

Evolutionary Algorithm as an Approach for
Computer Assisted Structure Elucidation of
Organic and Bioorganic Compounds

Dissertation

zur Erlangung des akademischen Grades doctor rerum naturalium
(Dr. rer. nat.)

vorgelegt dem Rat der Chemisch-Geowissenschaftlichen Fakultät der
Friedrich-Schiller-Universität Jena
von M. Sc. Yongquan Han
geboren am 25.09.1970 in Shanxi (China)

Gutachter:

1. Prof. Dr. Ernst Anders
2. Prof. Dr. Wilhelm Boland (Max Planck Institute for Chemical Ecology, Jena)
3. PD Dr. Christoph Steinbeck (Cologne University Bioinformatics Center)

Tag der öffentlichen Verteidigung: 17. 12. 2003

For Weiwei

Contents

Zusammenfassung

1 Introduction

2 The Evolutionary Search Approach

2.1 Characteristics of the Constitution Space

2.2 Deterministic and Stochastic Approaches

2.3 Evolutionary Algorithms

2.4 Customizing Evolution Schemes

3 Representation of Candidate Solutions

3.1 Basic Consideration

3.2 The Data Structure

3.2.1 Labeled Molecular Graph

3.2.2 Parameters and Attributes

4 Knowledge-based Structure Reconstruction

4.1 Design Principles

4.2 Mutation

4.2.1 Concept

4.2.2 Parameter Description

4.3 Crossover

4.3.1 Concept

4.3.2 Parameter Description

4.4 Niche Search

4.5 Auxiliary Operators

4.6 Customization

5 Fitness Function

- 5.1 Components of the Fitness Function
- 5.2 Construction of the Fitness Function
 - 5.2.1 Standard Fitness Function
 - 5.2.2 Advanced Assemble Strategy
- 6 Selection Policy
 - 6.1 Selection Mechanisms
 - 6.2 Fitness Scaling
- 7 Population Strategies
 - 7.1 Similarity Measures and Population Diversity
 - 7.2 Diversity-guided Step Size Control
- 8 Evolution Schemes
 - 8.1 Simple Evolutionary Algorithm
 - 8.2 Steady-State Evolutionary Algorithm
 - 8.3 Diversity-Driven Evolutionary Algorithm
- 9 Results and Discussion
- 10 Conclusions and Outlook
- Appendix
- References
- Publications
- Curriculum Vitae
- Acknowledgement

Evolutionäre Suchmechanismen als Zugang zur Computergestützten Strukturaufklärung Organischer und Bioorganischer Verbindungen

Deutschsprachige Zusammenfassung der Doktorarbeit
von Yongquan Han aus Shanxi, China

Einleitung

Computergestützte Strukturaufklärung in der organischen Chemie hat zum Ziel, innerhalb eines gegebenen Suchraumes einen möglichst kleinen Satz von Strukturen zu finden, die mit den gegebenen chemischen und spektroskopischen Randbedingungen in Einklang stehen^{1,2}. In der Literatur hat sich das Akronym CASE (Computer-Assisted Structure Elucidation) als Kurzform für dieses Gebiet der Chemoinformatik eingebürgert.

Seit den ersten Ansätzen zur automatischen Strukturaufklärung wurden sämtliche zu diesem Zwecke nützlichen spektroskopischen Verfahren, wie Massenspektrometrie (MS), Infrarotspektroskopie (IR) und vor allem NMR-Spektroskopie, als Basis verwendet. Mit dem Aufkommen der 2D-NMR Spektrometrie in der Mitte der 1970'er Jahre wurden die restlichen Verfahren in ihrer Bedeutung zurückgedrängt, was sich bis heute in einer NMR-Lastigkeit der existierenden CASE-Computersysteme niederschlägt³. In der Regel dienen als Eingaben die Summenformel der unbekanntenen Verbindung, abgeleitet aus Elementaranalyse oder hochaufgelöster Massenspektrometrie (HR-MS), sowie die 1D NMR Experimente ¹H- und ¹³C-NMR (BB, DEPT) und die 2D Korrelations-Experimente HH COSY , HMQC, HMBC, und andere.

Als Standard-Verfahren zur automatischen Strukturaufklärung hat sich der folgende 3-Schritt-Prozess eingebürgert:

Exzerpieren von Strukturfragmenten aus der spektroskopischen Information, die entweder in der Zielstruktur vorhanden sein müssen (Goodlist) oder nicht vorhanden sein dürfen (Badlist).

Erschöpfende und irredundante Generierung aller Strukturen, die sowohl alle Strukturfragmente aus Schritt 1 enthalten, als auch Konstitutionsisomere der gegebenen Summenformel sind. Dieser Schritt wird mit Hilfe eines s. g. Strukturgenerators durchgeführt.

Abschließende Untersuchung der Ergebnisstrukturen auf Validität. Eventuell Vorhersage von Spektren für alle Kandidaten und Erstellen einer Rangliste nach Vergleich der Übereinstimmung zwischen berechneten Spektren der Kandidaten und tatsächlichem Spektrum der unbekanntes Verbindung.

Während sich aus den älteren spektroskopischen Verfahren MS, IR und 1D-NMR Strukturfragmente des in Punkt 1 erwähnten Typs ableiten lassen, liefert die 2D-NMR-Spektrometrie einen weiteren Typ von Randbedingung, der Aussagen über Pfade zwischen korrelierenden Atomen in molekularen Graphen macht. Ein Kreuzsignal im 2-dimensionalen HMBC NMR-Experiment z. B. macht die Aussage, dass die zwei an der Entstehung des Signals beteiligten Kerne im molekularen Bindungsgerüst entweder zwei oder drei Bindungen voneinander entfernt liegen. Es lässt sich jedoch in diesem Fall nicht feststellen, um welche Pfadlänge es sich handelt; auch gibt es – wenn auch seltene – Ausnahmen von dieser Regel, derer das CASE System Rechnung tragen muss. Diese Art von Information lässt sich besonders gut prospektiv innerhalb der im o. g. Schritt 2 verwendeten Strukturgeneratoren bereits beim Aufbau der Konstitutionsisomere verwenden.

Rechenverfahren

Die im Laufe der fast 20-jährigen Geschichte von CASE-Programmen verwendeten Ansätze lassen sich am besten anhand der von ihnen verwendeten Strukturgeneratoren klassifizieren. Im klassischen, bis vor wenigen Jahren ausschließlich verwendeten deterministischen Verfahren, werden erschöpfend alle mit den Randbedingungen in Einklang stehende Konstitutionsisomere erzeugt. Eine Handvoll prominenter Implementierungen dieses Zuganges wurden von Gruppen um die Pioniere der automatischen Strukturaufklärung, wie Munk, Sasaki oder Chen, veröffentlicht^{1,4}. Hierbei lassen sich die deterministischen Strukturgeneratoren in solche, die mit Strukturreduktion und solche, die sich mit Struktursynthese arbeiten, unterscheiden¹.

In neuerer Zeit wurden verschiedenen Möglichkeiten gesucht, um mit neuen Rechenverfahren evidente Probleme des deterministischen Ansatzes zu überwinden. Dessen Probleme liegen z. B. in der exponentiellen Abhängigkeit der Anzahl der Konstitutionsisomere einer Summenformel von der Anzahl der Schweratome¹ in derselben. Diese exponentielle Abhängigkeit macht es ab einer bestimmten, von der Effizienz des verwendeten Systems abhängigen Schweratomzahl unmöglich, den Konstitutionsraum noch vollständig zu untersuchen.

Hierbei darf man sich nicht von der Größe solcher Molekülen beeindrucken lassen, die immer wieder als mit deterministischen Verfahren behandelt beschrieben werden. Zum einen werden dort z. B. spektroskopische Verfahren verwendet, die einem durchschnittlichen Labor in der Regel nicht zu Verfügung stehen (z. B. 1,1-ADEQUATE oder INADEQUATE) und die durch die Vorhersage von zahlreichen direkten C-C-Bindungen den zu durchsuchenden Konstitutionsraum auf das

¹ Unter Schweratome verstehen wir alle Nicht-Wasserstoff-Atome – ein Terminus, der sich eingebürgert hat, weil die Wasserstoffatom in der Regel als den Schweratomen inhärent zugeordnet behandelt werden und nicht in die Kombinatorik des Strukturgenerators eingehen.

behandelbare Maß zusammenschrumpfen lassen. Zum anderen werden z. B. in einem prominenten kommerziellen System mit Hilfe einer großen, proprietären Struktur-Spektrendatenbank und dem Spektrensatz des CASE-Problems sehr große Strukturfragmente vorhergesagt, die dann ebenfalls die Kombinatorik des Strukturgenerierungsprozesses stark vereinfachen. In diesem letzteren Fall muss eingewandt werden, dass die Generierung von Goodlist-Fragmenten mit einem solchen wissensbasierter Ansatz stets die Gefahr birgt, dass a) die Datenbank das tatsächliche zu einem bestimmten spektralen Muster gehörige Strukturfragment nicht enthält und man dann wieder vor dem gleichen kombinatorischen Problem steht wie ohne Datenbank, und dass b) durch einen Datenbankfehler falsch-positive Treffer für die Good-List gefunden werden, die dann den ganzen Strukturaufklärungsprozess zum Scheitern verurteilen.

Aus diesem Grunde wendet sich diese Arbeit einer Alternative zu deterministischen Suchverfahren, den stochastischen Optimierungsmethoden, zu. Diese haben in anderen Bereichen des naturwissenschaftlichen Rechnens (Astronomische Vielteilchensimulationen, etc.) gezeigt, dass sie zur Suche nach dem Optimum in sehr großen Lösungsräumen befähigt sind. Sie sind außerdem, wie gezeigt wird, bei gutem Design der Zielfunktion sehr fehlertolerant und kommen ohne oder mit sehr wenig empirischem Wissen aus. Nachdem Steinbeck ⁵ erfolgreich die Verwendung des stochastischen Suchverfahrens Simulated Annealing ⁶ in der automatischen Strukturaufklärung demonstriert hat, wurden in der vorliegenden Arbeit evolutionären Methoden zur Anwendung gebracht, die sich als noch deutlich effizienter als der von Steinbeck zunächst publizierte Ansatz erwiesen haben.

Evolutionäre Verfahren zur Automatischen Strukturaufklärung

Evolutionäre Optimierungsverfahren nutzen eine Analogie zwischen der Entwicklung der natürlichen Spezies in einem Zyklus von Mutation, Fortpflanzung und Selektion gemäß der Darwin'schen Theorie und der Suche nach einem globalen Optimum in einem allgemeiner gehaltenen System aus. Man unterscheidet hier genetische und evolutionäre Algorithmen sowie genetische und evolutionäre Programmierung.

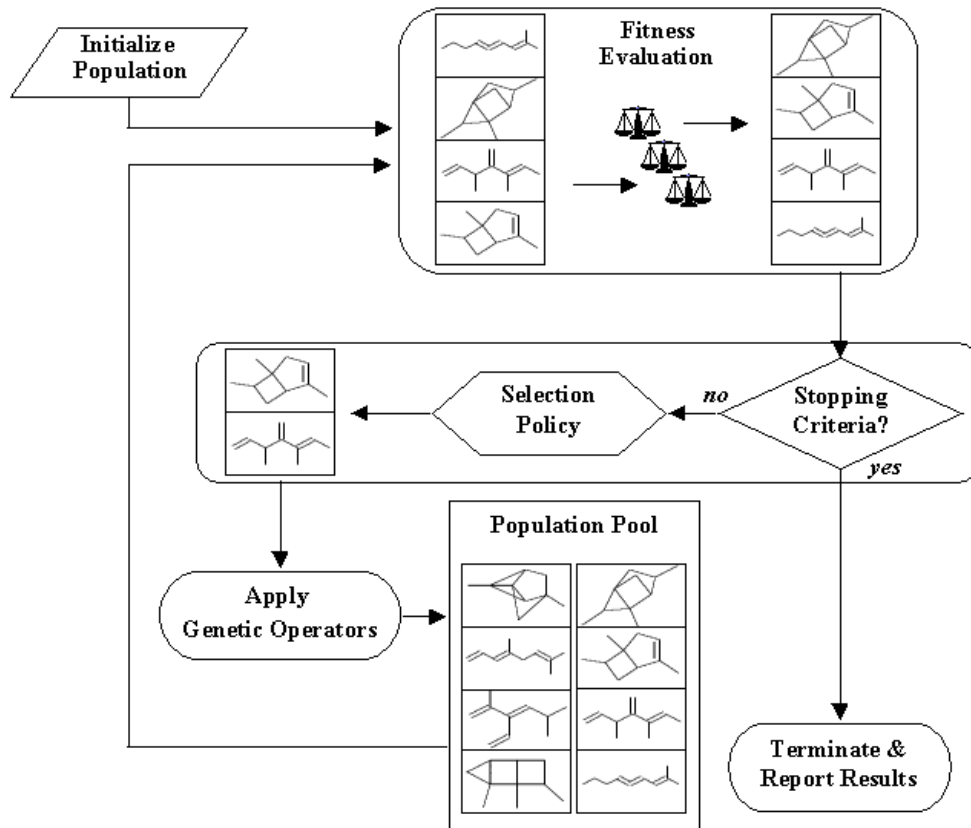


Abbildung 1: Evolutionäre Suche in einem Einzelpopulationsansatz

Zur erfolgreichen Anwendung eines solchen evolutionären Ansatzes bedarf es mehrerer Komponenten, von denen einige problemspezifisch sind, andere aber theoretisch problemunabhängig implementiert werden können. Neben einer Datenstruktur zur Kodierung des Genotyps (potentiell problemunabhängig, s.u.) und einer Methode zur Konversion des Geno- in den Phenotypus (problemabhängig) benötigt man eine Ziel-, Bewertungs- oder Fitnessfunktion, um während des Evolutionsprozesses die Eignung von Kandidaten zu überprüfen.

