Decision Engineering Report Series

Edited by Rajkumar Roy and Clive Kerr

CHALLENGES IN REAL WORLD OPTIMISATION USING EVOLUTIONARY COMPUTING

Guest Editors: Ashutosh Tiwari and Rajkumar Roy

September 2004

Cranfield University
Cranfield
Bedfordshire
MK43 OAL
United Kingdom

http://www.cranfield.ac.uk

ISBN 1-861941-09-9

'*Decision Engineering*' is an emerging discipline that focuses on developing tools and techniques for informed operational and business decision-making within industry by utilising data and information available at the time (facts) and distributed organisational knowledge.

The '*Decision Engineering Report Series*' from Cranfield University publishes the research results from the *Decision Engineering Group* in Enterprise Integration. The group aims to establish itself as the leader in applied decision engineering research. The group's client base includes: Airbus, BAE SYSTEMS, BT Exact, Corus, EDS (Electronic Data Systems), Ford Motor Company, GKN Aerospace, Ministry of Defence (UK MOD), Nissan Technology Centre Europe, Johnson Controls, PRICE Systems, Rolls-Royce, Society of Motor Manufacturers and Traders (SMMT) and XR Associates.

The intention of the report series is to disseminate the group's findings faster and with greater detail than regular publications. The reports are produced on the core research interests within the group:

- Applied soft computing
- Concurrent Engineering
- Cost engineering and estimating
- Engineering design and requirements management
- Enterprise computing
- Micro knowledge management

Edited by:

Dr. Rajkumar Roy
r.roy@cranfield.ac.uk

Dr. Clive Kerr
c.i.kerr@cranfield.ac.uk

Enterprise Integration
Cranfield University
Cranfield
Bedfordshire
MK43 OAL
United Kingdom

http://www.cranfield.ac.uk

Series librarian:

John Harrington
j.harrington@cranfield.ac.uk

Kings Norton Library
Cranfield University

Publisher:

Cranfield University

Preface

With rising global competition, it is becoming increasingly more important for industry to optimise its activities. However, the complexity of real-life optimisation problems has prevented industry from exploiting the potential of optimisation algorithms. Industry has therefore relied on either trial-and-error or over-simplification for dealing with its optimisation problems. This has led to the loss of opportunity for improving performance, saving costs and time. The growth of research in the field of evolutionary computing has been encouraged by a desire to harness this opportunity. There are a number of benefits of evolutionary-based optimisation that justify the effort invested in this area. The most significant advantage lies in the gain of flexibility and adaptability to the task in hand, in combination with robust performance and global search characteristics.

This report presents the proceedings of the workshop on ‘Challenges in Real World Optimisation Using Evolutionary Computing’. This workshop is organised in association with the Eighth International Conference on Parallel Problem Solving from Nature (PPSN VIII) held in Birmingham (UK) on 18-22 September 2004. The aim of this workshop is to explore the use of evolutionary computing techniques for solving real-life optimisation problems. It is the purpose of this workshop to bring together researchers working in the area of industrial application of evolutionary-based computing techniques such as genetic algorithms, evolutionary programming, genetic programming and evolutionary strategies. The workshop provides a great opportunity for presenting and disseminating latest work in optimisation applications of evolutionary computing in varied industry sectors and application areas, e.g. manufacturing, service, bioinformatics and retail. It provides a forum for identifying and exploring the key issues that affect the industrial application of evolutionary-based computing techniques.

This report presents three papers from the workshop. The first paper examines the possibilities of train running time control using genetic algorithms for the minimisation of energy costs in DC rapid transit systems. The second paper provides an overview of soft computing techniques used in the lead identification and optimisation stages of the drug discovery process. The third paper proposes a micro-evolutionary programming technique for optimisation of continuous space.

Dr. Ashutosh Tiwari & Dr. Rajkumar Roy
Cranfield University
September 2004

Table Of Contents

Train Running Time Control Using Genetic Algorithms for the Minimization of Energy Costs in DC Rapid Transit Systems

Thomas Albrecht

Dresden University of Technology,
Faculty of Traffic and Transportation Sciences "Friedrich List"
Chair of Traffic Control and Process Automation,
D-01062 Dresden, Germany,
albrecht@vina.vkw.tu-dresden.de,
WWW home page: http://vina.vkw.tu-dresden.de/albrecht.html

Abstract. Costs for traction energy in DC electric rail transit systems depend on the energy actually demanded in the substations as well as on average power peaks there. Both are strongly influenced by the applied train timetable, because synchronous powering of multiple trains causes high power peaks whereas coordinated powering and braking of trains leads to good usage of regenerative energy from braking and consequently less energy need at the substation.

This paper examines the possibilities of train running time modification in order to reduce power peaks and energy consumption. The problem can be described as the search for an optimal distribution of a train's running time reserve along its ride. Due to the very high complexity of the problem, the non-linearities in the model for the electric network and the very large search space, the application of Genetic Algorithms is proposed and examined in a case study for one line of the Berlin suburban railway network.

1 Motivation

1.1 Energy Costs in DC Rail Systems and the Influence of the Timetable

In DC-electric railway systems with non-inverting substations the efficient use of regenerative energy is of special importance. Firstly, energy can only be regenerated during braking, when consumers in form of motoring trains are available at the same time within the power supply network. Additionally, energy billing is realized at substation level in almost all systems and the efficient use of regenerative energy can directly contribute to reducing the amount of energy to be purchased from the energy supplier.

But energy costs are not only determined by the consumed energy itself, but also by the average power peaks in the substations. According to a UITP survey of underground railway system operators [7], the part of the energy price paid for power peaks makes up one quarter of the energy bill on average.

Both, power peaks and energy consumption, are largely influenced by the timetable the trains are travelling with. Synchronous powering of multiple trains causes high power peaks, coordinated powering and braking leads to an efficient use of regenerative energy from braking.

All these effects can only be computed taking into account the real behaviour of trains and the power supply network, the positions of the trains and their consumed or regenerated power changing every second.

1.2 The Optimization Case: Constant Headway Operation

Today, constant headway operation on a single line is the mode of choice for most of the existing rail transit lines, even when in flexible and automatic railways the headway can be adapted smoothly to demand.

There are two parameters in constant headway operation which have a big influence on energy consumption: These are headway itself and the synchronisation time (difference between departure times from the two terminus stations). But these parameters are primarily fixed in order to fulfil traffic and operational requirements and not to minimize energy costs.

The only remaining free variables within such a timetable are running time reserves that have to be included in a timetable to make it feasible even when small delays occur. But as this happens with small probability only, running time reserves can be used to de-synchronise power peaks and synchronize powering and braking trains in non-disturbed operation.

Typically, all the trains on a line are travelling according to the same running profile. That's why, finding an optimal distribution of running time reserve along a line in constant headway operation always means simultaneous optimization of multiple trains. The cost function is therefore not MARKOVian, which eliminates further solving methods, e.g. Dynamic Programming.

So, the application of Genetic Algorithms (GA) is proposed here for the solution of this problem. It is explained in detail in the next section before a case study on the Berlin suburban network is presented in Sect. 3.

