ed. Springer Berlin Heidelberg, Jun. 2004, no. 3102, pp. 48–59, dOI: 10.1007/978-3-540-24854-5_5. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-24854-5_5

[186] J. M. Pasia, R. F. Hartl, and K. F. Doerner, “Solving a Bi-objective Flowshop Scheduling Problem by Pareto-ant Colony Optimization,” in *Proceedings of the 5th International Conference on Ant Colony Optimization and Swarm Intelligence*, ser. ANTS’06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 294–305. [Online]. Available: http://dx.doi.org/10.1007/11839088_26

[187] K. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer, “Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection,” *Annals of Operations Research*, vol. 131, no. 1-4, pp. 79–99, Oct. 2004. [Online]. Available: http://link.springer.com/article/10.1023/B:ANOR.0000039513.99038.c6

[188] K. F. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer, “Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection,” *European Journal of Operational Research*, vol. 171, no. 3, pp. 830–841, Jun. 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221704005880

[189] R. Y. Rubinstein and D. P. Kroese, *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-carlo Simulation (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2004.

[190] A. Uenveren and A. Acan, “Multi-objective optimization with cross entropy method: Stochastic learning with clustered pareto fronts,” in *2007 IEEE Congress on Evolutionary Computation*, Sep. 2007, pp. 3065–3071.

[191] K.-H. Han and J.-H. Kim, “Genetic quantum algorithm and its application to combinatorial optimization problem,” in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, vol. 2, 2000, pp. 1354–1360 vol.2.

[192] W. Wei, B. Li, Y. Zou, W. Zhang, and Z. Zhuang, “A multi-objective hwsw co-synthesis algorithm based on quantum-inspired evolutionary algorithm,” *International Journal of Computational Intelligence and Applications*, vol. 07, no. 02, pp. 129–148, Jun. 2008. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S146902680800220X

[193] B. B. Li and L. Wang, “A Hybrid Quantum-Inspired Genetic Algorithm for Multiobjective Flow Shop Scheduling,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 3, pp. 576–591, Jun. 2007.

[194] H. Mhlenbein and G. Paa, “From recombination of genes to the estimation of distributions I. Binary parameters,” in *Parallel Problem Solving from Nature PPSN IV*, ser. Lecture Notes in Computer Science, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Springer Berlin Heidelberg, Sep. 1996, pp. 178–187, dOI: 10.1007/3-540-61723-X_982. [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-61723-X_982

[195] P. Larraaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms - A New Tool for Evolutionary Computation*. Springer US, 2002. [Online]. Available: http://www.springer.com/gp/book/9780792374664

[196] T. Okabe, Y. Jin, B. Sendoff, and M. Olhofer, “Voronoi-based estimation of distribution algorithm for multi-objective optimization,” in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, vol. 2, Jun. 2004, pp. 1594–1601 Vol.2.

[197] P. A. N. Bosman and D. Thierens, “The Naive MIDEA: A Baseline Multi-objective EA,” in *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, ser. EMO’05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 428–442. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-31880-4_30

[198] W. Dong and X. Yao, “Unified Eigen Analysis on Multivariate Gaussian Based Estimation of Distribution Algorithms,” *Inf. Sci.*, vol. 178, no. 15, pp. 3000–3023, Aug. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2008.01.021

[199] M. Laumanns and J. Ocenasek, “Bayesian Optimization Algorithms for Multi-objective Optimization,” in *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, ser. PPSN VII. London, UK, UK: Springer-Verlag, 2002, pp. 298–307. [Online]. Available: http://dl.acm.org/citation.cfm?id=645826.669574

[200] M. Pelikan, K. Sastry, and D. E. Goldberg, “Multiobjective hBOA, Clustering, and Scalability,” in *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO ’05. New York, NY, USA: ACM, 2005, pp. 663–670. [Online]. Available: http://doi.acm.org/10.1145/1068009.1068122

[201] K. Miettinen, *Nonlinear Multiobjective Optimization*. Springer US, 1998. [Online]. Available: http://www.springer.com/gp/book/9780792382782