So sieht z. B. die Standardtheorie der o. g. Verfahren oftmals die Kodierung der Datenstruktur als lineare Datenstruktur in Form etwa eines Bitstrings vor. Auch chemische Strukturen lassen sich leicht in eine solche Form bringen, z. B. indem man für eine der signifikanten Hälften der Konnektivitätsmatrix alle Spalten oder Reihen aneinander hängt.

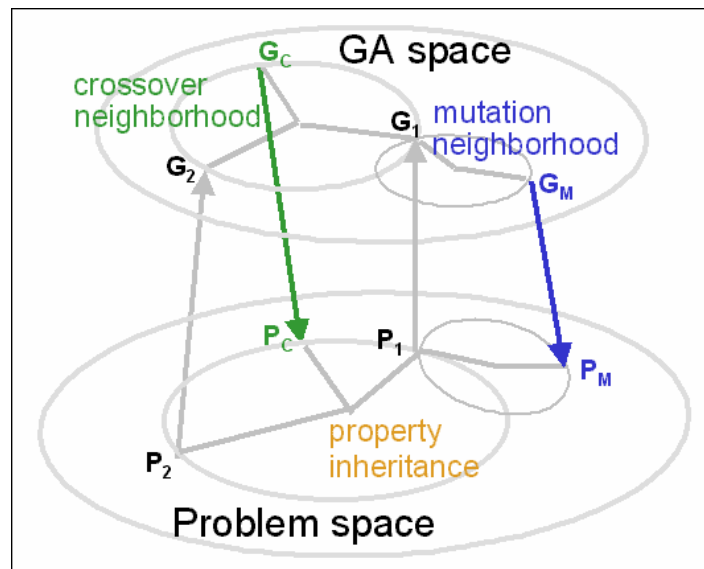


Abbildung 2: Durch die Verwendung eines phänotypischen Datentyps, auf dem Mutations- und Crossover-Operatoren arbeiten, wird eine Projektion des Phänotypes aus dem GA-Raum in den Problemraum notwendig.

Der so entstehende Genotyp lässt sich offensichtlich einfach mit Hilfe geeigneter Programmmodule wieder in den für den Chemiker besser verständlichen Phänotyp des 2D Strukturdiagramms überführen. Auf dem besagten Genotyp lassen sich ebenfalls besonders einfach die bekannten genetischen Operationen Mutation und Crossover anwenden. Bei näherer Betrachtung erschließt sich jedoch umgehend, dass diese Operationen in der Regel in chemisch invaliden Resultaten führen. So wird beispielsweise bei einer Punktmutation einer „1“ zu einer „0“ in der linearen Bindungsmatrix eine Bindung zwischen den beiden beteiligten Atomen gebrochen, ohne dass für eine Sättigung der entstehenden offenen Valenz gesorgt wird. Ein 2-Punkt-Crossover führt in der Regel zu ähnlich fatalen Ergebnissen. Bisherige Ansätze zur Anwendung von EA/GA in der Konstitutionsaufklärung haben diese Methode der Genotyp-Manipulation verwendet und die entstehenden chemisch-invaliden Strukturen mit Hilfe der Fitnessfunktion ausselektiert. Offensichtlich wird dabei für ein Problem mit ohnehin großem Rechenaufwand eine gehörige Menge von Rechenzeit zur Eliminierung von ungültigen Datenstrukturen verwendet, bevor man überhaupt zur Optimierung in Richtung des eigentlichen Evolutionsziels kommt.

Aus diesem Grunde wurde in der vorliegenden Arbeit eine graphen-basierte, objektorientierte Kodierung für chemische Strukturen und ein Set von robusten Mutations- und Crossover-Operatoren entworfen, die während der Operation sicherstellen, dass aus einer chemisch validen Ausgangsstruktur auch eine ebensolche Zielstruktur entsteht. Der Mutationsoperator wurde als eine Erweiterung der von Faulon vorgeschlagenen ⁷ und von Steinbeck zur Strukturaufklärung erfolgreich eingesetzten ⁵ Modifikationsoperation implementiert (Abbildung 3).

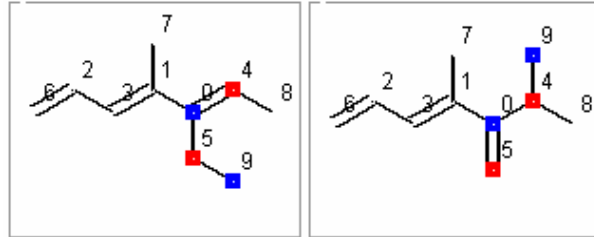


Abbildung 3: Arbeitsweise des Mutationsoperators. Nach zufälliger Selektion von vier Atomen werden durch Anwendungen eines einfachen Gleichungssystems unter Erhalt der Summe der Bindungsordnungen jedes Atoms Bindungen gebrochen und neu geknüpft, so dass eine Konstitutionsisomer der Ausgangsverbindung entsteht.

Der ursprünglich unbeschränkte Wirkungsradius des Operators ist jedoch jetzt je nach Anforderung während des Evolutionsprozesses auf bestimmte Molekülteile beschränkbar. (Abbildung 4).

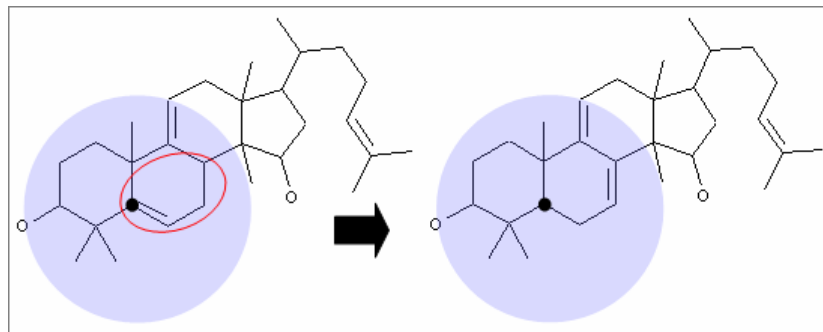


Abbildung 4: Beschränkung des Wirkungsradius' des Mutations-Operators

Der Crossover-Operator, der wie üblich aus zwei Elternteilen durch Zerschneiden und Neuzusammenfügen zwei Nachkommen erzeugt, sucht sich zunächst für jede zu zerlegende Kandidatenstruktur durch Tiefen- oder Breitensuche² einen zusammenhängenden Molekülteil konfigurierbarer Größe und trennt diesen dann durch Bindungsbrüche vom Rest des Moleküls.

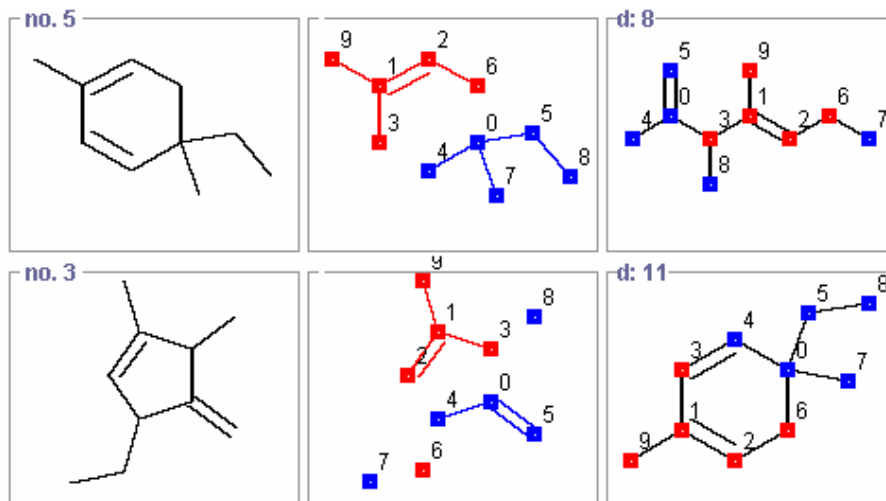


Abbildung 5: Arbeitsweise des Crossover-Operators.

Wenn dies für jedes der Elternmoleküle geschehen ist, werden die Einzelteile kreuzweise ausgetauscht und durch systematisches Einsetzen der fehlenden Bindungen zu neuen Nachkommen wieder zusammengefügt (Abbildung 5).

² Je nach Verwendung des einen oder des anderen Suchverfahrens ergeben sich unterschiedliche Verhaltensweisen beim Evolutionsprozess

Als Zielfunktion wurde in der vorliegenden Arbeit die bereits von Steinbeck vorgestellte Zielfunktion verwendet. In dieser Zielfunktion wird die Kompatibilität einer jeweiligen Kandidatenstruktur mit einem Satz von Erwartungswerten getestet.

$$E_{tot} = E_{HBMC} + E_{HHCOSY} + E_{Shift} + E_{Symmetry} + \dots + E_{Features} \quad (1)$$

Dies sind entweder tatsächlich gemessene Spektren (1D ^{13}C NMR, HHCOSY, HMBC, HSQC) oder aus Spektren oder anderen Informationen abgeleitete Randbedingungen (Substrukturen, erlaubte oder verbotene Ringgrößen). Für jede der Randbedingungen trägt ein additiver, skalierbarer Term zum Gesamtwert der Fitnessfunktion bei (Gleichung (1)).

Resultate

Die Leistungsfähigkeit der entwickelten Methoden wurde an Beispielen getestet, die sich in der CASE-Literatur als Benchmarks etabliert haben, d. h. für die sowohl der Satz an korrekten Lösungen als auch Bearbeitungszeiten anderer Programme bekannt sind. Um eine qualitative Evaluierung der Skalierung der Rechenzeit mit wachsender Größe und ein Vergleich mit der bereits existierenden Simulated-Annealing-Implementierung vornehmen zu können, wurde die in ⁵ verwandte Reihe von Terpenen wachsender Größe herangezogen (Abbildung 6).

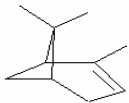
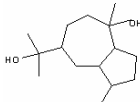
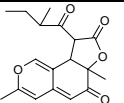
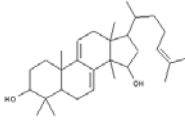
Struktur	Summenformel	Besuchte Punkte vs. Suchraumgröße	Benötigte Generationen	Zahl der Lösungen	Rechenzeit
	$C_{10}H_{16}$	48 / 4,305	6	1	< 1 s
	$C_{15}H_{28}O_2$	2,088 / ?	60	1	40 s
	$C_{18}H_{20}O_5$	2,600 / ?	70	1	120 s
	$C_{30}H_{48}O_2$	6,400 / ?	48	6	120 s

Abbildung 6: Laufzeit von EA Optimierungen an trukturaufklärungsproblemen wachsender Größe. Die Messungen wurden auf einem Standard-PC mit Pentium III Prozessor (500 MHz, 256 MB RAM) durchgeführt. Die sechs Lösungen, die im Falle des Triterpens Polycarpol gefunden wurden, sind der vollständige mit dem Spektrensatz in Einklang stehende Satz von Lösungen, wie durch Rechnungen mit dem deterministischen CASE-Programm LUCY 8 verifiziert wurde.

Für alle Beispiele lagen wenigstens die Information aus 1D ^{13}C Spektren (BB, DEPT135, DEPT90) zum Erstellen einer vollständigen Bestandsaufnahme der CH_x -Fragmente durch das Programm, sowie die mehrdeutigen, weitreichigen HMBC-Korrelationen vor. Letztere wurden vom betreffenden Bewertungsmodul entweder als

$^2J_{CH}$, $^3J_{CH}$, oder mit geringerer Bewertung auch als $^4J_{CH}$ -Kopplung interpretiert. HMBC-Information kann im System nur bei gleichzeitigem Vorliegen der $^1J_{CH}$ -Informationen aus beispielsweise HSQC-Spektren verwendet werden.

Zusätzlich wurde ein Bewertungsmodul zur Rückrechnung von ^{13}C -Spektren eingesetzt, das auf der Basis von HOSE-Codes ⁹ eine Einschätzung erlaubt, ob alle Atome in der Kandidatenstruktur die korrekte Hybridisierung besitzen.

Wenn HH COSY Informationen vorlagen, wurde diese, beispielsweise beim Polycarpol, auch zur Strukturaufklärung herangezogen. Während sich bei kleinen Strukturaufklärungsproblemen, wie z. B. beim α -Pinen, oftmals schon alleine aus dem mehrdeutigen HMBC-Spektrum eine *einzig*e Lösungsstruktur ergibt, benötigt man bei wachsender Problemgröße alle verfügbaren Informationen, um den Lösungsraum auf ein erträgliches Maß einzuschränken.

Abbildung 7 zeigt Fits der für die Lösung des α -Pinen-, Eurabidiol- und des Polycarpol-Datensatzes benötigten Zeit gegen deren Molekülgröße für drei verschiedene CASE-Algorithmen. Als Vertreter eines deterministischen Verfahrens wurde LUCY ⁸ verwendet. Die Leistung des hier beschriebenen EA-Algorithmus wurde weiterhin mit dem in ⁵ veröffentlichten Simulated Annealing (SA) Verfahren verglichen. Deutlich ist zu sehen, dass bei exakt gleicher Datenlage die beiden stochastischen Algorithmen deutlich besser skalieren und der hier beschriebene Evolutionäre Algorithmus favorisiert werden muss.