2 The Application of Genetic Algorithms

With automatic train control systems it is possible to keep a timetable with the precision of one second. Distributing a certain amount of running time reserve k among n sections of a line becomes an integer problem. The coding of this problem onto a chromosome is done in such a way, that each unit of running time reserve (e.g. 1 unit = 1 sec) makes up one gene. The information the gene contents is the section of the line on which this particular unit of running time reserve is to be spent. The information is then brought into binary form using Gray-coding.

With this coding, every chromosome can be decoded into a valid solution for the given problem. The solutions are binomially distributed among the search space, favouring timetables with equally distributed running time reserve. These are in general solutions with small energy absorption of the single train, which contributes to finding the minimum of system energy consumption.

The independence of the content of the gene from its locus is favourable to good convergence as building blocks can appear on multiple locations. The use of operators changing the order within a chromosome can therefore be avoided.

By limiting the representations of a gene to selected sections of the line, operational restrictions may be taken into account, e.g. using only one third of the reserve for the first half of the ride.

The initial population is created randomly except for one individual, which presents the timetable with minimal energy consumption for the single train that can be computed using Dynamic Programming[2]. The solution is coded in such a way that at first k_1 genes are initialized with number 1, where k_1 is the number of units of running time reserve to spend on the first section of the line. Following are k_2 genes with number 2, etc. The coding principle is illustrated in Fig. 1.

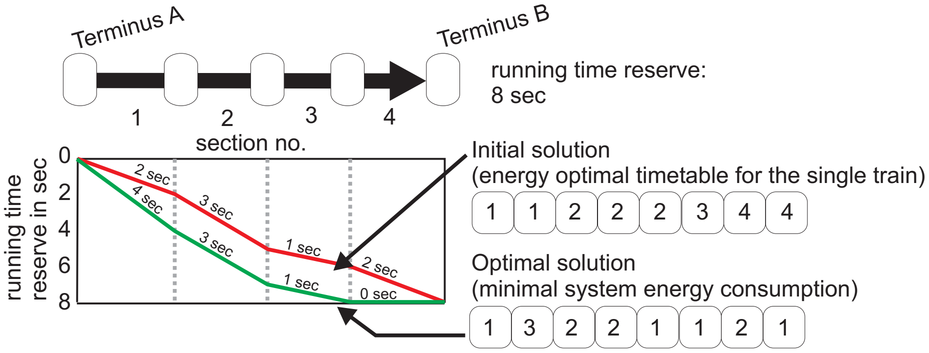

Fig. 1. Coding of running time reserve for a simplified example of distributing 8 sec on 4 sections

The number of different solutions can be calculated using simple combinatorics: The size of the search space N for the particular problem is equal to the number of combinations with allowed repetitions

$$N = \binom{n+k-1}{k} . \tag{1}$$

For a typical problem like the one presented in the next section the solution can be found using only 25 inviduals in one population for 50 generations, this is extremely fast taking into account the size of the search space $N \approx 10^{14}$ per direction.

The cost function to be minimized can be chosen freely. During simulation studies the minimization of system energy consumption, of 15-min-average power for all or selected substations as well as the maximization of the minimal voltage at the train pantograph have been used.

As the timetable with minimal energy consumption of the single train is close to the optimal solution in many cases, the Truncation method is applied for selection with the best 25% of the individuals being selected, so the probability is high for the initial solution to give its information to a comparably large number of descendants.

The binary mutation is done with a rate of 0.7 per bit, for crossover, the "Shuffle with Reduced Surrogate" method is used. Again, to protect the good existing initial solution from disappearing too early, an elitest strategy is used for reinsertion with a generation gap of 0.9.

3 Case study

A case study has been carried out for one line of the Berlin suburban railway network (S-Bahn), details can be found in [1]. The quality criteria are computed using a network simulator based on the solution of the nodal voltage equations, specificities of DC systems are taken into account as proposed in [4].

The whole algorithm was implemented in MATLAB 6.5 using an existing toolbox for the Genetic Algorithms (GEAtbx [6]). One optimisation takes between 60 and 90 mins on a 2.4 GHz Standard PC.

The problem consists of distributing 79 sec of running time reserve in the in-bound direction and 80 sec in out-bound direction which leads to a length of the chromosome of 159 sec because both directions of movement are regarded simultaneously. The optimization was conducted for different synchronization times at a headway of 10 mins.

Results for two different optimization criteria are plotted in Fig. 2. It can be seen, that for all synchronization times the values of energy consumption and 15-min-average power are much smaller for the timetables optimized for system energy and power than with the initial timetable. Although the influence of the synchronization time on the objective criteria is still important, the optimization using GA can lead to savings of 5% in system energy consumption and up to 17% in 15-min-average power.

It can also be recognised, that the GA does not necessarily find the optimal solution, e.g. at 70 sec synchronisation time energy consumption is minimal for the 15-min.-av. power optimized timetable and not for the timetable optimized for system energy consumption. But, as the difference of the energy values is very small (in the order of 0.1 %), both solutions can be regarded as very good.

The chromosomes of the best solutions obtained from the optimization of 15-min-average power have been decoded to integer values and are plotted in Fig. 3: Dark blocks vary only slightly compared to the initial solution and bright blocks have significantly different values. It can be seen, that there are synchronization times, where many building blocks [3, 5] from the initial solution survive.

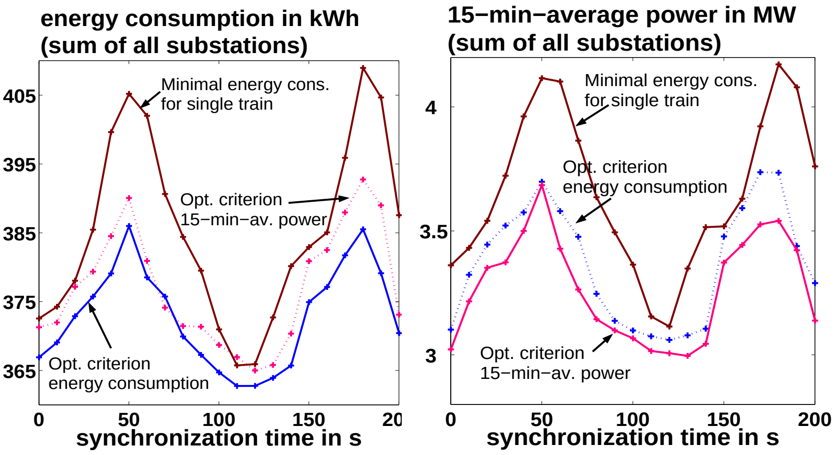

Fig. 2. Energy consumption and 15-min-average power for different synchronization times and a headway of 10 min[1]