[202] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, “Prediction-Based Population Re-initialization for Evolutionary Dynamic Multi-objective Optimization,” in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Springer Berlin Heidelberg, Mar. 2007, pp. 832–846, dOI: 10.1007/978-3-540-70928-2_62. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-70928-2_62

[203] A. Zhou, Q. Zhang, Y. Jin, B. Sendhoff, and E. Tsang, “Global Multiobjective Optimization via Estimation of Distribution Algorithm with Biased Initialization and Crossover,” in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO ’07. New York, NY, USA: ACM, 2007, pp. 617–623. [Online]. Available: http://doi.acm.org/10.1145/1276958.1277082

[204] Y. Jin, A. Zhou, Q. Zhang, B. Sendhoff, and E. Tsang, “Modeling Regularity to Improve Scalability of Model-Based Multiobjective Optimization Algorithms,” in *Multiobjective Problem Solving from Nature*, ser. Natural Computing Series, J. Knowles, D. Corne, K. Deb, and D. R. Chair, Eds. Springer Berlin Heidelberg, 2008, pp. 331–355, dOI: 10.1007/978-3-540-72964-8_16. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-72964-8_16

[205] L. Mo, G. Dai, and J. Zhu, “The RM-MEDA Based on Elitist Strategy,” in *Advances in Computation and Intelligence*, ser. Lecture Notes in Computer Science, Z. Cai, C. Hu, Z. Kang, and Y. Liu, Eds. Springer Berlin Heidelberg, Oct. 2010, pp. 229–239, dOI: 10.1007/978-3-642-16493-4_24. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-16493-4_24

[206] A. K. A. Talukder, M. Kirley, and R. Buyya, “A Pareto Following Variation Operator for Fast-converging Multiobjective Evolutionary Algorithms,” in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO ’08. New York, NY, USA: ACM, 2008, pp. 721–728. [Online]. Available: http://doi.acm.org/10.1145/1389095.1389234

[207] D. Yang, L. Jiao, M. Gong, and H. Feng, “Hybrid multiobjective estimation of distribution algorithm by local linear embedding and an immune inspired algorithm,” in *2009 IEEE Congress on Evolutionary Computation*, May 2009, pp. 463–470.

[208] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, May 1983. [Online]. Available: http://science.sciencemag.org/content/220/4598/671

[209] L. Snchez and J. R. Villar, “Obtaining Transparent Models of Chaotic Systems with Multi-objective Simulated Annealing Algorithms,” *Inf. Sci.*, vol. 178, no. 4, pp. 952–970, Feb. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2007.09.029

[210] K. I. Smith, R. M. Everson, J. E. Fieldsend, C. Murphy, and R. Misra, “Dominance-Based Multiobjective Simulated Annealing,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 323–342, Jun. 2008.

[211] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, “A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 269–283, Jun. 2008.

[212] E. Aggelogiannaki and H. Sarimveis, “A Simulated Annealing Algorithm for Prioritized Multiobjective Optimization mdash;Implementation in an Adaptive Model Predictive Control Configuration,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 4, pp. 902–915, Aug. 2007.

[213] L. Belfares, W. Klibi, N. Lo, and A. Guitouni, “Multi-objectives Tabu Search based algorithm for progressive resource allocation,” *European Journal of Operational Research*, vol. 177, no. 3, pp. 1779–1799, Mar. 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221705006466

[214] R. P. Beausoleil, “MOSS multiobjective scatter search applied to non-linear multiple criteria optimization,” *European Journal of Operational Research*, vol. 169, no. 2, pp. 426–449, Mar. 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221704005478

[215] A. P. Reynolds and B. d. l. Iglesia, “A multi-objective GRASP for partial classification,” *Soft Computing*, vol. 13, no. 3, pp. 227–243, Feb. 2009. [Online]. Available: http://link.springer.com/article/10.1007/s00500-008-0320-1

[216] I. Rechenberg and H. P. Schwefel, *Adaptive Mechanismen in der Biologischen Evolution und ihr Einfluss auf die Evolutionsgeschwindigkeit: Arbeitsbericht.*, 1974, oCLC: 258504612.