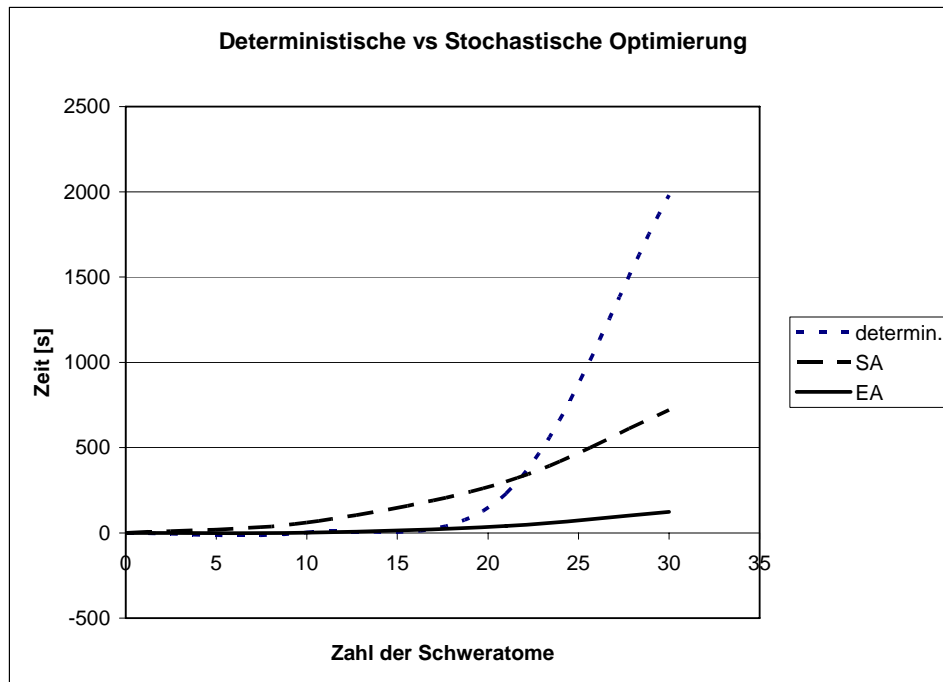


Abbildung 7: Skalierung der Rechenzeit unserer deterministischen und stochastischen Implementierungen mit wachsender Atomzahl im zu findenden Molekül. Während bei der Optimierung mit LUCY (deterministisch) die Tendenz zur kombinatorischen Explosion zu erkennen ist, skalieren Simulated Annealing (SA) und Evolutionärer Algorithmus (EA) deutlich günstiger.

Literatur

- (1) Munk, M. E. Computer-Based Structure Determination: Then and Now. *JCICS* **1998**, *38*, 997-1009.
- (2) Steinbeck, C. Computer-Assisted Structure Elucidation. *Handbook on Chemoinformatics*; Wiley-VCH: Weinheim, 2003; pp 1378-1406.
- (3) Steinbeck, C. Correlations between Chemical Structures and NMR Data. *Handbook on Chemoinformatics*; Wiley-VCH: Weinheim, 2003; pp 1368-1377.
- (4) Steinbeck, C. The Automation of Natural Product Structure Elucidation. *Curr. Opin. Drug Discovery Dev.* **2001**, *4*, 338-342.
- (5) Steinbeck, C. SENECA: A platform-independent, distributed, and parallel system for computer-assisted structure elucidation in organic chemistry. *JCICS* **2001**, *41*, 1500-1507.
- (6) Kirkpatrick, S.; Gerlatt, C. D. J.; Vecchi, M. P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671-680.
- (7) Faulon, J. L. Stochastic generator of chemical structure .2. Using simulated annealing to search the space of constitutional isomers. *JCICS* **1996**, *36*, 731-740.
- (8) Steinbeck, C. Lucy - A Program For Structure Elucidation From NMR Correlation Experiments. *Angewandte* **1996**, *35*, 1984-1986.
- (9) Bremser, W. HOSE - A Novel Substructure Code. *Analytica Chimica Acta* **1978**, *103*, 355-365.

Chapter 1: Introduction

The goal of computer assisted structure elucidation (CASE) is to find, within a given solution space, the single structure which best fits a set of chemical and spectral boundary conditions. In its most general form, the structure elucidation problem is defined as follows: given structural information of an unknown compound derived from chemical and/or spectral evidence, find out the fittest structure formula that satisfies all of these constraints. The input information consists of molecular formula derived from mass spectrometry or element analysis, and routine 1D and 2D NMR spectral data (^1H , ^{13}C , DEPT, INADEQUATE, COSY, HMQC, HMBC, NOESY, HSQC, COLOC and other HETCOR experiments).

Computer-assisted structure elucidation of organic molecules has been studied from 1977¹. The past 20 years witness the great development of CASE methods both in the NMR spectroscopic techniques and software applications^{2,4,8,10-46}. Yet the problem still attracts interests of chemists and spectroscopists. This section is not to outline a review of available products that could be found in literature, but rather to re-clarify the problem and focus on the underlying methodology of the diverse approaches.

The mainstream strategy of structure elucidation classifies the procedure in three steps^{2,4,13,47}: 1) pre-processing of substructure information and preparation of constraint conditions; 2) exhaustive and irredundant generation of all candidate structures in agreement with the constraints above; and 3) Spectrum prediction and comparison evaluating their relative probability of being correct.

Most CASE programs available include software modules for the three major components of structure elucidation. The starting point for the structure elucidation is

molecular formula derived from MS, 1D and 2D NMR spectra. The collective spectral information is interpreted as a set of substructures predicted to be present or absent in the unknown. The deduced information, together with its molecular formula, is the usual input in structure generation. A high-quality reference library containing both structures and complete spectra or substructures and subspectra being representative of the types of compounds encountered in the laboratory, is an invaluable component for a CASE system^{12,16,21,27,48-50}.

The premise implicit in the spectrum interpretation is that if the spectrum of the unknown and a reference library spectrum have a subspectrum in common, then the corresponding reference substructure is also present in the unknown²¹. Note that the fragment types are limited to those present in the database used, therefore the output isomers are limited to those structures that can be assembled from the substructure library of each program. While 1D NMR information has been widely used in deterministic search based CASE products, the interpretation and utilization of 2D NMR is not well utilized¹³. There are some highly ambiguous information such as the HMBC-derived long-range C-H correlations that do not distinguish between two and three (sometimes four) intervening bonds^{11,12}.

The components generated by spectra interpretation are fed into the structure generator, which will exhaustively generate all possible structures from these components. Although many structure generators have been reported, the underlying paradigms fall into one of two classes: structure assembly or structure reduction⁴⁷. The structure assembly can be described as a procedure to systematically search for all valid interconnections between the residual bonding sites on the inferred substructures and on the unaccounted for atoms. The process can be viewed as the expansion of a partial structure to all complete molecular structures compatible with it.

One of well-known structure assemblers is MOLGEN^{26,51}. In MOLGEN, there are three types of constraints that can be entered optionally, namely:

Macro atoms The most important substructures are macro atoms, which mean substructures that are not allowed to overlap. The use of macro atoms is very important, since they may reduce the work of the generator tremendously.

Good list The second type substructures form the so-called good list; they may overlap. This list is applied as a filter after the generation process.

Bad list The forbidden substructures form the so-called bad list. This list is used in the analogous way as a filter following the generation.

The application of MOLGEN in molecular structure elucidation stands or falls with the input. The main emphasis lies on the macro atoms, since a big set of prescribed and non-overlapping substructures reduces the problem of generation considerably, while the good list and the bad list can be applied only after structures were suggested.

The generated structures are checked for consistency with the spectra data^{13,21,25,29,48,52-65}. Generated substructures can be checked in the course of structure generation (prospective checking) or after all complete structures have been generated (retrospective checking). Clearly, prospective checking is faster as those substructures that are not consistent with the spectra data are removed from the structure generation process. Whereas in the case of retrospective checking, a combinatorial explosion occurs for exhaustive structure generation, even for molecules of a moderate size. Another very important aspect of a structure generator is so-called isomorphism check, assuring that it is not producing the same structure more than once^{24,66,67}. At the end of all the procedures, the complete set of structures that are consistent with all the spectra data should have been generated.

What Makes CASE Problems Hard? In structure elucidation problems, the spectral evidence available is often insufficient to permit a structure to be proposed. While spectrum interpretation provide valuable structure information, it is practically impossible to extract all of the useful information ⁶⁸. This requires the systematic search to be run in a huge space, and the information content of the substructure inferences is often insufficient to lead directly to the generation of a single structural assignment.

A structure elucidation problem is equivalent to a combinatorial optimization problem if the spectra-based structural information of the unknown is treated as constraints to be satisfied. The central task is thus to prune the size of the search space to a computationally acceptable extent. The methods mentioned above attempt to reduce the size of the search by taking advantage of problem-specific information. Nevertheless, pruning heuristics are not always enough because the incompleteness of chemical and/or spectroscopic evidence. And the existence of vague information makes the actual search space expand drastically.

The search space itself is a discrete one. How the solutions are distributed in the space is not unveiled. The introduction of structural constraints into the search space makes its structure even less organized ⁶⁹⁻⁷⁵. When many constraints are added, traversal of the search space is confounded.

This work reports a new evolutionary optimization strategy tailored for chemical structure elucidation of organic compounds ⁷⁶. This algorithm contributes a graph-based data structure and a suite of robust graph operators. The labeled molecular graph data structure facilitates efficient genetic manipulation and exempts from the transformation between genotype and phenotype of the candidate solution. Flexible parameter control strategy enables the genetic operators to adjust their behavior and achieve higher search efficiency.

An evolutionary algorithm framework is implemented serving as a search engine in the SENECA system, a distributed, platform-independent CASE program⁷⁷. This EA implementation integrates the three steps in a CASE expert system – structure inference, structure generation and structure verification – into one procedure. Based on this framework, different EA formulations and schemes can be easily configured for problems of specific characteristics. The framework also serves as a test bed and a class library for other forms of chemical constitution optimization; new optimization task can be carried out by this framework with little further coding work by the end user.

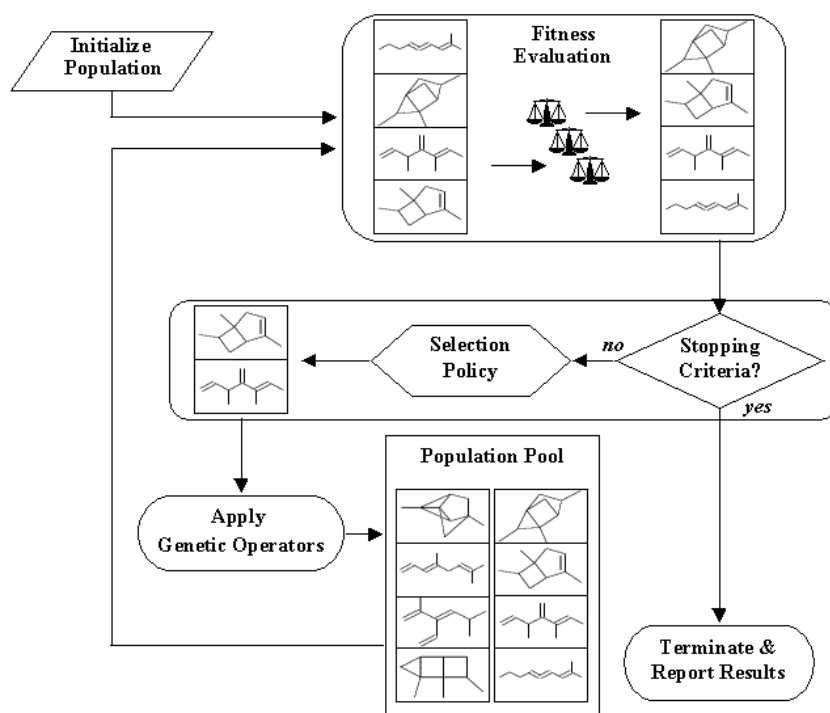


Figure 1: Evolutionary search flowchart for a single-population based EA scheme (after⁷⁸).

Any evolutionary algorithm should at least include an appropriate representation of the plausible solution to the problem under investigation, a fitness function scoring the candidates, a set of reproduction operators prompting evolution, and strategies of initialization, selection, and termination. Among them, the two major components - the specification of the representation and the definition of fitness function, form the bridge between the original problem context and the problem-solving algorithm. Figure 1 illustrates the workflow of an evolution scheme based on a single-population formulation. Chapters 2-9 detail the methods and implementations of the purposed EA framework. Chapter 2 justifies the evolutionary algorithm based strategy for chemical structure elucidation problem. Chapter 3 discusses appropriate data structure representing the solution to the constitutional optimization problems. Chapter 4 purposes a suite of robust genetic operators devised according to the central data structure adopted in this framework. Chapter 5 issues the construction of fitness functions and the influence of fitness landscapes to evolution routes. Chapter 6 compares different selection policies. Chapter 7 analyses three population management strategies. Chapter 8 exemplifies several evolution schemas bearing self-adaptive mechanism. Chapter 9 summarizes the results of structure elucidation problems solved by this EA approach. Chapter 10 draws conclusions on the evolutionary search strategy for to chemical constitution optimization, and outlines some directions of further work.

Chapter 2: The Evolutionary Search Approach

2.1 Characteristics of the Constitution Space

Structure elucidation of an unknown compound, based on knowledge on its molecular formula, spectral data and other prior information is a process of searching the best-matched constitutional formula among usually isomeric structures. The constitutional isomers for a given molecular formula constitute the so-called constitution space - an assembly of finite-numbered isomers. How these isomers are distributed in such a discrete space is not unveiled. An arbitrary structure at one point in the constitution space holds little information about its neighbors. Due to the implicit existence of constitutional constraints, an indiscriminate tiny substitution in one structure is liable to result in an ill-formed structure. The amount of the isomers is usually large, and the size of the constitution space expands exponentially proportional to the number of skeletal atoms in the molecule of the unknown. The complexity of the constitution space makes its exploration difficult and appeals elaborate strategies ².

2.2 Deterministic and Stochastic Approaches

Two strategies for searching constitution spaces – deterministic as well as stochastic approaches – have been described in the literature ^{1,10,27,77,79}.

A deterministic approach makes an exhaustive search in constitution space, thus guarantees the optimum to be found if the algorithm has enough time to finish its job and if the constraints are error-free. This paradigm has been the first (and maybe the

only) choice in previous CASE approaches and utilized in the CASE software products in different ways ¹.

Deterministic search is very efficient for structure elucidation of small molecules, while its major drawback is its deadly demand for computing power in the case of large molecules due to combinatorial explosion. Much effort has thus been devoted to decrease the search space before structure generation and evaluation. A key measure is trying to deduce as much structural constraints from the spectral evidence with the help of previous knowledge accumulations ^{1,80}. Libraries of structural fragments connected to certain spectral features are a typical way to drastically decrease the size of the search space by combining nodes from the atom set into larger fragments, with those atoms thus no longer taking part in the combinatorial process. In case of proton rich compounds, large numbers of constraints derived from 2D NMR long range correlations also cut down the search space significantly ^{2,3}. It was reported that the search space reduced 99.9% after spectra interpretation ¹², however, due to the fact of incompleteness of spectra evidence, the narrowed search space of a large molecule (Mass > 600) is still too large to be systematically explored in a reasonable time scale.