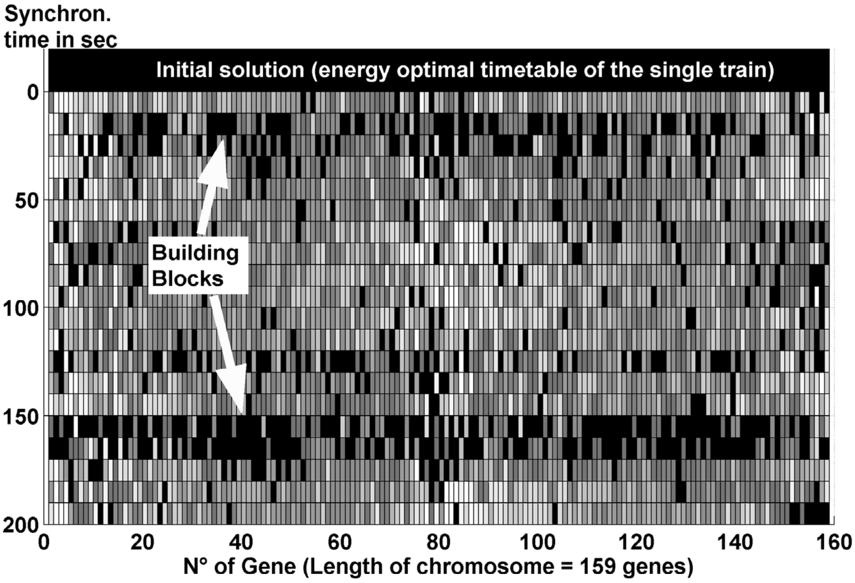

Fig. 3. Optimal solutions (15-min-av. power) for different synchronization times: The surviving Building blocks from the initial solution are clearly visible (in black).

4 Conclusions and Future Work

With the application of Genetic Algorithms it was possible to find distributions of running time reserves along a line, by which system energy consumption and 15-min-average power at the substations could be reduced significantly compared to single train optimization. The proposed coding leads to very fast convergence which is necessary taking into account the computationally expensive cost function.

In this paper, only the optimization of train running times in constant headway operation has been regarded. For the optimization of a complete timetable, transitions between periods of different headways have to be examined as well. They can also be regarded as a problem of distributing time reserves, this time between consecutive trains. The application of GA in this case is part of ongoing research.

A practical application of train dwell time optimization using GA is currently under examination in cooperation with Transpole, the operator of the fully automated metro lines VAL in Lille, France.

Acknowledgements

This paper was elaborated within the research project ”intermobil Region Dresden” funded by the German Federal Government, the Ministry of Research and Eduction (BMBF) under the project no. 19 B 9907 A 8. The author wishes to thank Prof. H. Strobel for his helpful advice and motivation during the research.

References

1. Albrecht, T., *Reducing power peaks and energy consumption in rail transit systems by simultaneous train running time control.* J. Allan, R.J. Hill, C. A. Brebbia, G. Sciutto, S. Sone (eds.), Computers in Railways IX, WIT Press, pp.885-894, 2004.
2. Albrecht, T. & Oettich, S., *A new integrated approach to dynamic schedule synchronization and energy saving train control.* J. Allan et al. (eds.), Computers in Railways VIII, WIT Press, pp. 847-856, 2002.
3. Altenberg, L., *The Schema Theorem and Price's Theorem.* in: Whitley, L. D. and Vose, M. D., Foundations of Genetic Algorithms 3, San Francisco, California, USA: Morgan Kaufmann Publishers, pp. 23-49, 1995.
4. Cai, Y., Irving, M.R. & Case, S.H., Iterative techniques for the solution of complex DC-rail-traction systems including regenerative braking. *IEE Proc.-Gener. Transm. Distrib.*, **142**(5), pp. 445-452, 1995.
5. Goldberg, D. E., *Genetic Algorithms in Search, Optimisation and Machine Learning.* Addison-Wesley, Reading, Massachussets, 1989.
6. Pohlheim, H., *Evolutionäre Algorithmen - Verfahren, Operatoren und Hinweise für die Praxis.* Springer, Berlin, Heidelberg, New York, 2000.
7. UITP, *Reducing energy consumption in Underground systems - an important contribution to protecting the environment.* Proc. of the 52$^{\text{nd}}$ International Congress, Stuttgart 1997.

Overview of Soft Computing Techniques in Lead Identification and Optimization for the Drug Discovery Process

Abiola Oduguwa, Ashutosh Tiwari and Rajkumar Roy

Dept. of Enterprise Integration, Cranfield University,
Cranfield, Bedfordshire, UK. MK43 0AL.

Abstract. Drug discovery (DD) research has evolved to the point of critical dependence on computerized systems, databases and newer disciplines. Such disciplines include but are not limited to bioinformatics, chemo informatics and soft computing. Their applications range from protein folding methods for determining protein structures to design of combinatorial libraries for identifying and optimizing new drug compounds. This paper presents a brief overview of techniques in lead identification and optimization stages of DD with their limitations, and outlines current SC based techniques in this research area.

Keywords: Drug discovery, GA, neural networks, fuzzy logic, bioinformatics

1. Introduction

Drug discovery has become an increasingly time-consuming and expensive process. Only a small fraction of the drug discovery (DD) projects undertaken eventually lead to successful medicines. Such programmes can take between 12 -16 years. Increasingly, the industry is compelled to find novel drugs that are more effective and safer than existing ones. New methods for DD are therefore receiving considerable attention in the industry. This is largely driven by several genome projects since the realization that common human diseases have genetic component. The pharmaceutical industry has thus embraced the field of genomics as a source of novel drug target in their DD process. Developments in areas of proteomics, bioinformatics and chemoinformatics are also providing solutions to the need to enhance the DD process. Bioinformatics is used to exploit the data produced on this genome-wide scale. Proteomics studies help to understand the role of gene products (protein) particularly structural information of protein for use in drug design. Chemoinformatics supports drug research by creating tools for evaluating the molecular properties of potential drug compounds in the chemical databases.

Soft computing (SC) techniques are emerging as solution alternatives for dealing with the problems of biological sciences and DD research [1-3]. It is being applied in several areas of the DD research to achieve better solutions. SC is a collection of methodologies including as its main constituents Evolutionary Computation (EC), Fuzzy Logic (FL), Neuro-computing (NC), probabilistic computing (PC), chaotic theory and parts of machine learning theory. These methodologies are parallel to the remarkable ability of the human brain to reason and learn in an environment of uncertainty, imprecision, and implicit knowledge to achieve tractability, robustness and low cost solutions [4]. SC differs from conventional techniques by providing an attractive opportunity to represent the ambiguity in human thinking with real life uncertainty which can result in more realistic solutions. It is these features that make SC a promising technology for dealing with DD problems.

This paper therefore focuses on the description of the main SC methodologies used in the lead identification and lead optimisation stages of the DD process. The paper is organised as follows. Section 2 gives an overview of the DD process; section 3 discusses the classical techniques used in DD and the application areas of SC in DD research. Section 4 discusses key findings and finally section 5 concludes this paper.

2. Drug Discovery

Drug discovery (DD) is a process of developing drugs for the safe and effective treatment of a disease. This process identifies, evaluates, and optimizes compounds and molecules with desired biological activity against a specified target or function [5]. This section briefly describes the DD process and the different stages of the process.

2.1. The Drug Discovery Process

The DD process starts with a disease target which originates from the discovery of a gene or from the elucidation of the molecular mechanism of a genetic defect. Once suitability for DD is established, new chemical entities are identified through random screening and/or rational drug design. The chemical leads with positive response in the screening process are selected and optimised as potential drug candidates. The result is a compound, or a small number of compounds that proceed to clinical trials for development into drug.