[217] D. B. Fogel, L. J. Fogel, and J. W. Atmar, “Meta-evolutionary programming,” in *[1991] Conference Record of the Twenty-Fifth Asilomar Conference on Signals, Systems Computers*, Nov. 1991, pp. 540–545 vol.1.

[218] K. Deb and H. Jain, “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[219] H. Jain and K. Deb, “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

[220] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, “Many-Objective Optimization: An Engineering Design Perspective,” in *Evolutionary Multi-Criterion Optimization*, ser.

Lecture Notes in Computer Science, C. A. C. Coello, A. H. Aguirre, and E. Zitzler, Eds. Springer Berlin Heidelberg, Jan. 2005, no. 3410, pp. 14–32. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-31880-4_2

[221] C.-L. Hwang and K. Yoon, *Multiple Attribute Decision Making - Methods and Applications: A Stae-of-the-Art Survey*, 1981. [Online]. Available: http://www.springer.com/gp/book/9783540105589

[222] K. Yoon, "A Reconciliation Among Discrete Compromise Solutions," *Journal of the Operational Research Society*, vol. 38, no. 3, pp. 277–286, Mar. 1987. [Online]. Available: http://link.springer.com/article/10.1057/jors.1987.44

[223] C.-L. Hwang, Y.-J. Lai, and T.-Y. Liu, "A new approach for multiple objective decision making," *Computers & Operations Research*, vol. 20, no. 8, pp. 889–899, Oct. 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/030505489390109V

[224] G. Landrum, "RDKit: Open-source cheminformatics." [Online]. Available: http://www.rdkit.org/

[225] R. Barone and M. Chanon, "A New and Simple Approach to Chemical Complexity. Application to the Synthesis of Natural Products," *Journal of Chemical Information and Computer Sciences*, vol. 41, no. 2, pp. 269–272, Mar. 2001. [Online]. Available: http://dx.doi.org/10.1021/ci000145p

[226] H. L. Morgan, "The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service." *Journal of Chemical Information and Modeling*, vol. 5, no. 2, pp. 107–113, 1965. [Online]. Available: http://dx.doi.org/10.1021/c160017a018

[227] J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse, "Reoptimization of MDL Keys for Use in Drug Discovery," *Journal of Chemical Information and Computer Sciences*, vol. 42, no. 6, pp. 1273–1280, Nov. 2002. [Online]. Available: http://dx.doi.org/10.1021/ci010132r

[228] R. E. Carhart, D. H. Smith, and R. Venkataraghavan, "Atom pairs as molecular features in structure-activity studies: definition and applications," *Journal of Chemical Information and Computer Sciences*, vol. 25, no. 2, pp. 64–73, May 1985. [Online]. Available: http://dx.doi.org/10.1021/ci00046a002

[229] R. Nilakantan, N. Bauman, J. S. Dixon, and R. Venkataraghavan, "Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors," *Journal of Chemical Information and Computer Sciences*, vol. 27, no. 2, pp. 82–85, May 1987. [Online]. Available: http://dx.doi.org/10.1021/ci00054a008

[230] X. Lewell, D. Judd, S. Watson, and M. Hann, "RECAP-Retrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry," *Journal of Chemical Information and Modeling*, vol. 38, no. 3, pp. 511–522, May 1998. [Online]. Available: http://pubs.acs.org/cgi-bin/doilookup/?10.1021/ci970429i

[231] G. W. Bemis and M. A. Murcko, "The Properties of Known Drugs. 1. Molecular Frameworks," *Journal of Medicinal Chemistry*, vol. 39, no. 15, pp. 2887–2893, Jan. 1996. [Online]. Available: http://dx.doi.org/10.1021/jm9602928

[232] O. Trott and A. J. Olson, "AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading," *Journal of Computational Chemistry*, vol. 31, no. 2, pp. 455–461, Jan. 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19499576