Another problem faced by a deterministic approach is that the correctness of the inferred structural constraints must be guaranteed. The existence of a false constraint leads to the search falling in a wrong part of the constitution space and no solution will be found.

The quality of the knowledge base invoked by a CASE system also plays important role. The spectra interpretation based on database search could fail to identify a substructure if it is not included the knowledge base. This case is not unusual in a chemistry or spectroscopy lab.

The second approach for constitutional optimization is the stochastic method. In contrast to the deterministic method, a stochastic approach runs a randomized but

guided search in constitution space. Starting with one or more initial structures, it evolves candidates into ones with more desired properties. A stochastic search may locate its target quickly, as it bypasses the combinatorial explosion by focusing on most profitable (and remarkably small) regions in constitution space while not entirely neglecting the others. And a stochastic method relies little on extra knowledge bases compared to an efficient deterministic approach, because the evaluation of the candidate solution could be purely based on the experimental evidence and no information is misused.

Due to its random nature, however, a single stochastic optimization may not guarantee to find the global optimum at all, which is likely to be one of the reasons why, despite its conceptual advantages over the deterministic approach, the stochastic method is not generally realized as the method of first resort in CASE applications. The danger of not finding the optimum in a CASE problem is usually regarded a serious problem, since the user is interested in the one, single correct structure and not just a good one that reasonably fits some given constraints. We can show, however, that at least for molecular sizes usually treated by deterministic systems, our stochastic approaches find all correct solutions within about the same order of magnitude in calculation time.

In the following chapters, we present a unique evolutionary algorithm (EA) for tackling the CASE problem. The new EA paradigm integrates the three steps in a CASE expert system – structure inference, structure generation and structure verification¹ – into one procedure. This implementation contributes a graph-based data structure and a suite of robust graph operators. The labeled molecular graph data structure facilitates efficient genetic manipulation and exempts from the transformation between genotype and phenotype of the candidate solution. Flexible parameter control strategy enables the genetic operators to adjust their behavior and achieve higher search efficiency.

2.3 Evolutionary Algorithms

An evolutionary algorithm acts as a crude version of species evolution⁸¹. It inherits from nature the principles like natural selection and survival of the fittest. A typical EA starts with an initial population of candidate solutions. Each solution is evaluated by a fitness function and assigned a value indicating its relative correctness. The population evolves over generations by applying reconstruction operators on selected solutions. The selection of solutions, allowed to survive from one generation to the next one, is biased to those with higher fitness value. The algorithm terminates when satisfying solutions are found.

Recently, evolutionary algorithms have been applied for some important chemistry problems like discovery and optimization of lead compounds, and computer assisted molecule design⁸²⁻⁸⁵. Most of these applications involve the chemical structure search in a conformational space. Yet there are few examples that evolutionary algorithms are utilized to explore the constitution space^{10,86-88}. These attempts, albeit designed for different purposes, were all unable to release the evolutionary algorithms' power due to some restrictions in their design. These approaches were restrained in traditional data structures for molecule representation using coding schemes based on strings or trees. Such representations do either not cover the entire constitution space (e.g. only no-cross-linked structure can be generated in JavaGenes⁸⁶, or are inappropriate for direct and efficient genetic operations. The corresponding genetic operators could, for example, not prevent the generation of ill-formed candidates violating basic chemical valence rules¹⁰, thus imposing a large computational overhead on their implementations, preventing them from being applied to molecules larger than, say, 20 heavy atoms. And above all, because these approaches inherit directly (and simply) from the traditional EA paradigm, they were unable to guarantee that one of the correct structures will be discovered, let alone all of them.

In order to apply evolutionary algorithms to CASE problems, however, there is no doubt that a system needs to be able to find the full set of correct solutions with a good probability in order to be accepted by the user community.

As will be shown in the section on fitness function construction (Chapter 5), the fitness function of a stochastic algorithm based on spectral data can be constructed such that the maximum possible fitness value of the final target is known. The awareness of its target value allows an evolutionary algorithm to detect whether or not it is being trapped in local optimum. In this case it can take measures by automatically modifying its parameters to escape the local optimum. Taking the target value as a stopping criterion, a carefully designed evolutionary algorithm should thus be able to find at least one structure complying with all input constraints.

Like other stochastic methods, evolutionary algorithm may not assure that no structure equally compatible with the input has been overlooked in just one run. But as have been observed, the hit structures fully complying with the input constraints are of great chemical similarity, and the optimal structures are located in a small region, a niche, in the constitution space. New correct structures, if they exist, are to be screened out by applying a niche search around each of the known correct structures. The niche search makes the hit list complete with small extra computing effort which is affordable even for large organic compounds.

In a deterministic search based CASE system, it is often necessary to have a verification step for hit structures, through spectra prediction and comparison, so as to identify the fittest structure among the solution set. Our evolutionary algorithm strategy suggests that this step is in no need when the algorithm is able to perform the spectra prediction implicitly during evolution run, by including spectra prediction as part of the fitness function. A HOSE-code-based^{9,89} NMR shift prediction module has been employed in this strategy. This HOSE code evaluation function is under continuous development to achieve higher resolving power for chemical

environments of carbon atoms in the questioned molecule, coupled with the open-source and open-access web database NMRShiftsDB^{49,90,91}.

2.4 Customizing Evolution Schemes

One of the targets of this work is to realize the evolutionary strategy and establish a framework for CASE problem-solving. The framework should provide a collection of evolutionary methods and plenty of strategies of hybridization which are general enough for problems of other realms related to exploration of the constitutional space. It should be more practical than an algorithm library. With a standard representation and uniformed interface, users can design their own evolution schemes and are free to any meaningful arbitrary control strategy.

What makes this problem a challenge?

1. We need to find a highly efficient encoding scheme for chemical structure representation. In traditional EA paradigms genetic operators act on the genotype, i.e., they blindly manipulate the string or trees representing the problem. This is not necessary; one alternative is discarding genotype representation and operating on phenotype directly. The complicity of the genotype space search might be dodged this way.
2. We need a close understanding and an appropriate representation of the fitness landscape in this discrete space. How to evaluate the structural similarity of two structures in constitutional space? Can we draw a link between the fitness landscape and the problem space?
3. We need to tailor genetic operators which are in accordance with graph-based structure representation. If the genetic operators are constructed in a traditional way,

we can not guarantee the offspring are always valid structures as the problem space is discontinuous, and the EA performance will be retarded dramatically.

Chapter 3: Representation of Candidate Solutions

3.1 Basic Consideration

The choice of representation for candidate solutions has a crucial impact on the EA performance.

Traditionally, evolutionary methods do not handle the problem space search directly. This is because the candidate solution was usually encoded into a simple data structure such as tree or (bit-) string. The genetic operations are conducted in genotype space while the corresponding evaluations of candidate solutions are conducted in real problem space (phenotype space). A procedure of mapping transformation is needed to bridge these two data types, as shown in Figure 2.

Although much of its literature has focused on bit representations, evolutionary algorithms can operate on any data type. It is always welcomed to have a natural representation of the candidate solution so the evolutionary search can be done directly in problem space which will exempt from the computational cost for genotype-phenotype mapping transformation. Whatever form of the data structure is employed, a representation must have appropriate genetic operators defined for it. The representation specifies the realm of the search space, and the operators determine how the space is explored.

For an efficient evolutionary algorithm, the internal data structure should be a precise and complete expression of a solution to the problem. A well-designed representation contains all and only the information needed to represent a solution to the problem. If a data structure can represent an infeasible solution, the search space will be larger

than necessary because care must be taken in the fitness function sufficiently penalizing it for being infeasible. In general, it is much more desirable that the fitness function measures only optimality, not feasibility.

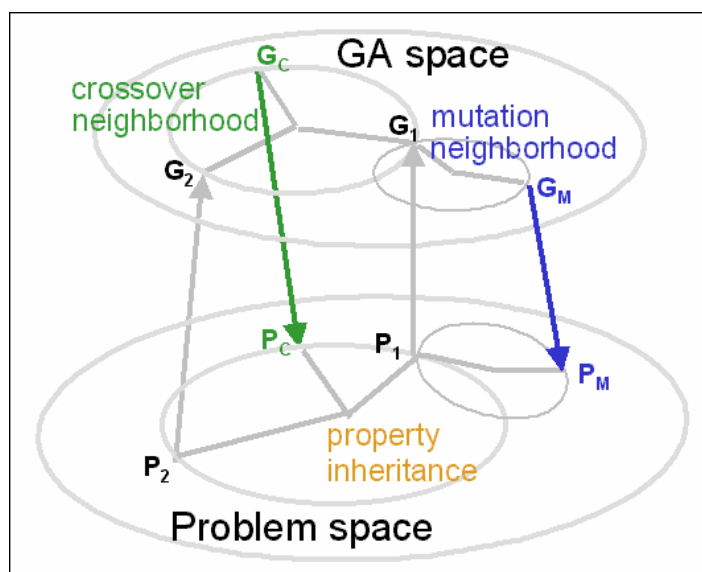


Figure 2: The mapping transformation between genotype space and phenotype space and the mechanism of crossover and mutation operation

3.2 The Data Structure

3.2.1 Labeled Molecular Graph

There are a number of ways to represent a molecular structure in a computer, such as the connectivity matrix and its variants, trees, molecular graphs, structure descriptors, line notations etc. These schemes have been devised for different purposes: data

management, structure manipulation, etc. The connectivity matrix and its variants are the most common coding method for molecular structures, yet its sparse structure leads to great time and space cost (computational complexity in many cases $O(n^2)$) for storage and manipulation. Line notations, like SMILES^{92,93}, on the other hand, are very compressed exact representations for storing, retrieving and communicating constitutional information, but the manipulation of linear data structures by recombination and mutation operators is of poor efficiency. Upon careful study of the pioneering works of Globus⁸⁶, Nachbar⁸⁷ and Meiler¹⁰, we concluded that a direct manipulation of an object oriented graph representation under constant control of the basic chemical valence rules should be the most efficient way to perform genetic operations on molecular constitutional structures.

For the implementation described in the following, we took advantage of the Chemistry Development Kit (CDK)⁶⁷, an open-source Java class library for structural chemo- and bioinformatics, developed by our group and a team of international collaborators. Within the CDK, a molecular structure is coded as a set of atom objects with connectivity information stored in bond objects, all contained in a data structure called an AtomContainer. Manipulations of the AtomContainer throughout the evolution procedure keep the atom array ordered and fixed, while the bond array is varied, with the overall sum of bond orders being constant. The hydrogen atoms and hydrogen-involving bonds in the molecule are treated as implicitly belonging to certain heavy atoms – a usual procedure in constitutional chemoinformatics – and are thus not taking part in the combinatorial process. A variety of methods are provided for structure manipulation in the CDK. One can add or delete an atom or bond, modify bond order, split a structure in two, merge two fragments in one, and so on.

AtomContainer is a faithful holder of information on the molecular structure, having no limit to represent any plausible candidate structures, artificial or natural, common or unusual. Figure 3 gives some of the possible structure for the molecule formula $C_{10}H_{16}$.

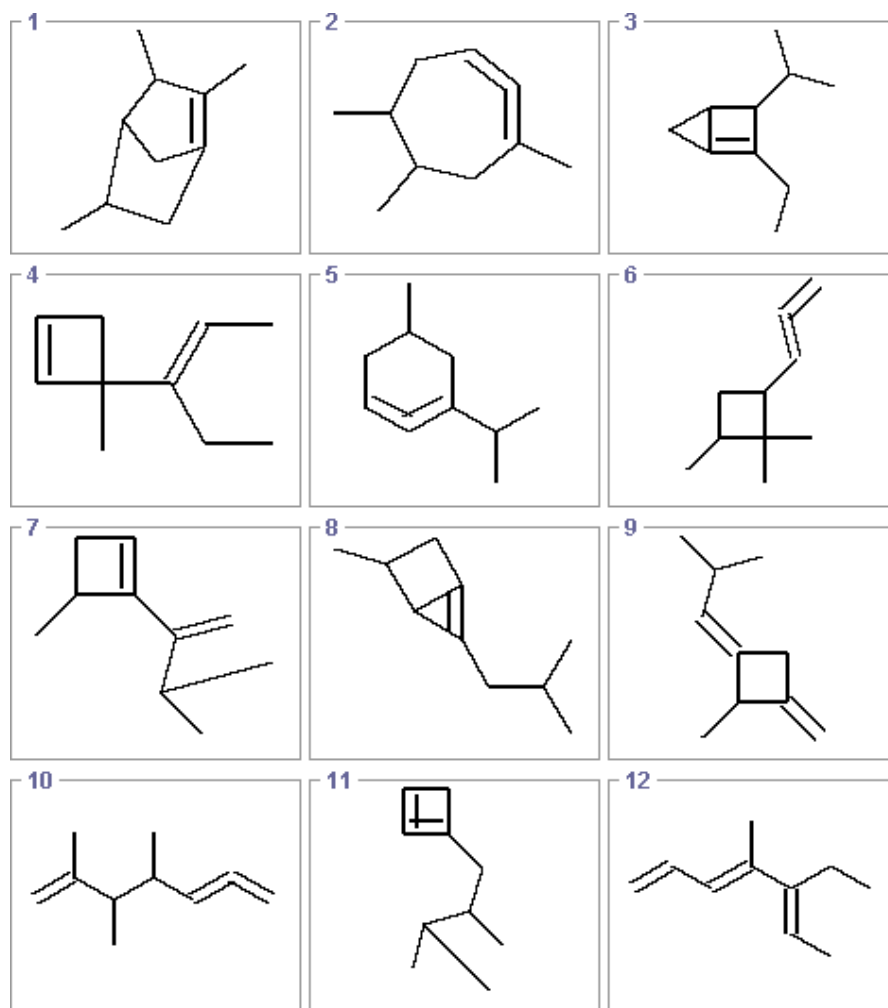


Figure 3: Different candidate solutions for the molecule formula $C_{10}H_{16}$.