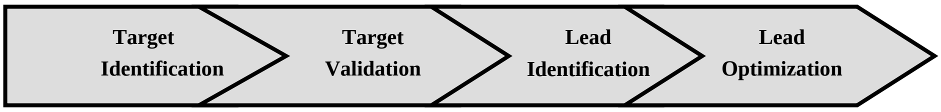

Figure 1: The drug discovery process

The DD process is divided into four main steps: target identification, target validation, lead identification and lead optimisation. This process is depicted in figure 1 and described briefly in the section that follows.

2.1.1. Target Identification

This stage of the DD process aims to identify genes or gene products that may be correlated with a disease process. This is achieved by quantifying and analyzing the gene expression in diseased and healthy states [6-8]. The ultimate goal of this step is to find macromolecules that can become binding targets for potential drug compounds.

2.1.2. Target Validation

Once the gene involved in a disease has been identified, it is necessary to validate them as drug targets. Target validation verifies the DNA or protein molecule that is directly involved in a disease process. That is, its role in disease must be clearly defined before drugs are sought that act against it. The aim is to understand the pathways/interaction of genes and to test whether the gene has the potential to be a therapeutic target [6-8].

2.1.3. Lead Identification

This is the process of identifying biologically active chemical entities that could be optimized into drugs. In this stage, compounds which interact with the target protein and modulate its activity are identified.

2.1.4. Lead Optimisation

Lead optimization is the complex multi-step process of refining the chemical structure of a hit to improve its drug characteristics, with the goal of producing a pre-clinical drug candidate. This process generally involves iterative rounds of synthetic organic chemistry and compound evaluation of a potential drug compound to ensure optimal properties in drug development [5, 9]. These properties include potency, adsorption, metabolism, distribution, toxicity (ADME/Tox).

3. Techniques in Lead Identification and Lead Optimization

This section describes the classical techniques used in the lead identification and optimization stages of the DD process with their limitations. It also describes the application of SC in these two stages.

3.1. Classical Techniques in Lead Identification and Lead Optimization

Lead Identification: The methods used for lead compounds are random screening or directed design approach and virtual screening approach [9, 10].

High-throughput Screening (HTS) is used to test large collection of compounds or structurally selected compounds in the database for their ability to affect the activity of the target protein. A compound database made up of millions of compounds is screened with a throughput of 10 000 (HTS) up to 100 000 compounds per day (µHTS, ultra high throughput screening) [5, 9, 11, 12]. The main problem of HTS is cost. It can cost up to ~$300K to set up a high throughput screen. And for most companies HTS is not an option because it can take months to scale a low throughput screen up to high throughput capacity. Additionally, many screens are not yet possible with high throughput techniques. Hit rates against some receptors are reported to be very low, necessitating screening of very large numbers of compounds (tens to hundreds of thousands) [12].

The second approach is *in silico* or virtual screening. It involves computational analysis of a subset of compounds considered to be appropriate for a given target. Three-dimensional structures of compound from virtual or physically existing libraries are docked into the binding sites of target proteins with known or predicted structures. Empirical scoring functions are used to evaluate the fit between the compounds and the target protein. The highest ranked compounds are suggested for further biological testing. One of the problems of virtual screening is the availability of protein structures. Structure prediction methods include computationally intensive strategies that simulate the physical and chemical forces involved in protein structure determination. Despite several years of research, this problem is still unsolved [13-15]. Experimental determination of protein structures by X- ray crystallography is time consuming and expensive [9]; [14]. Current prediction techniques like homology modeling and threading techniques require at least an experimentally determined protein in a fold class to model hundreds of related proteins. These techniques involve using database search tools to identify similarity between sequences and structures. The researcher then identifies the biological significance of the sequences and determines if the sequences are derived from a common ancestor. These rational techniques however, produce low resolution models due to lack of adequate understanding on how the primary structure of the protein determines its tertiary structure.

Lead Optimization: The process of lead optimization begins with evaluating hits in secondary test assays and analogs (a set of related compounds) which are then synthesized and screened. The resulting quantitative information is known as structure-activity relationships (SAR). SAR shows the relationship between chemical structure and biological data. Verkman *et al* [5] reported several approaches that are available to maximize the utility of this SAR information for directed acquisition and synthesis of structural analogues to improve compound potency. In terms of utilizing SAR data to accelerate compound optimization, visual inspection reveals many important structural features associated with activity. Several computational approaches are also reported in the literature adopted in lead optimisation; these are rational drug design, pharmacophore analysis, and quantitative structural activity relationship (QSAR) analysis. These approaches are used individually or combined in various forms. The rational design method involves the use of high resolution structure of a target to direct the synthesis of new analogues. The process usually involves generation of large library of potential derivatives and use of computational docking methods to select derivatives that may interact with the target on the basis of shape complementarities or charge placement. The pharmacophore methods involve definition of minimal unit (e.g. hydrophobic group or other functional groups) that leads to activity in a 3-D space. The consensus pharmacophore is then used to examine the allowed placements of groups in a set of candidate compounds. The last method is to establish QSAR models. This relates calculated physiochemical properties of molecules, to activity, rather than strictly structural characteristics. It usually requires a set of structurally related compounds with a wide range of activities. The main limitation of these techniques is that they are labour-intensive and time consuming. Additionally, there is a need to have a better understanding of the mechanism underlying toxicity of drugs. Current methods using animal models are time-consuming, low throughput and can be unreliable. The use of cell-based assays to predict efficacy and toxicity of hits also poses problem of reproducibility. This shows the need for more robust methods to address these problems and yield better solutions.

3.2. Soft Computing in Lead Identification and Optimization

Section 3.1 outlined the classical techniques in lead identification and optimization. It also discusses some of the limitations of these techniques. SC is now emerging as a solution alternative to deal with some of these issues. This section thus presents an overview of interesting applications of SC techniques in these two stages of the DD process. Table 1 presents a summary of the proportion of different SC techniques in the sample of publications reviewed in this paper.

3.2.1. SC in Lead Identification

Literature reveals SC techniques have been applied in the field of rational drug design to generate new leads. Lead compounds are found in existing chemical databases by fast searching or docking protocols, synthesized and isolated by combinatorial chemistry, or designed *de novo* by computational design programs [16]. SC techniques have been applied in each of these areas.

GA has been applied to the problem of finding two dimensional matches to a query in chemical databases [16-18]. It has also used to compare 3-D structures, both to determine optimal alignments of molecular electrostatic potential fields in rigid searches [19] and in flexible searching for a pharmacophoric pattern [20].

Another method for lead identification is docking of ligands into the active site of a target [16, 21-28]. Computational methods for docking involve a good scoring function and an efficient searching algorithm for searching conformational spaces. Trial ligands are taken from a 3D database, placed into the template site, and ranked in order of predicted binding affinity. A variety of methods have been used to obtain plausible binding orientations. One philosophy requires a user to manipulate the ligand, while the computer interactively reports a binding score [16]. Several groups have used GA in this area [16, 29-34]. [34] developed an evolutionary method, GEMDOCK for molecular docking and empirical scoring function. The program combines discrete and continuous global search strategies to speed up convergence. DOCK program [16, 29] uses GA to dock flexible ligands into rigid receptor after characterizing and identifying binding sites using a surface sphere cluster method. For each of the iteration, selective pressure is applied to encourage high-scoring features of current generation to be preserved in the next cycle. Random 'mutations' are permitted, while 'crossover' moves allow molecules to exchange characteristics.