[233] M. Kuhn, “Building Predictive Models in R Using the caret Package,” *Journal of Statistical Software*, vol. 28, no. 5, pp. 1–26, 2008. [Online]. Available: http://www.jstatsoft.org/v28/i05

[234] N. M. O’Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison, “Open Babel: An open chemical toolbox,” *Journal of Cheminformatics*, vol. 3, no. 1, p. 33, Oct. 2011. [Online]. Available: http://www.jcheminf.com/content/3/1/33/abstract

[235] N. M. O’Boyle, C. Morley, and G. R. Hutchison, “Pybel: a Python wrapper for the OpenBabel cheminformatics toolkit,” *Chemistry Central Journal*, vol. 2, p. 5, 2008. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18328109

[236] R. Development Core Team, *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2008, iSBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

[237] J. L. Medina-Franco, F. Lpez-Vallejo, D. Kuck, and F. Lyko, “Natural products as DNA methyltransferase inhibitors: a computer-aided discovery approach,” *Molecular Diversity*, vol. 15, pp. 293–304, Aug. 2010. [Online]. Available: http://www.springerlink.com/index/10.1007/s11030-010-9262-5

[238] C. Kannas, I. Kalvari, G. Lambrinidis, C. Neophytou, C. Savva, I. Kirmitzoglou, Z. Antoniou, K. Achilleos, D. Scherf, C. Pitta, C. Nicolaou, E. Mikros, V. Promponas, C. Gerhauser, R. Mehta, A. Constantinou, and C. Pattichis, “LiSIs: An Online Scientific Workflow System for Virtual Screening,” *Combinatorial Chemistry & High Throughput Screening*, vol. 18, no. 3, pp. 281 – 295, Mar. 2015. [Online]. Available: http://www.eurekaselect.com/openurl/content.php?genre=article&doi=10.2174/1386207318666150305123341

[239] B. M. Collins-Burow, M. E. Burow, B. N. Duong, and J. A. McLachlan, “Estrogenic and Antiestrogenic Activities of FlavonoidPhytochemicals Through Estrogen ReceptorBinding-Dependent and -Independent Mechanisms,” *Nutrition and Cancer*, vol. 38, no. 2, pp. 229–244, 2000. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1207/S15327914NC382_13

[240] Y. Jacquot, I. Laos, A. Cleeren, D. Nonclercq, L. Bermont, B. Refouvelet, K. Boubekeur, A. Xicluna, G. Leclercq, and G. Laurent, “Synthesis, structure, and estrogenic activity of 4-amino-3-(2-methylbenzyl)coumarins on human breast carcinoma cells,” *Bioorganic & Medicinal Chemistry*, vol. 15, no. 6, pp. 2269–2282, Mar. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968089607000430

[241] J. R. Gunther, Y. Du, E. Rhoden, I. Lewis, B. Revennaugh, T. W. Moore, S. H. Kim, R. Dingledine, H. Fu, and J. A. Katzenellenbogen, “A Set of Time-Resolved Fluorescence Resonance Energy Transfer Assays for the Discovery of Inhibitors of Estrogen Receptor-Coactivator Binding,” *Journal of biomolecular screening*, vol. 14, no. 2, pp. 181–193, Feb. 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2731238/

[242] H. Gurer-Orhan, J. Kool, N. P. E. Vermeulen, and J. H. N. Meerman, “A novel microplate reader-based high-throughput assay for estrogen receptor binding,” *International Journal of Environmental Analytical Chemistry*, vol. 85, no. 3, pp. 149–161, 2005. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/03067310500042236

[243] G. M. Anstead, K. E. Carlson, and J. A. Katzenellenbogen, “The estradiol pharmacophore: Ligand structure-estrogen receptor binding affinity relationships and a model for the receptor binding site,” *Steroids*, vol. 62, no. 3, pp. 268–303, Mar. 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0039128X96002425

[244] S. Agatonovic-Kustrin and J. V. Turner, “Molecular Structural Characteristics of Estrogen Receptor Modulators as Determinants of Estrogen Receptor Selectivity,” *Mini-Reviews*

in Medicinal Chemistry, vol. 8, no. 9, pp. 943–951, Aug. 2008. [Online]. Available: http://www.eurekaselect.com/83047/article