In the case of CASE, there exists even more rigid demand to the data structure of the candidate solution: not only the topological distribution of the molecule structure, but also the chemical shift information of all carbon atoms should be strictly preserved. The efficiency of this data structure is augmented by adding in the AtomContainer an attribute `chemicalShift` to save chemical shift values of carbon atoms in the

unknown compound. The atoms in the atom array are sorted in a descending sequence of the chemical shifts of carbon atoms, followed by other heteroatoms. This makes the molecular structure to be a labeled molecular graph.

Carbon atoms of the same proton connections but different chemical shifts are seen as non-equivalent atoms. The traditional concept of structure isomorphism is not applicable here; a pair of differently labeled, “traditionally isomorphic” molecular graphs may have different fitness values and are thought as being positioned in distinct points in search space.

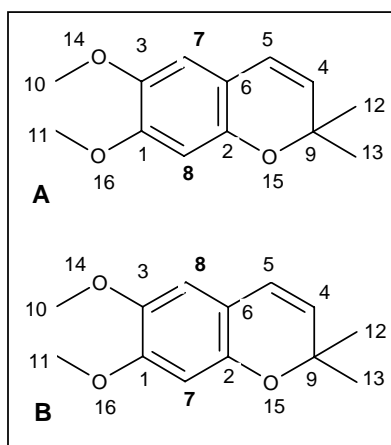


Figure 4: Two candidate structures for the molecule α -Preconene with the correct one as A. The two similar structures differ only in the labeling and consequently the carbon shifts of atom 7 and 8, as the labeling follows the descending sequence of the chemical shifts of carbon atoms in the molecule (Table 1). Structure A has a fitness score of 3000 and structure B has a score of 2900. The structure relabeler is designed to reshuffle the chemical shifts for a group of equivalent carbon atoms in the molecule preventing from the search being trapped in a local optimum.

One point should be clarified: this EA implementation is not trying to use the labeled molecular graph as the canonical one. The isomorphism problem is not solved, but got less serious for this specific application. While the introduction of the attribute `chemicalShift` does not make the labeled graph a canonical one due to at least two reasons: the possible existence of structural symmetry and the randomness of heteroatom labeling, the probability that two structures of random choice are isomorphic is decreased. Figure 4 gives an example. The two structures will be seen as isomorphic in a “normal” viewpoint. However, when the carbon atom labeling is considered, we propose they are different and as a result less effort of isomorphic check is needed to weed off duplicates in the population pool.

Table 1: Carbon chemical shifts for a sample compound with molecular formula as $C_{13}H_{16}O_3$

Carbon Number	HCount	Carbon Shift [ppm]
1	0	150.0
2	0	147.6
3	0	143.6
4	1	128.6
5	1	122.6
6	0	113.4
7	1	110.0
8	1	101.4
9	0	76.4
10	3	56.9
11	3	56.3
12	3	28.1
13	3	28.1

3.2.2 Parameters and Attributes

In order to achieve the highest efficiency of the evolutionary process, atoms in an `AtomContainer` can be tagged with particular properties.

The attribute `hCount` depicts how the hydrogen atoms are assigned to other non-hydrogen atoms. The hydrogen information is determined at the time the `AtomContainer` class is instanced according to the structural knowledge from DEPT spectra. For heteroatoms other than carbon atoms, there are multiple ways for `AtomContainer` class instancing, and each of the ways should be tested.

The attribute `atomStateTag` is a Boolean variable which tells whether an atom in the `AtomContainer` instance is active or dormant. When an atom is in dormant state, there is no way to change its bond type. A set of frozen atoms and the bonds between them forms a substructure free of breakage in the evolutionary process. A dormant atom may wake up triggered by a certain control parameter. This measure is desired in restricting the search to certain niches in constitutional space.

It is no doubt that information incorporated into representation is used far more efficiently than information classified as constraints and contained in fitness function. This graph-based representation is capable to assimilate the structural information in a dynamic way: the newly recognized structural implications are embedded into the representation in the course of evolution run. It is a strategy used in this evolutionary algorithm: embody constraints into representation, as many as possible, as early as possible. The size of constitutional space diminishes dramatically every time a new constraint is imported. An example, there are near 25,000 constitutional variants for the compound with molecular formula $C_{10}H_{16}$, while when DEPT NMR spectrum information considered, i.e. three CH, two CH₂ and three CH₃ are accepted as obligatory fragments, the isomer entries drops to 4306.

Chapter 4: Knowledge-based Structure Reconstruction

4.1 Design Principles

The exploration of the constitution space is ruled and propelled by so called genetic operation.

The operator must be able to construct any plausible structures in search space. Otherwise, "blind spots" arise in constitution space. If an optimal structure happens to be in one of these spots, then it will never be detected.

The second consideration is to deal with violation of constraints. In a careless design it is easy to find that new structures are not following the specified constraints or the chemical or physical correctness rule after genetic operations. Often it is possible to have a fitness function that severely penalizes any ill-formed structures. This is simple but expensive, as lots of resources are spent creating and then rejecting structures that do not really located in the real constitution space. Especially when the problem is of high dimensionality, almost all of the offspring structures are invalid, and so almost all the processing time would be wasted.

We prefer to make the reproduction operators aware of these restrictions and assure that each structure created can only be a valid one. This tends to be complex in design but economical in run time.

4.2 Mutation

Mutation involves one candidate structure. The new structure is generated after small change to its parent. The mutation operator acts as a refining operator in our

evolutionary algorithm strategy. This makes an important difference to the general habit in the evolutionary computation community, where take mutation as an exploration operator and the only operation to find new frontier⁹⁴.

4.2.1 Concept

Steinbeck implemented a replacement operator (suggested by Faulon⁷⁹) to guide the annealing process in a stochastic approach for chemical structure elucidation based on a simulated annealing technique¹¹. This operator perfectly meets the demand for single graph modification. For a molecular structure, the mutation operator adjusts the bond orders (with bond order 0 meaning no bond) between four arbitrary atoms while keeping the rest of the structure chemically valid. The structural validity is guaranteed by a set of valence equations (as shown in Algorithm 1). Figure 5 illustrates a mutation run on a structure with molecular formula C₁₀H₁₆. In the parent structure, atoms labeled as 0, 4, 5 and 9 are selected for operation. After mutation, the bond linking atom 5 and 9 is deleted; atom 5 connects to 0 with double bond in stead of single bond; the increased bond degree of atom 0 is balanced by lowering the order of bond coupling atom 0 and 4. A single bond is formed between atom 4 and 9 to make sure both are saturated.

1. Choose randomly four distinct atoms x_1, y_1, x_2, y_2 (two bonds)
2. Let $a_{11} = \text{Bond}(x_1, y_1)$, $a_{12} = \text{Bond}(x_1, y_2)$, $a_{21} = \text{Bond}(x_2, y_1)$, $a_{22} = \text{Bond}(x_2, y_2)$
3. Choose $b_{11} \neq a_{11}$ at random in the range $[\text{Max}(0, a_{11}-a_{22}, a_{11}+a_{12}-3, a_{11}+a_{21}-3), \text{Min}(3, a_{11}+a_{12}, a_{11}+a_{21}, a_{11}-a_{22}+3)]$ and displace a_{11} with b_{11}
4. Displace other bonds accordingly:
 $b_{12} = a_{11}+a_{12}-b_{11}$, $b_{21} = a_{22}-a_{11}+b_{11}$, $b_{22} = a_{11}+a_{21}-b_{11}$

Algorithm 1: the graph mutation algorithm

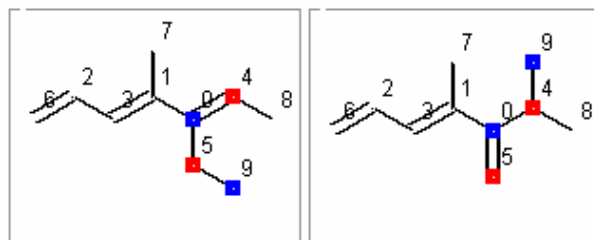


Figure 5: A mutation operation on a structure with molecular formula $C_{10}H_{16}$, with a) the parent structure and b) the offspring structure.

The mutation operator tends to do a local refinement. This process involves only four distinct atoms (the four atoms forms the working unit), and at most four bonds between these atoms are reshuffled. Thus the offspring has minimum difference from its ancestor.

4.2.2 Parameter Description

The originally proposed mutation operator is able to restrict the extent of the modification on the structure but unfortunately not the locus of the modification. To fix this shortcoming, we have added the possibility to impose further restrictions to the selection of the working atoms. A parameter `mutation radius` is defined as the maximum distance (the number of bonds) from the randomly selected seed atom to other members in the working unit. The mutation radius delimits a neighborhood in which the bond type is allowed to change. The rest part of the structure will survive to the offspring. A mutation operator leads to deeper disruption with the increase of its mutation radius. In Figure 6, the atom marked with black dot is selected as seed atom, and let the mutation radius assigned a value of 3. With this constraint, the structural reshuffle by mutation is limited inside the shaded circle, and the rest structural segment stays in tact.

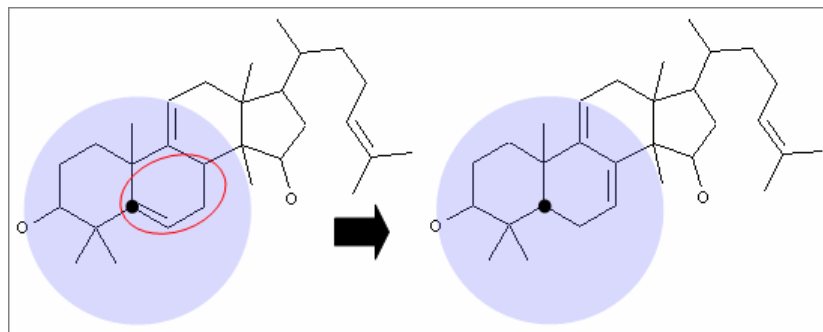


Figure 6: The effect of the mutation radius. A value of 3 is chosen in this example. The atom marked with black dot in parent structure was selected as the seed atom. The atoms within interval of three intervening bonds are qualified, and four of them (within the ellipse) are selected to take part in mutation. The structural segment outside the color-filled circle survives to offspring.

As a rule for operator design, we want to maintain the validity of offspring during reconstruction process. Therefore the mutation operator must respect the constraints employed so far. For imperative fragment constraints, we can count on changing the attribute of the parameter `atomStateTag` by freezing the atoms and bonds which form those fragments. For forbidden fragment constraints, a structural evaluation module can be embedded in the fitness function.

4.3 Crossover

4.3.1 Concept

In the context of graph recombination, crossover is the process of cleaving, merging and saturating two parent structures to form one or two children, as shown in Algorithm 2. Figure 7 gives two structures selected as parents from the population pool. The skeleton atoms of each of the parent structures are randomly segmented into two sets, red and blue. For parent 1, the red set consists of atoms labeled as 1, 2, 3, 6 and 9. The blue set contains atoms 0, 4, 5, 7 and 8. A similar portioning scheme is used for parent 2. For the crossover operation, bonds connecting two atoms in each atom set are preserved, while those connecting two atoms from different sets are deleted (see bond 3-4 and 6-0 in parent 1).

- 1. Select parent structures from population pool*
- 2. Decide cutting scale and mode*
- 3. Split parent structures in two fragment clusters each*
- 4. Partition skeletal atoms into two sets*
- 5. Preserve bond that connects two atoms in the same set, remove bond that connects two atoms from different set*
- 6. Merge the opposite cluster from each parent and obtain a pair unsaturated offspring structures*
- 7. Saturate offspring structures.*

Algorithm 2: the graph crossover algorithm

The two resulting fragments of each parent structure are now crosswise combined to form two offspring. The red cluster in parent structure 1 combines with the blue cluster in parent structure 2, and the blue cluster in parent structure 1 joins the red cluster in parent structure 2. By doing so, two incomplete structures are obtained with some atoms being unsaturated. After appending the missing bonds in the offspring structures (6-7, 0-3, and 3-8 in child structure 1, 2-6 and 3-4 in child structure 1), a pair of new valid structures is yielded.

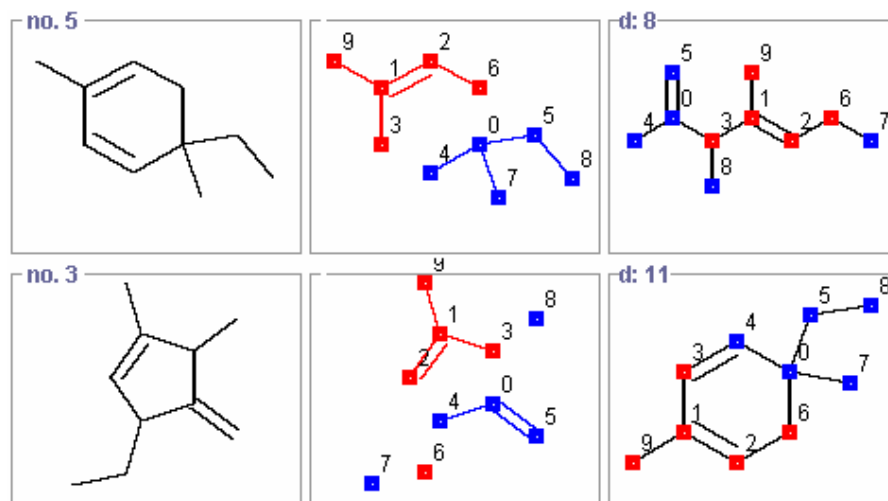


Figure 7: A crossover operation on a pair of structures with molecular formula C₁₀H₁₆. The minimum chemical distance between an offspring and its ancestor is given by the parameter d.

Each of the two offspring structures inherits certain fragments from both of its parents. The correctness in molecular formula and the connectivity of new structures is guaranteed by strictly complying with the valence rule. Because the partitioning of skeletal atoms, the assembly and saturation of the partially-filled offspring structures are in a stochastic way, all the plausible recombination are taken into consideration,

e.g. bond type, ring size etc, thus there is no possibility that a “blind spot” could appear in the constitution space.