In addition to finding lead compounds by the previously described methods, they can also be developed experimentally by *de novo* design [16, 21-25]. Glen and Payne applied GA to the design of substructures based on a wide variety of user-defined constraints. The constraints selected for various design experiments provide the basis for the fitness function.

Other application areas of SC in lead identification are protein folding simulations for predicting the three-dimensional structure of a target protein [2, 35-38] and combinatorial library design [39].

3.2.2. SC in Lead Optimisation

The application areas of SC in lead optimisation include: combinatorial chemistry, *de novo* design leads and quantitative structure activity relationship measurement (QSAR) [16].

Combinatorial chemistry involves synthesis and screening of large libraries of compounds to determine lead compounds that exhibit biological activities of interest. GA has been applied in this area with successes in the design and automated synthesis of combinatorial compound libraries [16]. The optimization behavior of GA perfectly matches the discontinuous, non-steady structure space of chemistry. Lutz Weber demonstrated use of GA to develop thrombin inhibitors using non-peptidic molecules [40-42]. The study involves a Ugi reaction and 10x40x10x40 building blocks gives 160,000 combinatorial products. GA was also used in combinatorial chemistry to select fragments for assembly into the library. Sheridan and Keasley [16, 40-42] applied GA to the optimisation of tripeptoids constructed from a wide variety of primary and secondary amines. Each tripeptoid generated in the GA run was evaluated based on its similarity to a target tripeptoid. The ones with high fitness values were chosen for use in library synthesis.

The application of SC in *de novo* design is shown in the optimisation of the results of the design program PRO_LIGAND [16]. This program generates new lead compounds by assembling fragments for substructure libraries.GA uses these leads as the initial population for the optimisation process.

The third approach for optimisation of lead compounds is QSAR modeling. The method attempts to find relationship between the properties of bioactive molecules and the biological responses they produce when applied to a biological system. Winkler and Burden developed QSAR models to improve the efficiency of bioactive molecules with Bayesian regularized artificial neural networks (BRANNs) [43]. A hybrid GA/neural network method have also been used to suggest more potent descriptors from a set of variables in a QSAR model for dihydrofolate reductase inhibitors [3].

Table 1: Summary of applications of SC in drug discovery

DD Process	Applications	Characteristics	SC component
Lead Identification	Database Searching	• Comparison of two and three – dimensional structures using GA	GA
	Protein Folding / Structure Prediction	• Coded GA for simulating protein folding problem	ANNs, FL, GA
	Virtual Screening and molecular Docking	• Docking of ligands in to receptor sites of target protein molecule using GA	GA
	De novo drug design	• Design of protein substructures, receptors, enzymes, ion channels using GA	GA
Lead Optimisation	Combinatorial Chemistry	• Development of thrombin inhibitors with non-peptidic molecules • Design of similar and dissimilar compounds in a combinatorial library	GA
	De novo drug design	• Optimisation of leads generated by PRO-LIGAND program	GA
	QSAR	• BRANNs for producing better descriptors and designing molecules with desired activity	GA, ANNs

4. Discussion and Conclusion

This paper gives a brief overview of the classical techniques in lead identification and optimization. It was shown that these techniques have several limitations which led to the introduction of soft computing. This paper reveals the several application areas of SC in these two stages. The main SC techniques are GA, FL, and ANNs. Literature reveals a number of studies of the application of GAs in these stages over the last 15 years. This is largely due to the nature of the data and the search space being explored. Many phases of the drug design involve finding solutions to large combinatorial problems for which exhaustive search is intractable. GA has been particularly useful in this area to rapidly find good solutions to such problems [16].

ANNs has also been successful in gene expression data analysis and pattern recognition in protein structure determination. FL has also been shown to help in extraction of information from biological datasets. The key feature of DNA that makes it appropriate for fuzzy computing is the uncertainty and incompleteness in the information of the double stranded duplex. These three core techniques of soft computing have also been used in its hybrid form. For example, neuro-fuzzy algorithm that was used in protein motif extraction [1]. GA, ANNs and FL have thus proven their strengths in handling the imprecise nature of biological data.

In conclusion, this paper presents a review of the application of SC techniques in lead identification and optimization. It explores the discovery phase of drug research. The paper shows a brief review of the classical techniques in these research areas with their limitations and critically evaluates how current SC techniques are suited for such complex biological sciences problems. Current research shows that SC methods have significant potential in dealing with the limitation posed by traditional methods.

Acknowledgements

The authors will like to thank Cranfield University, UK for financially supporting this research.

References

1. Chang, B.C.H. and S.K. Halgamuge, *Protein Motif Extraction with Neuro-Fuzzy Optimization.* Bioinformatics, 2002. **18**(8): p. 1084-1090.
2. Pedersen, J.T. and J. Moult, *Protein Folding simulations with genetic algorithms and a detailed description.* Journal of Molecular Biology, 1997. **269**: p. 240 - 259.
3. Manallack, D.T. and D.J. Livingstone, *Neural networks in drug discovery: have they lived up to their promise?* Europen Jopurnal of Medicinal Chemistry, 1999. **34**: p. 195 - 208.
4. Oduguwa, V., *Rolling System Design Optimisation Using Soft Computing Techniuques*, in *Enterprise Integration*. 2003, Cranfield: Bedfordshire. p. 332.
5. Verkman, A.S., *Drug Discovery in academia.* American Journal of Physiology - Cell Physiology, 2004. **286**: p. C465-C474.
6. Searls, D.B., *Using Bioinformatics in gene and drug discovery.* Drug Discovery Today, 2000. **5**(4): p. 135-143.
7. Swindells, M.B. and J.P. Overington, *Prioritizing the proteome: identifying pharmaceutically relevant targets.* Drug Discovery Today, 2002. **7**(9): p. 516-521.
8. Knowles, J. and G. Gromo, *Target Selection in Drug Discovery.* Nature Reviews: Drug Discovery, 2002. **2**: p. 63-69.
9. Hillisch, A. and R. Hilgenfield, *Modern Methods of Drug Discovery*. 2003: Springer Verlag. 304.
10. Castrodale, B., *Leading Genomic Approaches for Breaking Bottlenecks in Drug Discovery and Development.* 2002, Cambridge Healthtech Institute: Massachusetts. p. 1-7.
11. FitzGerald, K., *In vitro display technologies - new tools for drug discovery.* Drug Discovery Today, 2000. **5**(6): p. 253-258.

12. Bleicher, K.H., et al., *Hit and Lead Generation: Beyond High-Throughput Screening.* Nature Review Drug Discovery, 2003. **2**(5): p. 369-378.

13. Clark, D.E. and S.D. Pickett, *Computational methods for the prediction of 'drug-likeness'.* Drug Discovery Today, 2000. **5**(2): p. 49-57.