[245] C. A. Nicolaou, C. Kannas, and C. S. Pattichis, "Optimal graph design using a knowledge-driven multi-objective evolutionary graph algorithm," in *2009 9th International Conference on Information Technology and Applications in Biomedicine*. Larnaka, Cyprus: IEEE, Nov. 2009, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5394397

[246] M. Feher, Y. Gao, J. C. Baber, W. A. Shirley, and J. Saunders, "The use of ligand-based de novo design for scaffold hopping and sidechain optimization: two case studies," *Bioorganic & Medicinal Chemistry*, vol. 16, no. 1, pp. 422–427, Jan. 2008. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17920281

[247] F. Dey and A. Caflisch, "Fragment-based de novo ligand design by multiobjective evolutionary optimization," *Journal of Chemical Information and Modeling*, vol. 48, no. 3, pp. 679–690, Mar. 2008. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18307332

[248] P. S. Kutchukian, D. Lou, and E. I. Shakhnovich, "FOG: Fragment Optimized Growth algorithm for the de novo generation of molecules occupying druglike chemical space," *Journal of Chemical Information and Modeling*, vol. 49, no. 7, pp. 1630–1642, Jul. 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19527020

[249] S. Ekins, J. D. Honeycutt, and J. T. Metz, "Evolving molecules using multi-objective optimization: applying to ADME/Tox," *Drug Discovery Today*, vol. 15, no. 11-12, pp. 451–460, Jun. 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20438859

[250] J. R. Damewood, Jr, C. L. Lerman, and B. B. Masek, "NovoFLAP: A ligand-based de novo design approach for the generation of medicinally relevant ideas," *Journal of Chemical Information and Modeling*, vol. 50, no. 7, pp. 1296–1303, Jul. 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20586434

[251] Q. Huang, L.-L. Li, and S.-Y. Yang, "PhDD: a new pharmacophore-based de novo design method of drug-like molecules combined with assessment of synthetic accessibility," *Journal of Molecular Graphics and Modelling*, vol. 28, no. 8, pp. 775–787, Jun. 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20206562

[252] M. Hartenfeller, H. Zettl, M. Walter, M. Rupp, F. Reisen, E. Proschak, S. Weggen, H. Stark, and G. Schneider, "DOGS: Reaction-Driven de novo Design of Bioactive Compounds," *PLoS Comput Biol*, vol. 8, no. 2, p. e1002380, Feb. 2012. [Online]. Available: http://dx.doi.org/10.1371/journal.pcbi.1002380

[253] A. R. Beccari, C. Cavazzoni, C. Beato, and G. Costantino, "LiGen: A High Performance Workflow for Chemistry Driven de Novo Design," *Journal of Chemical Information and Modeling*, Apr. 2013. [Online]. Available: http://dx.doi.org/10.1021/ci400078g

[254] N. C. Firth, B. Atrash, N. Brown, and J. Blagg, "MOARF, an Integrated Workflow for Multiobjective Optimization: Implementation, Synthesis, and Biological Evaluation," *Journal of Chemical Information and Modeling*, vol. 55, no. 6, pp. 1169 – 1180, Jun. 2015. [Online]. Available: http://dx.doi.org/10.1021/acs.jcim.5b00073

[255] F. Daeyaert and M. W. Deem, "A Pareto Algorithm for Efficient De Novo Design of Multi-functional Molecules," *Molecular Informatics*, vol. 36, no. 1-2, Jul. 2016. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/minf.201600044/abstract

[256] C. A. Nicolaou and C. C. Kannas, "Molecular Library Design Using Multi-Objective Optimization Methods," in *Chemical Library Design*, ser. Methods in Molecular Biology, J. Z. Zhou, Ed. Humana Press, Jan. 2011, no. 685, pp. 53–69. [Online]. Available: http://link.springer.com/protocol/10.1007/978-1-60761-931-4_3

[257] C. A. Nicolaou, N. Brown, and C. S. Pattichis, “Molecular optimization using computational multi-objective methods,” *Current Opinion in Drug Discovery & Development*, vol. 10, no. 3, pp. 316–324, May 2007. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17554858

[258] C. A. Nicolaou, C. Kannas, and E. Loizidou, “Multi-objective optimization methods in de novo drug design,” *Mini reviews in medicinal chemistry*, vol. 12, no. 10, pp. 979–987, Sep. 2012.