4.3.2 Parameter Description

The crossover operator can be refined by a number of parameters: the `Match` mode defines how to select parent structures from the population pool; the `Partition` `scale` specifies at what extent the parents are allowed to spoil; and the `Partition` mode decides in which way to cleave the parent when doing the crossover operation. It has been noticed that different partition modes lead to different search inclinations.

It became necessary to introduce these parameters due to our observation that if this three step process of cleaving, exchanging and merging is performed in a totally random way, it is very likely that a large number of unconnected segments arise in the intermediate steps and some obligatory substructures are broken. It is obviously more difficult to rationally assemble these segments than dealing with a slightly damaged structure.

The initial purpose of a partition strategy is to make sure that at least one fragment cluster is inter-connected (and as a result the bonds in this cluster preserved) so that only a few fragments are generated at all. The procedure is to firstly select an arbitrary atom in the farther structure as seed atom, then starting from the seed atom, make a breadth-first or depth-first walk in the structure until the number of visited atoms reaches pre-decided limit. Atoms covered by the traverse path form one cluster. The rest atoms compose another cluster.

Breadth-first-search based partitioning tends to restrict the structure modification in a small range, and make the crossover function more locally. On the contrary, depth-first-search based partitioning enables the crossover to have global influence.

The experimental results show that the graph operators outplay standard two-point crossover. The increase of performance is visible for nearly all considered experimental settings, indicating that manipulating the structure in a way that is consistent with its representation improves the results. It is believed that these operators can be applied to other constitutional optimization problems for which the natural representation is a molecular graph. Due to the nature of our application domain being the space of constitutional isomers, we only allowed the exchange of fragments of the same sizes. However, this constraint can be removed, enabling the application to a wider variety of problems.

4.4 Niche Search

Regular evolutionary algorithm is known to be strong in global search while weak in local search⁹⁴. As a remedy, an evolutionary algorithm could bind with a kind of efficient local search algorithms. Our EA implementation described here is integrated with a genetic operator for vicinity search. For a specified root structure, the entire structures that could be generated from the root by a single mutation run constructs the `proximate neighborhood` of the root structure. Vicinity search checks every structure in such a niche and returns the best one. Vicinity search can be thought of as a self-learning process of the selected structure during its lifetime.

Within an EA run, a niche searcher is applied to a few outstanding candidates. These elite structures are intensively refined by learning from their neighbors. As a result, a gather of avant-garde is formed which is far ahead of the rest population. Via recombination and selection, the population tends to share the achievement of the model structures very rapidly. Iterating this procedure - sending elite structures ahead by niche search and then catching them up by genetic reproduction - speeds up the progress of global optimization.

4.5 Auxiliary Operators

Some supplementary features of our EA implementation include the `population filter` and the `structure relabeler`. These supplementary operators are designed to monitor the evolution status and guide the search direction.

A structure relabeler is a tuning operator changing the labels of the carbon atoms of an equivalent class. Reconsider the example given in Figure 4 in Chapter 3: Structure A, the correct one, has a fitness score of 3000, while the second structure has a score of 2900 (please refer to the score system section below). A structure relabeler operates on a candidate (e.g. structure B here) in an attempt to find a fitter structure. The structure relabeler is supposed to be activated at late evolutionary stage and work on the best structures found so far.

The population filter makes sure that copies of any structure in a population are within a threshold, called the `inhibitive value`. This simple operator alleviates the trend of premature convergence without employing difficult isomorphism check routine. It is advisable to enhance algorithms' online performance.

4.6 Customization

The obligatory constraints can be enforced by specializing in the genetic operators. For example, crossover may force a substructure intact for a given period. The genetic operators reinforce constraint satisfaction whenever possible so that ill-formed solutions are impossible to generate. As a result, the algorithm searches only in the feasible realm rather than also in the whole space of constitutional isomers. Of particular note is the ease with which various constraints can be included and/or modified.

Chapter 5: Fitness Function

5.1 Components of the Fitness Function

The fitness function in an evolutionary algorithm framework is a quantitative measure of how far a candidate structure is away from the target structure.

The fitness function is the primary place in which an evolutionary algorithm is tailored to a specific problem. Fitness function used for the evaluation of candidate chemical structure of the unknown compound is built of a suite of judges, each representing a contribution of a certain structure criterion. For NMR-based structure elucidation, the spectra judges handling data from 1D ¹³C NMR, HHCOSY, HMBC, HSQC experiments are routinely used.

Abstracted from 1D ¹³C NMR data, the chemical shift and signal multiplicity yield one-bond carbon-hydrogen coupling information. Among the 2D NMR experiments, the HMQC spectra contain one-bond C-H correlation information; the COSY spectra describe H-H correlations via 2, 3 or more bonds and the HMBC spectra 2 or 3-bond C-H correlations.

An important judge contributed to fitness function in this implementation is the `HOSECodeJudge`. The HOSE (Hierarchical Organization of Spherical Environments) code^{9,95} is a well-established method for correlation of structural properties with corresponding chemical shift values. HOSE code describes the chemical environment of a carbon atom in the molecule and contains rich information on the bonding property in its neighborhood. The `HOSECodeJudge` calculates matches between the HOSE codes for individual carbon atom environments in the

candidate structure and HOSE code environments inside the database to model the predicted shifts

Some other general-purpose constraints based on knowledge on chemical structure validity are also provided to enhance the resolving power of the fitness function. One example is the Bredt's rule judge, ensuring that no bridgehead atom in a multi-bridges ring system is involved in a double bond. Since judges all share the same API, it is easy to write and add a judge to the system to include a new kind of constraints, e.g. the constraints inferred from other spectroscopic techniques such as mass spectroscopy, IR and UV spectroscopy.

The fitness function is implemented as `ChiefJustice` which consists of a suite of `Judge` instances. Because all sorts of the available constraint information are integrated in `ChiefJustice` simultaneously, it has full advantage of the synergistic effects of their combination.

5.2 Construction of the Fitness Function

5.2.1 Standard Fitness Function

The construction of the standard fitness function is quite straightforward. Taking α -Preconene ($C_{13}H_{16}O_3$) as an example for an unknown compound, Table 2 illustrates the configuration of its fitness function. In this sample case three types of constraints - HH COSY, HMBC and HOSE code are provided, and three kinds of judges are employed - `HHCOSYJudge`, `HMBCJudge` and `HOSECodeJudge`. The ^{13}C NMR spectrum is also available, but no explicit 1D spectrum judge is needed because the information inferred from DEPT 90 and DEPT 135 experiments is digested directly by the structure representation and preserved during the evolutionary process. `HHCOSYJudge` holds 2, `HMBCJudge` 15 and `HOSECodeJudge` 13 constraint

entries. Satisfaction for a particular constraint entry leads to granting a predefined number of points for the candidate. This number can be individually configured, but is generally set to a value of 100. A fine tuning taking into account probabilities, by which a certain spectral feature is observed, can be applied to the scoring of a judge. The HMBC judge, for example, grants 100 points, if a constraint entry can be explained by 2 or 3-bond CH correlations, whereas the rarer 4 or 5-bond CH correlations yield only 5 points. Due to this scheme, the maximum achievable fitness score for a candidate structure can easily be calculated by summing up the points earned by this structure from each constraint entry in all the judges used.

Table 2: The construction of the fitness function for molecule α -Preconene

Judges	Constraint Entries	Entry Number	Maximum Score
HMBC	1-7, 1-8, 1-11, 2-5, 2-7, 2-8, 3-7, 3-8, 3-10, 4-12, 5-8, 6-4, 6-5, 6-7, 9-5	15	1500
HH COSY	4-5, 5-4	2	200
HOSE code	One entry for every carbon atom	13	1300

Target Score: 3000

In any evolutionary optimization algorithm, it is necessary for the fitness function to be able to distinguish structures even they are extremely similar, with a continuous fitness function being superior to a discrete one. For instance when the fitness function is be configured as the product instead of the sum of the scores from different judges. The fitness function gets more “continuous” than a sum-formed one

as the former has more scoring combinations. This EA implementation still has rooms to improve the means for fitness function construction. For example, more sensitive scoring criteria should be adopted – like a HOSECodeJudge based on improved multi-sphere HOSE codes – and new continuous judge types should be taken into account.

5.2.2 Advanced Assemble Strategy

It is possible to distinguish relative contributions of various judges to the best-so-far structures, and assign to each judge a weighting coefficient which determines the quota of this judge in the fitness function. A vector of weighting coefficients for all judges can be stipulated. At first the algorithm has no idea of the relative importance of each judge, so all judges have identical coefficients. After certain generations, the contributions of different judges to the fitness value are calculated. If constraints from one judge are satisfied by a higher percentage, it will be assigned with a decreased coefficient, and another poorly satisfied judge will increase its coefficient to maintain the sum of all coefficients as 1. This method is an extension of fitness scaling used in standard evolutionary algorithms. Through judge compromise a moderate fitness function is maintained which balances the search trend and avoids local convergence.

Another way making use of the weighting coefficients is to stipulate different weighting schemes for different parallel EA runs, so that each evolutionary algorithm thread follows a different route to achieve the final target. The rate of success to find all hit structures is thus improved.

Chapter 6: Selection Policy

6.1 Selection Mechanisms

Several ways of selecting which parents will be used to generate offspring to go into the next generation are commonly reported in the literature^{94,96}.

One commonly used method is fitness proportional selection (also called roulette wheel selection or Monte Carlo selection). In this method, fitness values are normalized so that each candidate is responsible for a certain amount of the total fitness in the population. The chance a candidate is selected is proportional to its fitness value. The fitness proportional selection may bring some danger of premature convergence when outstanding structures predominate in the entire population very quickly. On the other hand, when fitness values get close to each other in the population, fitness proportional selection hardly provides impetus for evolution.

When creating new population by conducting crossover or mutation to parent structure, there is a possibility that the succeeding population loses the best solution. Elitism selection provides remedy for this risk. Elitism selection first clones the best-so-far structure (or a set of best structures) to new population. The remaining part of the new population is built by classical means.

Tournament selection selects a small set of candidates and picks the best one among this set. Tournament selection has higher selection pressure compared to fitness proportional selection.

Ranking order selection ranks the population and then every candidate receives a new fitness from this ranking. The worst structure will have fitness value of 1, second worst 2 etc., and the best will have fitness N (population size). After this all candidates in the population have a chance to be selected. Ranking order selection may lead to slower convergence, because the best ones do not differ so much from others.

6.2 Fitness Scaling

Fitness Scaling is used to maintain competitive level during evolution. In early evolution stage it helps decreasing the competitive power of some supernormal candidates by reducing their fitness to prevent premature convergence. While in late evolution stage the competitive power of candidates is increased by magnifying their fitness to prevent random wandering. The fitness scaling methods can be classified into linear scaling, power function scaling, exponential scaling, etc.

Chapter 7: Population Strategies

7.1 Similarity Measures and Population Diversity

In an EA scheme, it is desirable to have a measure of the similarity between two structures. Such measure favors the detection of genetic convergence in a population, and the selection of genetically distant entities. The similarity function compares two solutions and returns a value that indicates how much the solutions differ. Often called a 'distance' function, this operator is typically used by speciating evolutionary algorithms. Many different similarity measures can be defined for any given representation.

For chemical structures, structural keys and molecular fingerprints are used as aiding selector to measure the similarity between different candidate structures in the population pool. Three types of chemical similarity measures are provided in our framework: Chemical Fingerprint, Tanimoto coefficient and Minimum Chemical Distance.

A fast way for pair wise comparison of candidate structures is using fingerprint overlap as a measure of similarity and usually calculated using the Tanimoto Coefficient. Following the notation of Daylight Systems, Tanimoto Coefficient (T_C) is defined as⁹⁷

$$T_C = B_C / (B_1 + B_2 - B_C)$$

where, B_C is the number of 1's common to both keys/fingerprints, B_1 is the number of 1's in the first key/fingerprint and B_2 the number of 1's in the second key/fingerprint. If two structures are identical, their Tanimoto Coefficient will be 1.0, and it will decrease to 0.0 as they become more dissimilar.

The selection of parent structures for crossover operation may follow different criteria. In an attempt to prevent the loss of population diversity, the parent structures could be forced to have a Tanimoto Coefficient larger than a threshold, e.g., no less than 0.5. The threshold takes a decreasing value when search is closing to the optimum.

7.2 Diversity-guided Step Size Control

Like in other evolutionary optimization techniques, there are mainly two chains in our evolutionary algorithm which determine the overall efficiency of constitutional search. One is candidate structure evaluation and ranking as discussed in Chapter 5. The other one is the mechanism to keep population diversity. Evolution tends to converge to a single solution because of stochastic noise; it is therefore necessary to avoid this with an additional mechanism. While the process of structure evaluation and ranking has less space to be accelerated, the efficient control of population diversity might be achieved by some tricks.

The major challenge for an EA approach is premature convergence. Figure 8 is a typical evolution curve of an unknown ($C_{15}H_{28}O_2$) without parameter tuning. It is shown that after around 80 generations, the average fitness of the population had become very close to that of the best candidate and the evolution almost entirely stagnated. The impact of the number of generations on the likelihood of losing population diversity was analyzed. It is expected that the decreasing size of the search step should be corresponding to the increasing resemblance of the top candidates to the optimum. If the reproduction operators decrease their step sizes faster than the rate

by which the candidates approach to the optimum, the search process may stagnate, since the step sizes become too small to make the candidates sufficiently different.

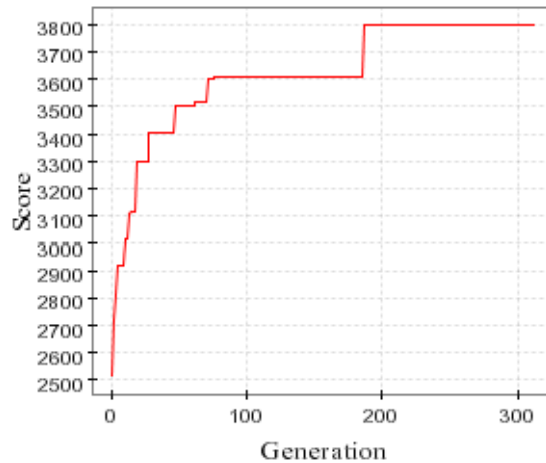


Figure 8: Search stagnates early at an average of 80 generations in the case of Eurabidiol.