14. Maggio, E.T. and K. Ramnarayan, *Recent developments in computational proteomics.* Drug Discovery Today, 2001. **6**(19): p. 996-1004.

15. Bajorath, J., *Rational drug discovery revisited: interfacing experimental programs with bio- and chemo-informatics.* Drug Discovery Today, 2001. **6**(19): p. 989-995.

16. Parrill, A., *Evolutionary and genetic methods in drug design.* Drug Discovery Today, 1996. **1**(12): p. 514 - 521.

17. Brown, R.D., et al., *Matching two-dimensional chemical graphs using genetic algorithms.* Journal of Chemical Information and Computer Sciences, 1994. **34**(1): p. 63 - 67.

18. Clark, D.E., et al., *Pharmacophoric pattern matching in files of three-dimensional chemical structures: Comparison of conformational-searching algorithms for flexible searching.* Journal of Chemical Information and Computer Sciences, 1994. **34**(1): p. 197-206.

19. Fontain, E., *Application of genetic algorithms in the field of constitutional similarity.* Journal of Chemical Information and Computer Sciences, 1992. **32**(1): p. 748-752.

20. Wild, D.J. and P. Willett, *Similarity Searching in Files of Three-Dimensional Chemical Structures. Alignment of Molecular Electrostatic Potential Fields with a Genetic Algorithm.* Journal of Chemical Information and Computer Sciences, 1996. **36**(2): p. 159-167.

21. Glen, R.C. and A.W.R. Payne, *A genetic algorithm for the automated generation of molecules within constraints.* Journal of Computer-Aided Molecular Design., 1995. **9**(2): p. 181-202.

22. Pegg, S.C.H., J.J. Haresco, and I.D. Kuntz, *A Genetic Algorithm for Structure-based De Novo Design.* Journal of Computer-Aided Molecular Design., 2001. **15**: p. 911-933.

23. Budin, N., et al., *Structure-based Ligand Design by a Build-up Approach and Genetic Algorithm Search in Conformational Space.* Journal of Computational Chemistry, 2001(22): p. 1956-1970.

24. Budin, N., et al., *An Evolutionary Approach for Structure-based Design of Natural and Non-natural Peptidic Ligands.* Combinatorial Chemistry and HTS, 2001. **4**: p. 661-673.

25. Schneider, G., et al., *Virtual Screening for Bioactive Molecules by Evolutionary De Novo Design.* Angewandte Chemie International Edition in English, 2000. **39**: p. 4130-4133.

26. Wang, R., Y. Gao, and L. Lai, *LigBuilder: A Multi-Purpose Program for Structure-based Drug Design.* Journal of Molecular Modeling, 2000. **6**: p. 498-516.

27. Jagla, B. and J. Schuchhardt, *Adaptive Encoding Neural Networks for the Recognition of Human Signal Peptide Cleavage Sites.* Bioinformatics, 2000. **16**: p. 245-250.

28. Schneider, G., et al., *De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks.* Journal Computer-Aided Molecular Design, 2000. **14**: p. 487-494.

29. Oshiro, C.M., I.D. Kuntz, and J.S. Dixon, *Flexible ligand docking using a genetic algorithm.* Journal of Computer-Aided Molecular Design, 1995. **9**(1): p. 113-130.

30. Jones, G., P. Willett, and R.C. Glen, *Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation.* Journal of Molecular biology, 1995. **245**: p. 43-53.

31. Globus, A., L. J, and W. T, *Automatic molecular design using evolutionary*

techniques. Nanotechnology, 1999. **10**: p. 290 - 299.

32. Bamborough, P. and F.E. Cohen, *Modelling Protein-Ligand complexes.* Current Opinion in Structural Biology, 1996. **6**: p. 236 - 241.
33. Desjarlais, J.R. and N.D. Clarke, *Computer search algorithms in protein modification and design.* Current opinion in Structural Biology, 1998. **8**: p. 471 - 475.
34. Yang, J.-M. and C.-C. Chen, *GEMDOCK: A generic evolutionary method for molecular docking.* PROTEINS: Structure, Function, and Bioinformatics, 2004. **55**: p. 288 - 304.
35. Cooper, L.R., D.W. Corne, and J.C. Crabbe, *Use of a novel Hill-Climbing genetic algoriothm in protein folding simulations.* Computational Biology and Chemistry, 2003. **27**: p. 575 - 580.
36. Agostini, L. and S. Morosetti, *A simple procedure to weight empirical potentials in a fitness function so as to optimise its performance in ab initio protein-folding problem.* Biophysical Chemistry, 2003. **105**: p. 105 - 118.
37. Pedersen, J.T. and J. Moult, *Genetic algorithms for protein structure prediction.* Current Opinion in Structural Biology, 1996. **6**: p. 227 -231.
38. Konig, R. and T. Dandekar, *Improving genetic algorithms for protien folding simulations by systematic crossover.* BioSystems, 1999. **50**: p. 17 - 25.
39. Illgen, K., et al., *Simulated molecular evolution in a full combinatorial library.* Chemistry and Biology, 2000. **7**: p. 433- 441.
40. Weber, L., *Evolutionary combinatorial chemistry: application of genetic algoithms.* Drug discovery Today, 1998. **3**(8): p. 379 - 385.
41. Weber, L., *Application of genetic algorithms in molecular diversity.* Current Opinion in Chemical Biology, 1998. **2**: p. 381 - 385.
42. Weber, L., et al., *Optimisation of the Biological Activity of Combinatorial Compound Libraries by a Genetic Algorithm.* Angewandte Chemie International Edition in English, 1995. **34**: p. 2280-2282.
43. Winkkler, D.A. and F.R. Burden, *Bayesian neural nets for modeling in drug discovery.* DDT: BIOSILICO, 2004. **2**(3): p. 104-111.

A Micro Evolutionary Programming for Optimization of Continuous Space

Xinchao Zhao and Xiao-Shan Gao

Key Laboratory of Mathematics Mechanization,
Institute of Systems Science, AMSS, Academia Sinica
Beijing, 100080, China
(xczhao,xgao)@mmrc.iss.ac.cn

Abstract. With the development of science, more and more powerful algorithms and problem-solving tools appeared. However, they are becoming more and more complex. In this paper, we propose a micro evolutionary programming (MEP) which is easy to use and ease of parallelization. It also performs excellent and robust. Population size of MEP is 2, each of which performs local search in its neighborhood and exchanges inherent information in a probability. Furthermore, not *Gaussian* or *Cauchy* mutation, but *non-uniform mutation*, is adopted in MEP which has the feature of searching the space uniformly initially and very locally at later stages of algorithms. MEP solve four functions whose dimensions rapidly increase (up to 1000). It also solves one low dimensional function, however, whose search domain is greatly enlarged (maximal 1000 times). These two groups of experiments proved its strong exploration ability, robustness, and excellent performance from two different perspectives.