[259] C. A. Nicolaou and N. Brown, “Multi-objective optimization methods in drug design,” *Drug Discovery Today: Technologies*, vol. 10, no. 3, pp. e427–e435, Sep. 2013. [Online]. Available: https://ucyvpn.ucy.ac.cy/+CSCO+00756767633A2F2F6A6A6A2E667076726170727176657270672E70627A++/science/article/pii/S1740674913000085

[260] C. A. Nicolaou, “Graph Design using Knowlwdge-Driven, Self-Adaptive Multi-Objective Evolutionary Graph Algorithms,” Ph.D. Thesis, University of Cyprus, Jun. 2010.

[261] “Welcome to Python.org.” [Online]. Available: https://www.python.org/

[262] E. Alba, “Parallel evolutionary algorithms can achieve super-linear performance,” *Information Processing Letters*, vol. 82, no. 1, pp. 7–13, Apr. 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020019001002812

[263] “Evolutionary Algorithms 8 Population models - Parallel implementations.” [Online]. Available: http://www.geatbx.com/docu/algindex-07.html

[264] P. Adamidis, “Parallel Evolutionary Algorithms: A Review,” in *4th Hellenic-European Conference on Computer Mathematics and its Applications*, Athens, Greece, Sep. 1998.

[265] M. Tomassini, *Parallel and Distributed Evolutionary Algorithms: A Review*, 1999.

[266] C. C. Kannas, C. A. Nicolaou, and C. S. Pattichis, “A Parallel implementation of a Multi-objective Evolutionary Algorithm,” in *2009 9th International Conference on Information Technology and Applications in Biomedicine*. Larnaka, Cyprus: IEEE, Nov. 2009, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5394393

[267] C. Nicolaou, C. Kannas, and C. Pattichis, “Knowledge-driven multi-objective de novo drug design,” *Chemistry Central Journal*, vol. 3, p. P22, 2009. [Online]. Available: http://www.journal.chemistrycentral.com/content/3/S1/P22

[268] “Cauchy Distribution.” [Online]. Available: http://www.math.uah.edu/stat/special/Cauchy.html

[269] J. W. Raymond, E. J. Gardiner, and P. Willett, “RASCAL: Calculation of Graph Similarity using Maximum Common Edge Subgraphs,” *The Computer Journal*, vol. 45, no. 6, pp. 631–644, Jan. 2002. [Online]. Available: http://comjnl.oxfordjournals.org/content/45/6/631

[270] R. P. Sheridan, M. D. Miller, D. J. Underwood, and S. K. Kearsley, “Chemical Similarity Using Geometric Atom Pair Descriptors,” *Journal of Chemical Information and Computer Sciences*, vol. 36, no. 1, pp. 128–136, Jan. 1996. [Online]. Available: http://dx.doi.org/10.1021/ci950275b

[271] S. K. Kearsley, S. Sallamack, E. M. Fluder, J. D. Andose, R. T. Mosley, and R. P. Sheridan, “Chemical Similarity Using Physiochemical Property Descriptors,” *Journal of Chemical Information and Computer Sciences*, vol. 36, no. 1, pp. 118–127, Jan. 1996. [Online]. Available: http://dx.doi.org/10.1021/ci950274j

[272] “Distance Measures.” [Online]. Available: http://www.sequentix.de/gelquest/help/distance_measures.htm