Taking into account that different stages in an evolutionary search require different step size, and different fitness function components require different step size, a dynamic step size control strategy is adopted in this EA implementation, which applies the information accumulated so far in evolutionary search. A large step size encourages long-range search and makes escaping from a poor local optimum easier, while a smaller step size is apt to exploit a small region. For an unknown structure to be optimized, basically it is hard to predict when to sample a large space and when to explore a small region. We propose that the measure of population diversity and the distance of best-so-far structure to the target structure could be used to guide the trade-off between coarse-grained sampling and fine-grained exploration.

The mechanism for diversity-guided step size control is to define a metric over the constitution space and assure that the distance between any parent-offspring pair involved in a genetic operation is greater than a specified minimum threshold. This minimum threshold is subjected to change based on its judgment on current search status (e.g., how close the search is to its target) and population distribution. Several computationally affordable similarity index and fingerprint methods are used in this implementation as the metric measures. At the beginning of search, the constitutional space is sampled over a relatively coarse grid, with the mesh size equal to the metric. As the search progresses, the grid size is gradually reduced such that adjacent structures are also considered. This mechanism is a compromise between speed and efficiency, because we can only make sure that an offspring is sufficiently different from its parent. It is very expensive, if not impossible, to maintain distance threshold between all candidate structures in a population because of the intractability to obtain an analytical representation of the constitutional space.

Chapter 8: Evolution Schemes

8.1 Simple Evolutionary Algorithm

The simple evolutionary algorithm uses non-overlapping populations. In each generation, the entire population is replaced with new individuals. Typically the best individual is carried over from one generation to the next (this is referred to as elitism) so that the algorithm does not inadvertently forget the best that it found. Maintaining the best individual also causes the algorithm to converge more quickly; in many selection algorithms, the best individual is more likely to be selected for mating.

1. *Generate random population of n chromosomes*
2. *Evaluate the fitness $f(x)$ of each chromosome x in the population*
3. *Create a new population by repeating following steps until the new population is complete.*
 - Select two parent chromosomes from a population according to their fitness*
 - Crossover recombines the parents to form new offspring*
 - Mutate new offspring at selected position*
 - Place new offspring in new population*
4. *If the end condition is satisfied, stop, and return the best solution in current population.*
5. *Go to step 2*

Algorithm 3: The outline of basic genetic algorithm

Since the entire population is replaced in each generation, the only memory the algorithm has is from the performance of the genetic operators. If the genetic operators convey good fragments or segments from parents to offspring, the population will improve. Otherwise, the population will not improve and the evolutionary algorithm will perform no better than a random search.

8.2 Steady-State Evolutionary Algorithm

In each generation the algorithm replaces only a fraction of the current population. A temporary population is created each generation, with a fraction of the size of the current population. Then this temporary population is added to the current population. After structure evaluation and ranking, worst chromosomes are removed to bring the current population to its original size.

The steady-state evolutionary algorithm uses overlapping populations. In each generation, a portion of the population is replaced by the newly generated candidates. This process is illustrated in Algorithm 3. At one extreme, only one or two structures may be replaced each generation (close to 100% overlap). At the other extreme, the steady-state algorithm becomes a simple evolutionary algorithm when the entire population is replaced (0% overlap). The best-so-far structure is always preserved.

In every generation

- 1. Establish a temporary population (size N_{temp}) by doing genetic operation to selected individuals in current population*
- 2. Add the temporary population into current population*
- 3. Crowd out worst N_{temp} individuals in this enlarged population*
- 4. The rest of population survives to new generation.*

Algorithm 4: the steady-state evolutionary algorithm I. This algorithm constructs the offspring population from a redundant population; Selection is biased to good structures. This algorithm may have difficulty to find full solution set when multiple solutions exist. A light selection pressure is suggested to prevent from premature convergence.

Since the algorithm only replaces a portion of the population of each generation, the best candidates are more likely to be selected and the population quickly converges to a local optimum. Once again, the crossover and mutation operators are key to the algorithm performance; a crossover operator that generates children unlike their parents and/or a high mutation rate can delay the convergence. Algorithm 4 makes a modification to the steady-state evolutionary algorithm. A pre-elimination mechanism is introduced to empty rooms for fresh structures with a hope to reshuffle population distribution.

In every generation

- 1. Establish a temporary population (size N_{temp}) by doing genetic operation to selected chromosomes in current population*
- 2. Discard worst N_{kill} chromosomes from current population*
- 3. Merge the temporary population into current population*
- 4. Crowd out worst ($N_{temp} - N_{kill}$) chromosomes in this enlarged population*
- 5. The rest of population survives to next generation.*

Algorithm 4: the enhanced steady-state evolutionary algorithm II. This algorithm involves redundant population and structure pre-elimination; only a portion of the population is replaced each generation. The size of temporary population and the amount of pre-elimination (percentage of the parent

population that is replaced) are subjected to change triggered by the algorithm itself.

8.3 Diversity-Driven Evolutionary Algorithm

The diversity-driven evolutionary algorithm is similar to the steady-state evolutionary algorithm. However, rather than replacing the worst candidate, a new structure replaces the candidate most similar to it; and the replacement is executed only if the new structure has a fitness score better than that of the one to which it is most similar. The diversity-driven evolutionary algorithm classifies the whole population into small regions, called niches. Search is thus allowed focusing simultaneously in more than one region of search space. This requires the use of one kind of similarity measure defined above. The similarity measure indicates how different two structures are, either in terms of their fitness scores or of their structural features. If the similarity function is properly defined, the diversity-driven algorithm maintains diversity extremely well.

Self-adaptation has been frequently employed in evolutionary methods. Three distinct adaptive levels are defined: population, individual and component levels. The implemented evolutionary algorithms framework provides self-adaptation at all these levels. Instead of using component level adaptation, we emphasize the scheme based on the success rate of the whole population. The population level adaptation has richer and more accurate information about the search status because it is the whole population that is evolving, not just separate individuals.



Figure 9: An evolution scheme using self-adaptive population control in a multi-population environment. The population size takes an unfixed value which is subjected to increase when the average fitness score in this population decrease.

Chapter 9: Results and Discussion

Table 3 characterizes of the test suites. All of the tests were run using a single implementation of the evolutionary algorithm; although minor changes were made to accommodate different sets of fitness functions, no changes to the representation or evolutionary algorithm is required.

First we take the Monochaetin problem⁹⁸ ($C_{18}H_{20}O_5$) as the proof-of-concept testcase, which has served as an example in various CASE papers^{8,11,41} and/or which the correct number and identity of all solutions is known from deterministic CASE runs. The spectra information comes from ^{13}C NMR (BB, DEPT 90 and DEPT 135), HH COSY and HMBC (data summarized in Table XXX and Table XXX). HH COSY judge has 8 constraint entries and HMBC 24. Together with the contribution of HOSE judge (18 entries), the maximum achievable fitness score is 5000. The three judges have identical and constant weighting coefficients in all EA runs. No other judges are employed to make the problem simple.

To reflect the process of structure evolution, a small population size (16 individuals) is selected, and only two genetic operators used, crossover and mutation. The probabilities of crossover and mutation are controlled by equation 1 and 2

$$P_{crossover} = 0.5 \times \sqrt{1 - S_{best} / S_{goal}} \quad (1)$$

$$P_{mutation} = 1 - P_{crossover} \quad (2)$$

where S_{best} stands for the fitness score of the best-so-far structure and S_{goal} is the fitness score of the target structure(s) for the unknown. For crossover operation, the

value of *Partition scale* ranges from 4 to 8 (one-third of the skeleton atom number 23), and is determined according to equation 3.

$$V_{partition} = (V_{max} - V_{min}) \times (1 + Dev / Dev_{empirical}) \quad (3)$$

Here, V_{max} and V_{min} are maximum and minimum allowed values of the partition scale; Dev is the standard deviation of the fitness scores of the last population; and $Dev_{empirical}$ is the empirical maximum standard deviation of the fitness scores of a population (in this case 200). $Dev_{empirical}$ is obtained by calculating the average of the standard deviations of a set of initial populations.

In each generation, a temporary population of 16 new structures is created through reproduction operations. Among current and the temporary populations (32 structures in all), 16 are selected to form the offspring population with the top two structures always been preserved. Selection is through either the `Tournament Selector` or the `Ranking Order Selector`.

The result is illustrated in Figure 10. One hit is found after about 350 generations within 3 minutes run. There are about 5600 points sampled in the constitution space, which is a tiny amount compared to the huge number of constitutional isomers for $C_{18}H_{20}O_5$. Notably, this result is obtained with a less appropriate population scale, and no other constraints introduced or other parameters optimized.

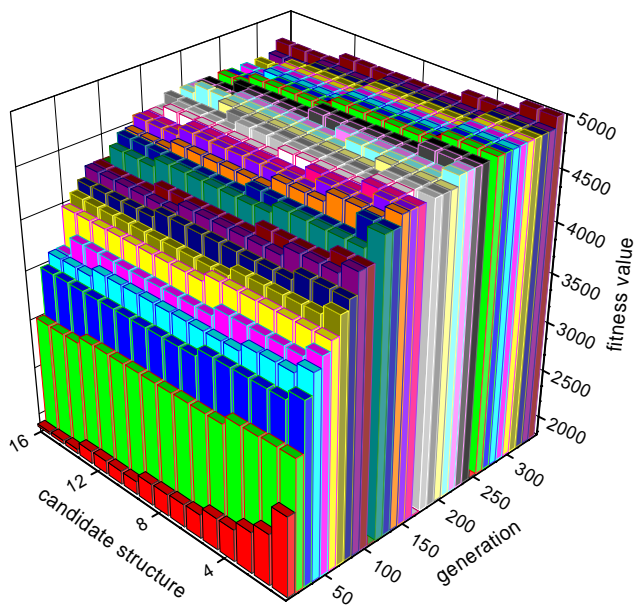


Figure 10: Evolution of Monochaetin ($C_{18}H_{20}O_5$). A population of 16 structures evolves over about 350 generations. In the end to the evolutionary process, about 10 percent of the candidate structures have reached the maximum achievable score of 5000 points.

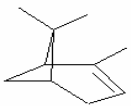
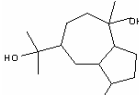
Table 3: Empirical parameter settings for sample problems

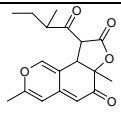
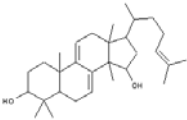
Samples	$C_{10}H_{16}$	$C_{15}H_{28}O_2$	$C_{18}H_{20}O_5$	$C_{30}H_{48}O_2$
population size	8	36/48	36/48	120/160
replacement rate	100	100	100	100
killing rate	0	0	0.1	0.2
mutation rate	0.5	0.25	adaptive	0.30

niche search rate	-	0.25	1*	0.30
crossover rate	0.5	0.50	adaptive	0.40
mutation strength	1	1	1	1
match mode	random	random	difference	random
partition mode	random	depth-first (df)	breadth-first (bf)	bf / df = 7 / 3
partition scale	-	4-6	4-8	4-11

*Niche search works on the top two structures and happens when the population scale is to change (from 36 to 48) after every 5 generations of no improvement.

Table 4: Performance overview of the EA implementation with parameter settings given in Table 3. Calculations were performed on a computer running Windows XP equipped with a Pentium 3 500 MHz CPU and 256MB RAM. Results are taken as average of 20 runs. The fitness function involved HHCOSY, HMBC and HOSE code judge. Calculation times are given as TRS50, defined as the mean time in which the 50% most successful processes reach the optimum.

Structure	MF	Points visited vs. general	Generations	Solutions	Calculation time
	C ₁₀ H ₁₆	48 / 4,305	6	1	< 1 sec.
	C ₁₅ H ₂₈ O ₂	2,088 / ?	60	1	40 sec.

	$C_{18}H_{20}O_5$	2,600 / ?	70	1	120 sec.
	$C_{30}H_{48}O_2$	6,400 / ?	48	6*	120 sec.