Keywords: evolutionary algorithm, non-uniform mutation, global optimization, greedy idea

1 Introduction

Genetic algorithms (GAs) refers to the meta-heuristical use of concepts, principles and mechanisms based on our understanding about the natural evolution to help solve the complex problems [1–3]. GAs has been successfully applied to many areas [4–8]. In order to obtain more satisfiable algorithms, more and more modified GAs, such as [9, 10, 6, 11] are proposed to enhance GAs. However, as Wolpert and Macready [12] stated, we have to pay out more computing cost or even stronger computing platform. That is to say, computing tools become more and more complicated as they become more and more powerful.

In this paper, a micro evolutionary programming (MEP) of easy to use and ease of parallelization is proposed. At the same time, it performs very robust and satisfiable. The population size of MEP is set to be 2 in order to exchange inherent information. Of course, larger population size will do to enhance its performance. Inspired by the well known incomplete algorithm GSAT, MEP performs greedy local search in the neighborhood of every individual and exchanges their loci genes in a probability. Experiments on the multimodal functions with increasing dimensions are done. These experiments show that the performance of MEP is nearly not affected by the increasing dimensions of benchmarks. As similar works are not found in literatures, we do not make comparisons with the related works. But the key point is that our MEP performs very robust and excellent.

The other experiments further prove the robustness of MEP with the search space of the function greatly expanding. Yao et al. [14] did similar experiments based on the *Cauchy* mutation-based evolutionary programming. They made comparisons between the classical evolutionary programming (*Gaussian mutation*, CEP) and the fast evolutionary programming (*Cauchy mutation*, FEP). It shows that FEP outperforms CEP because FEP generates more long jumps than CEP. We compared MEP with FEP&CEP with the expanding definition domains and showed that MEP performs even more robust than FEP. The feature of non-uniform mutation and the effective local search ensure the robust performance of MEP.

2 Micro Evolutionary Programming

Michalewicz [15] proposed a non-uniform mutation which has the feature of searching the space uniformly initially and very locally at later stages of algorithms. The operator of exchange inherent information is similar with two point crossover [3] in binary genetic algorithm.

2.1 Local Search of MEP

First of all, we give the definition of **neighborhood** of an individual.

Definition: Given a vector $\mathbf{X} = (x_1, \ldots, x_i, \ldots, x_m)$ (m is the dimension of vectors), we call $\mathbf{X}'$ is its neighbor, *if and only if* one of its component is changed and other components remain unchanged. Then the neighborhood $\mathbf{N}$ of a vector $\mathbf{X}$ is composed of all its neighbors. That is

$$\mathbf{N} = \{\mathbf{X}' | \mathbf{X}' \text{ is a neighbor of } \mathbf{X}\} \tag{1}$$

Different from the traditional local search that performs greedy local search until a local optimum is obtained, we will perform once non-uniform local search for a chromosome (vector) $\mathbf{X}$ in its neighborhood which is like to the well known GSAT [13] which just tries one assignment of an expression once. The current individual will be replaced by the new one only if the new is **not worse than** the current individual.

2.2 Non-uniform Mutation

Michalewicz [15] proposed a dynamical non-uniform mutation operator to reduce the disadvantage of random mutation in the real-coded evolutionary algorithm. This new operator is defined as follows.

For each individual X_i^t in a population, it creates an offspring X_i^{t+1} through non-uniform mutation as follows: if $X_i^t = \{x_1, x_2, \ldots, x_m\}$ is a chromosome (t is the generation number) and the element x_k is selected for mutation, the result is a vector $X_i^{t+1} = \{x_1', x_2', \ldots, x_m'\}$, where

$$x_k' = \begin{cases} x_k + \Delta(t, UB - x_k), & \text{random number is 0} \\ x_k - \Delta(t, x_k - LB), & \text{random number is 1} \end{cases} \tag{2}$$

and LB, UB are the lower and upper bounds of the variable x_k. The function $\Delta(t, y)$ returns a value in the range [0,y] such that the probability of $\Delta(t, y)$ being close to 0 as t increases. This property causes this operator to search the space uniformly initially (when t is small) and very locally at later stages. The probability of generating an offspring closer to its parent is increased which is higher than a random choice, which is similar to the working scheme of simulated annealing. We used the following function:

$$\Delta(t, y) = y \cdot (1 - r^{(1-\frac{t}{T})^b}), \tag{3}$$

where r is a random number from [0, 1], T is the maximal generation number, and b is a system parameter determining the degree of dependency on iteration number.

3 Experiments and Analysis

Functions	n	**D**	f_{min}
$f_{Sph} = \sum_{i=1}^{n} x_i^2$		[-100, 100]^n	0
$f_{Ros} = \sum_{i=1}^{n-1} [100(x^{i+1} - x_i^2)^2 + (x_i - 1)^2]$		[-5.12, 5.12]^n	0
$f_{Gri} = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})+1$		[-600, 600]^n	0
$f_{Pen} = \frac{\pi}{n}\{10 \sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})]$ $+(y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4), y_i = 1 + \frac{1}{4}(x_i + 1)$		[-50, 50]^n	0
$f_{Shekel} = -\sum_{j=1}^{5} [\sum_{i=1}^{4} (x_i - a_{ij})^2 + c_j]^{-1}$	4	[0, 10]^n	-10.1532

Table 1. functions used in the paper

Sphere model function f_{Sph}, generalized Rosenbrock's function f_{Ros}, generalized Griewank function f_{Gri} and generalized Penalized function f_{Pen} are chosen from [14] in this paper. Function f_{Sph} is a typical unimodal function, f_{Ros} is a continuous and unimodal function, however, its optimum located in a steep parabolic valley with a flat bottom and the variables having nonlinear interactions among them. Function f_{Gri} is nonseparable and the search algorithm has to climb a hill to reach the next valley and f_{Pen} is also a typical multimodal function.

3.1 Experiments on Increasing Dimension Numbers of Problems

The experiments were executed 10 times independently for the long computing time. Population size is 2, maximal evolutionary generation number is 1500, crossover probability $pc = 0.4$ and mutation probability in the neighborhood $pm = 0.6$. The computer is with a configuration of *2.40GHz CPU, 4G RAM.* The experimental results are given in Table 2.

Dim	f_{Sph}					f_{Ros}				
numbers	**A**	**SD**	**B**	**nFit**	**t**(s)	**A**	**SD**	**B**	**nFit**	**t**(s)
30	2.9e-13	6.1e-13	2.3e-15	3.7e4	0.12	5.3e1	3.8e1	6.5e0	3.7e4	0.2
100	1.9e-12	3.3e-12	7.3e-14	1.2e5	0.66	2.2e2	1.4e2	1.3e2	1.2e5	0.8
250	2.1e-12	2.5e-12	2.7e-13	3e5	1.95	2.7e2	4.5e2	2.4e2	3e5	3.48
500	6.7e-12	4.3e-12	1.7e-12	6e5	14.56	9.5e2	1e2	7.4e2	6e5	17.72
750	3.9e-11	7.2e-11	1.8e-12	9e5	15.46	1.4e3	3.5e2	1.1e3	9e5	26.55
1000	8e-12	7.1e-12	1.1e-12	1.2e6	25.56	1.8e3	9.3e1	1.7e3	1.1e6	48.92
Dims	f_{Gri}					f_{Pen}				
number	**A**	**SD**	**B**	**nFit**	**t**(s)	**A**	**SD**	**B**	**nFit**	**t**(s)
30	6.6e-3	1.3e-2	1.4e-14	3.7e4	0.45	2.9e-14	8.8e-14	3.3e-18	3.7e4	1.05
100	6.6e-3	8.7e-3	6e-14	1.2e5	2.77	1.3e-15	2.1e-15	1.4e-17	1.2e5	7.23
250	6.9e-3	7.4e-3	1.3e-13	3e5	15.31	5.2e-16	9.1e-16	3.1e-17	3e5	44.24
500	6.1e-3	6.9e-3	2.2e-13	6e5	68.40	5.3e-16	8.1e-16	7.8e-17	6e5	170.44
750	5.9e-3	5.2e-3	5.9e-13	9e5	133.04	5.7e-15	1.5e-14	1.2e-16	9e5	332.40
1000	9.3e-3	1.3e-2	2.2e-13	1.2e6	280.79	8.2e-16	6.4e-16	1.3e-16	1.2e6	683.91