[273] S. M. v. Liempd, J. Kool, W. M. A. Niessen, D. E. v. Elswijk, H. Irth, and N. P. E. Vermeulen, “On-line Formation, Separation, and Estrogen Receptor Affinity Screening of Cytochrome P450-Derived Metabolites of Selective Estrogen Receptor Modulators,” *Drug Metabolism and Disposition*, vol. 34, no. 9, pp. 1640–1649, Sep. 2006. [Online]. Available: http://dmd.aspetjournals.org/content/34/9/1640

[274] G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell, and A. J. Olson, “AutoDock4 and AutoDockTools4: Automated Docking with Selective Receptor Flexibility,” *Journal of computational chemistry*, vol. 30, no. 16, pp. 2785–2791, Dec. 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2760638/

Appendix A

LIST OF PUBLICATIONS

Book Chapters

C. A. Nicolaou and C. C. Kannas, ”**Molecular Library Design Using Multi-Objective Optimization Methods**,” in *Chemical Library Design*, J. Z. Zhou, Ed. Humana Press, 2011, pp. 53-69.

Journals

C. A. Nicolaou, C. Kannas, and E. Loizidou, ”**Multi-objective optimization methods in de novo drug design**,” *Mini Rev Med Chem*, vol. 12, no. 10, pp. 979-987, Sep. 2012.

C. Kannas, I. Kalvari, G. Lambrinidis, C. Neophytou, C. Savva, I. Kirmitzoglou, Z. Antoniou, K. Achilleos, D. Scherf, C. Pitta, C. Nicolaou, E. Mikros, V. Promponas, C. Gerhauser, R. Mehta, A. Constantinou, C. Pattichis, ”**LiSIs: An Online Scientific Workflow System for Virtual Screening**,” *Combinatorial Chemistry & High Throughput Screening*, vol. 18, no. 3, pp. 281-295, Mar. 2015.

C. C. Kannas, E. Z. Loizidou and C. S. Pattichis, ”**Self-Adaptive Multi-Objective Evolutionary Algorithm for Molecular Design**,” in *IEEE Transactions on Evolutionary Computation*, September 2017, (to be submitted).

Conference Papers

C. C. Kannas, C. A. Nicolaou, and C. S. Pattichis, ”**A Parallel implementation of a Multi-objective Evolutionary Algorithm**,” in *2009 9th International Conference on Information Technology and Applications in Biomedicine*, Larnaka, Cyprus, 2009, pp. 1-6.

C. A. Nicolaou, C. Kannas, and C. S. Pattichis, ”**Optimal graph design using a knowledge-driven multi-objective evolutionary graph algorithm**,” in *2009 9th International Conference on Information Technology and Applications in Biomedicine*, Larnaka, Cyprus, 2009, pp. 1-6.

K. G. Achilleos, C. C. Kannas, C. A. Nicolaou, C. S. Pattichis, and V. J. Promponas, ”**Open source workflow systems in life sciences informatics**,” in *2012 IEEE 12th International Conference on Bioinformatics Bioengineering (BIBE)*, 2012, pp. 552-558.

C. C. Kannas et al., ”**A workflow system for virtual screening in cancer chemoprevention**,” in *2012 IEEE 12th International Conference on Bioinformatics Bioengineering (BIBE)*, 2012, pp. 439-446.

P. Hasapis et al., ”**Molecular clustering via knowledge mining from biomedical scientific corpora**,” in *2013 IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE)*, 2013, pp. 1-5.

C. C. Kannas, and C. S. Pattichis, ”**Self-Adaptive Multi-Objective Evolutionary Algorithm for Molecular Design**,” in *30th IEEE International Symposium on Computer-Base Medical Systems*, Thessoloniki, Greece, 22-24 June 2017, pp. 1-6.

Abstracts

C. Nicolaou, C. Kannas, and C. Pattichis, ”**Knowledge-driven multi-objective de novo drug design**,” *Chemistry Central Journal*, vol. 3, p. P22, 2009.

C. C. Kannas, and C. S. Pattichis, ”**Self-Adaptive Multi-Objective Evolutionary Algorithm for Molecular Design**,” in *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Jeju Island, Korea, 11-15 July 2017 (submitted).