Table 2. Experimental results for increasing dimensions of functions. **A** is the average of the best result found at the end of each run. **SD** stands for standard deviation. **B** is the best of the fitness in the all runs. **nFit** is the average function evaluations. **t** is the average CPU time (second)

From Table 2, we find that the solution quality of the multimodal functions f_{Gri}, f_{Pen} and f_{Sph} have no significantly statistical difference at all for the increasing dimensions. Even for the nonseparable function f_{Ros}, the difficulty is also linearly increasing when its dimensions expand to 1000 from 100 observed from the **A** and **B** performances. Generally speaking, MEP has a very thick skin to the initial conditions and comparable results can be obtained even the search spaces become larger and larger[1]. As far as the computing speed is considered, the number of function evaluations is just **linearly** dependent on the dimension numbers of problems.

[1] We also did experiments with even higher dimensions (just computing two times) and obtained similar conclusion.

3.2 Experiments on Expanding Definition Domains

Another group of experiments with the constringent or expanded definition domains are done to validate the convergence or robustness of MEP from another different perspective. One of the **Shekel's Family** SQRN5 function [14] is chosen to do this experiment. Its global minimum is -10.15. The experimental conditions are equivalent to [14] and 50 independent runs are executed in this group of experiments.

	MEP		FEP		CEP	
Initial Range	Mean Best	Std Dev	Mean Best	Std Dev	Mean Best	Std Dev
$2.5 \leq x_i \leq 5.5$	-10.15	0	-5.62	1.71	-7.9	2.85
$\mathbf{0 \leq x_i \leq 10}$	**-5.68**	**2.36**	**-5.57**	**1.54**	**-6.86**	**2.94**
$0 \leq x_i \leq 100$	-5.06	0	-5.80	3.21	-5.59	2.97
$0 \leq x_i \leq 1000$	-5.06	0	-5.00	2.96	-5.33	2.76
$0 \leq x_i \leq 10000$	-5.00	0.16	-3.97	2.28	-2.60	2.43

Table 3. Comparison of MEP with FEP&CEP [14] on solution quality of f_{Shekel} when the initial population is generated uniformly in a constringent or expanded ranges of variables. The black line shows the normal variables ranges.

Generally speaking, observed from Table 3, the performance of MEP is greatly improved when the variables ranges become smaller and has no obvious degradation when the variable ranges are expanded. But contrary conclusions can be reached when other two algorithms are observed. If the current solution is close to the global optimum, MEP shows powerful exploitation (fine-tuning) ability which can be validated from the experimental results with $2.5 \leq x_i \leq 5.5$ that MEP all found its global optimum in 50 runs. If the current solution is far (or very far) from the global optimum, MEP shows mighty exploration ability as well which can be illuminated from the experiment with $0 \leq x_i \leq 10000$ when other two algorithms are already far worse than the results of smaller search spaces.

This group of convincing experiments, together with the experiments of Section 3.1, indicate that MEP is not sensitive to the initial conditions and performs very robust. The robust and nonsensitive features of MEP are highly suitable to the real-world problems (such as engineering computing) whose global optimum, even the problems themselves, are usually unknown. Under these situations, MEP should have broader applications.

4 Conclusions and Future Works

An easy use, robust, excellent and ease of parallelization micro evolutionary programming MEP is proposed in this paper. Two groups of experiments with greatly expanding search spaces are done. Results show that MEP has a sick skin to the initial conditions. Experiments validate the robustness and excellent performance of MEP. However, the current algorithm is not suitable to combinatorial optimization due to the non-uniform mutation. Subsequently, more comparisons with other algorithms, especially for real-world problems and theoretic proof how it works will be done.

ACKNOWLEDGEMENT

We would like to thank the committees for their good suggestions.

References

1. J.H. Holland, Adaptation in Nature and Artificial Systems, University of Michigan Press, Ann Arbor, 1975
2. M. Zhou, S.D. Sun, Genetic Algorithms: Theory and Applications (in Chinese), National University of Defense Technology Press, China, 1999
3. D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Reading, MA: Addison-Wesley, 1989
4. W.H. Au, K.C.C. Chan and X. Yao, Data Mining by Evolutionary Learning for Robust Churn Prediction in the Telecommunications Industry, IEEE Transactions on Evolutionary Computation, 7(6) (2003) 532-545
5. Y. Liu, X. Yao and T. Higuchi, Evolutionary Ensembles with Negative Correlation Learning, IEEE Transactions on Evolutionary Computation, 4(4) (2000) 380-387
6. J. Berger, M. Barkaoui, A Hybrid Genetic Algorithm for the Capacitated Vechicle Routing Problem, in GECCO 2003, (2003) 646–656
7. R. Unger, J. Moult, Genetic Algorithms for Protein Folding Simulations, J. Mol. Biol. 231 (1993) 75-81
8. W. Romao, A. Freitas, I.M.de.S. Gimenes, Discovering interesting knowledge from a science and technology database with a genetic algorithm, Applied Soft Computing 4 (2004), 121-137
9. J.J. Grefenstette, Incorporating Problems Specific Knowledge into Genetic Algorithms, in Genetic Algorithms and Simulated Annealing, L. Davis (Eds.) (1987) 42-60
10. S.W. Mahfoud, Niching Methods for Genetic Algorithm, Univ. Illinois at Urbana-Champaign, Illinois Genetic Algorithms Lab., IlliGAL Rep. 95001, 1995.
11. X.C. Zhao, X.S. Gao, A Hybrid Genetic Algorithm and its Applications to Optimization and SAT Problems, SNPD2004, A Publication of the International Association for Computer and Information Science, 12-18.
12. D.H. Wolpert, W.G. Macready, No Free Lunch Theorems for Optimization, IEEE Trans. Evol. Comput., (1997) vol1
13. B. Selman, H.J. Levesque and D.G. Mitchell, A New Method for Solving Hard Satisfiability Problems, in Proc. of the AAAI'92, San Jose, CA (1992) 440-446
14. X. Yao, Y. Liu and G. M. Lin, Evolutionary Programming Made Faster, IEEE Trans. Evol. Comput., (1999) vol3, No.2
15. Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, (3rd Edition), Springer, 1996