***In all, 6 structures are found after 20 runs, which are consistent with those found by a deterministic approach ⁸.**

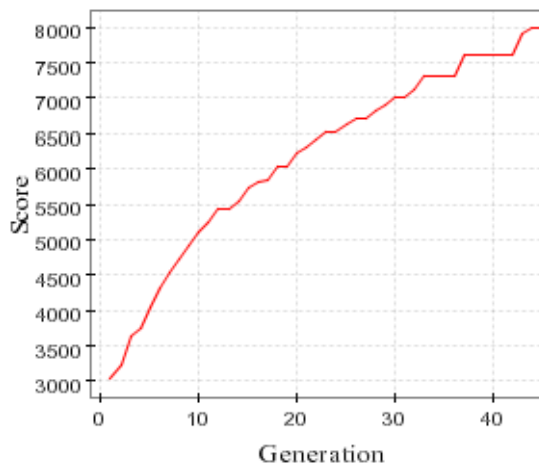
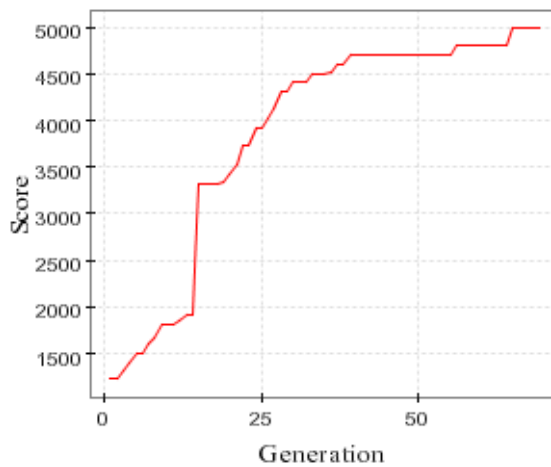
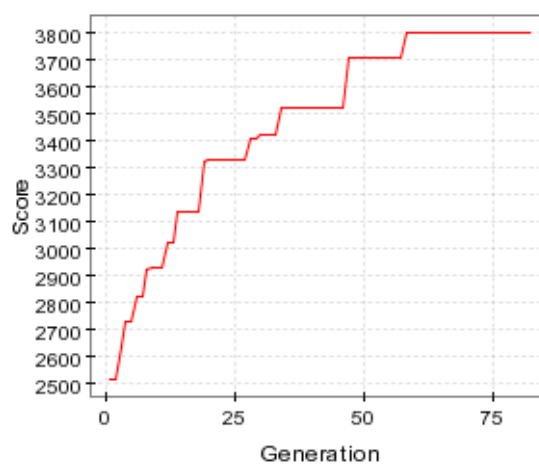
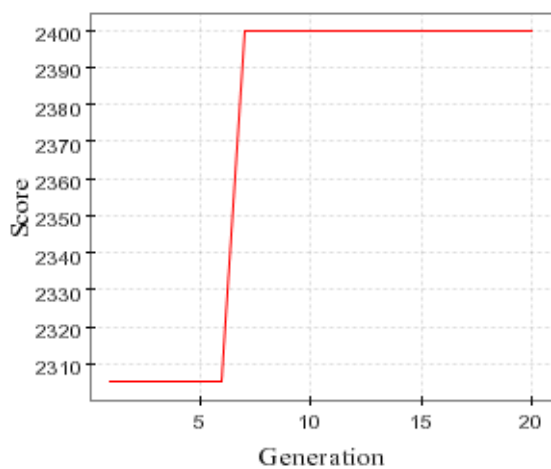


Figure 8: The evolution curves corresponding to samples 1, 2, 3 and 4.

Several structure elucidation examples of increasing constitution scale are shown in Table 3, Table 4 and Figure 8 to evaluate the algorithm's performance.

Table 3 gives typical parameter settings for 4 compounds of different size. The task of Monochaetin structure elucidation is approached again using a mechanism of population scale control. An EA run adjusts its population following procedures below.

- 1. Start the EA with population size as 36 and iterate step 2-6.*
- 2. If after 5 generations no better solutions appear, delete one-third of the worst candidates and expand the population size to 48. New individuals are created by doing a niche search around the top 5 structures in the population.*
- 3. EA now runs with population size as 48.*
- 4. If after 5 generations no new solutions found, shrink the population size back to 36. The removed individuals, half from good ones, half from bad ones (the best solutions are always preserved).*
- 5. EA now runs with population size as 36.*
- 6. For every 20 generations, an isomorphism check is done to reshuffle the population.*

Averagely, with this scheme, 50 percent of EA runs can converge to the optimum after sampling 3000 points in constitution space, other runs take longer time but almost all are able to find the solution.

In many cases, instead of one optimum structure, a number of structures exist satisfying all given inputs. In this case, finding all solutions cannot be achieved with a relatively small population scale. Therefore, a set of EA runs need to be carried out before any conclusions are drawn. For example, a single EA run configured as in Table 3 cannot find all 6 structures for compound 3 (Polycarpol), but running the algorithm 20 times will succeed. This is a well known property of stochastic optimization schemes, which is why frequently a statistic backing of the result is

created by collecting the outcome of multiple optimization runs performed either in parallel or sequential manner.

Compared to an earlier deterministic structure elucidation module created in our group (Steinbeck 1996), both stochastic algorithms, Simulated Annealing (Steinbeck 2001) as well as the Evolutionary Algorithm described in this paper, scale much smoother in their computation times versus number of heavy atoms in the problem set, as illustrated in Figure 11.

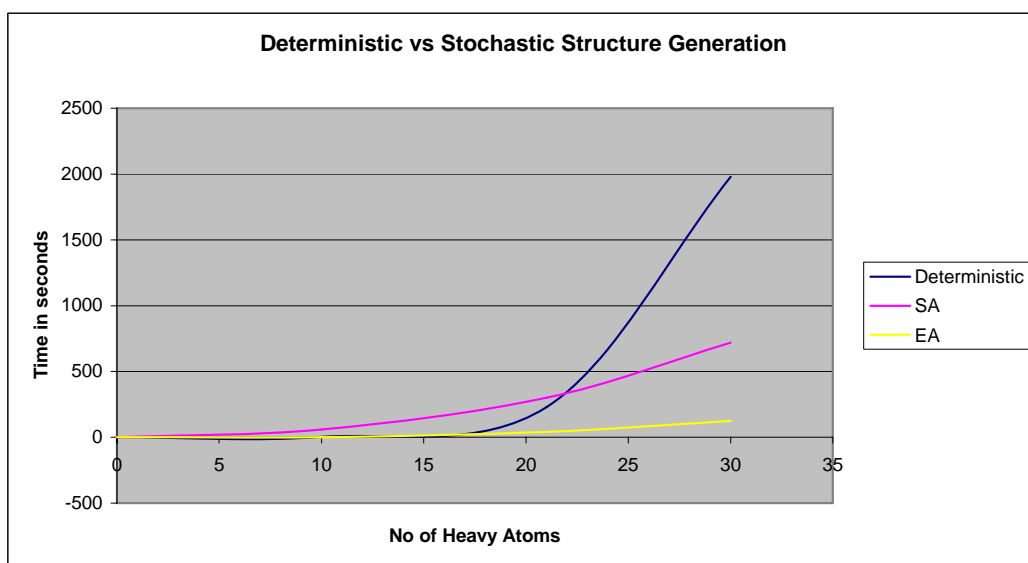


Figure 11: Qualitative comparison of runtimes for our deterministic and stochastic implementations^{5,8}. Calculation times were measured for problem sets listed in Table 4. Both Simulated Annealing (SA) and Evolutionary Algorithm (EA) show a significantly smoother increase of computation time versus number of heavy atom in the problem set.

Besides, we are currently testing a newly designed multi-population based evolution scheme to maintain EA's coverage of the problem space, especially for unknowns with huge search space. The new scheme inherits a concept of a two level evolution: a set of peripheral EA threads run independently with different configurations. For an interval of 60 seconds, each EA thread contributes a number of (cloned) good structures to generate the population of the core EA thread. The peripheral EA runs find local optima in different regions, and the core EA thread aims to refine the good structures by niche searches. This scheme is currently being tested with several larger compounds.

Chapter 10: Conclusions

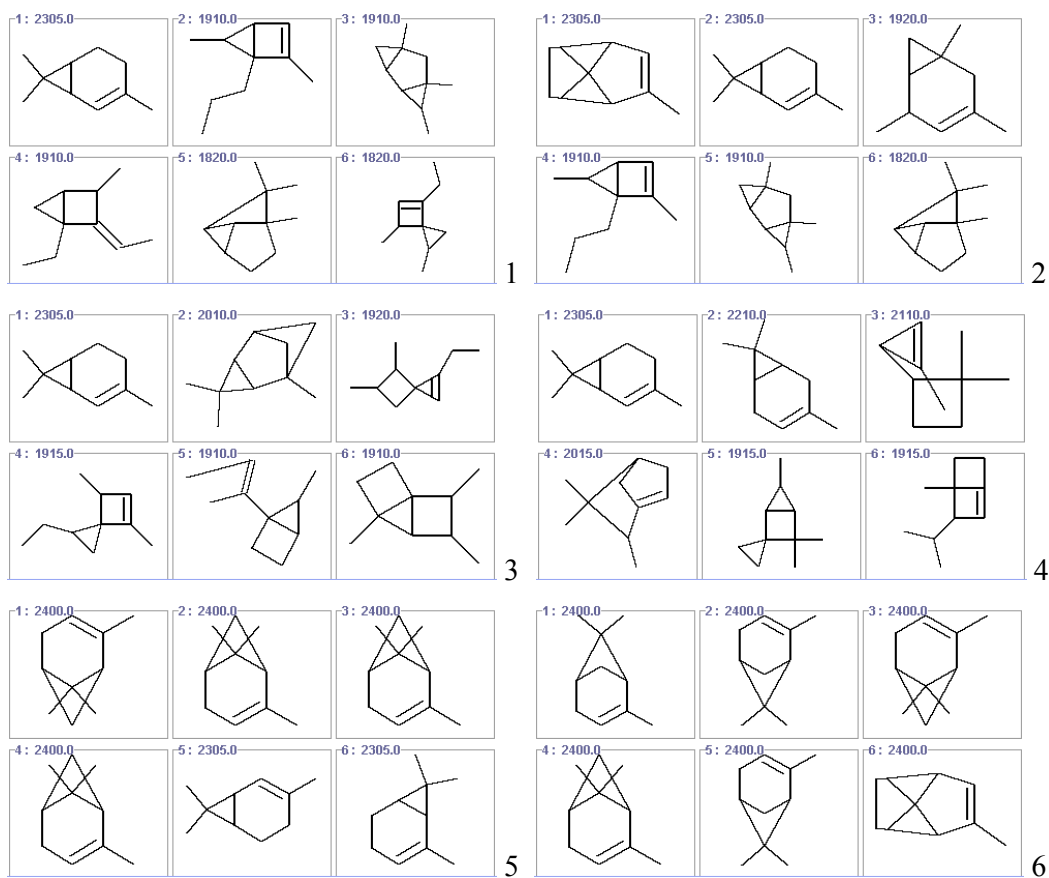
An evolutionary algorithm framework using a graph-based data structure to explore the molecular constitution space is developed. The algorithm framework provides a suite of robust graph operators to propel the evolution of molecular structures towards a set of desired properties. The graph data structure facilitates efficient genetic manipulation and exempts from the transformation between genotype and phenotype of the candidate solution. A strong control of their parameters enables the genetic operators to adjust their behavior and achieve higher search efficiency.

The EA implementation proves to be a promising alternative to deterministic approaches to the problem of computer assisted structure elucidation (CASE). While not relying on any external database, The EA-guided CASE program SENECA is able to find correct solution within calculation time comparable to that of other CASE expert systems. The implementation presented here significantly expands the size limit of constitutional optimization problems treatable with evolutionary algorithms by introducing efficient graph-based data structure and genetic operators.

The implemented search engine now is part of the CASE program SENECA, and its performance demonstrated by solving real-world structure elucidation problems.

Appendix A

Figure 1: Evolution of α -Pinene ($C_{10}H_{16}$): the top six structures in six successive generations are shown (constraints: molecular formula, DEPT 135/90, HMBC and carbon chemical shifts)



Appendix B

Table 1: Information provided by the most commonly used 2D-NMR experiments. The types of correlations listed are those giving rise to the majority of cross signals in the respective experiments. All experiments may also contain cross-signals from unsuppressed other correlation types.

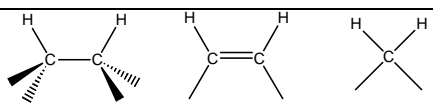

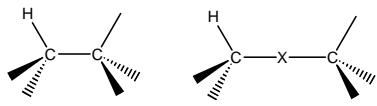
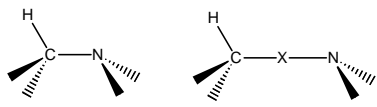
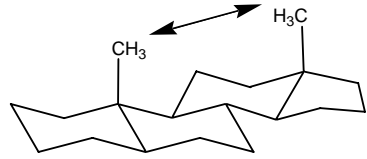
Experiment	Correlation Type	Sketch
HH-COSY	${}^3J_{\text{HH}}, {}^2J_{\text{HH}}$	
HetCor (CH-COSY, HSQC, HMQC)	${}^1J_{\text{CH}}$	
Long-range HetCor (COLOC, HMBC)	${}^2J_{\text{CH}}, {}^3J_{\text{CH}}$	
HN-HMBC	${}^2J_{\text{NH}}, {}^3J_{\text{NH}}$	
NOESY, ROESY	Dipolar ${}^1\text{H}{}^1\text{H}$ couplings through space: $1.5 \text{ \AA} <$ $D(\text{H}_a, \text{H}_b) < 5 \text{ \AA}$	

Table 2: A list of 2D NMR signals from HMQC and HMBC 2D NMR Experiments (A) and a list of heavy atom distance constraints derived by the CASE program by signal matching between HMBC and HMQC data (B). An "O" in table B indicates that the two heavy atoms are to be separated by one or two bonds in the resulting graph. Only the lower half of the symmetric relationship table B has been populated for the sake of clarity.

A

No.	¹³ C CPD	¹ J _{CH} (HMQC)	HMBC
1	148.24	-	4.0; 2.42; 2.28; 1.20
2	118.25	5.49	4.0; 2.28; 2.17
3	66.32	4.0	5.49
4	43.87	2.17	5.49; 4.0; 2.42; 1.32; 1.20; 0.86
5	41.40	2.14	5.49; 1.32; 0.86
6	38.32	-	2.42; 1.32; 1.20; 0.86

B

	1	2	3	4	5	6	7	8	9	
1	X									
2		X								
3	O	O	X							
4		O	O	X						
5		O			X					
6						X				
7	O			O		O	X			
8	O	O						X		
9				O	O				X	
1				O	O	O			O	X

1-D ^{13}C NMR Data		$^1\text{J}_{\text{CH}}$ Correlations		$^{2/3}\text{J}_{\text{CH}}$ Correlations	
Shift (ppm)	Mult.				
205.94	S	C143.30	H6.79	C18.92	H3.76
191.77	S	C107.04	H6.02	C19.49	H6.02
169.10	S	C105.73	H5.29	C26.27	H1.11
158.52	S	C43.66	H3.76	C43.66	H1.32
145.52	S	C52.13	H4.05	C43.66	H4.05
143.30	D	C46.70	H3.19	C46.70	H1.11
116.22	S	C26.27	H1.81	C52.13	H3.76
107.04	D	C26.27	H1.48	C82.55	H1.32
105.73	D	C11.45	H0.97	C82.55	H3.76
82.55	S	C19.49	H2.13	C82.55	H5.29
52.13	D	C18.92	H1.32	C105.73	H6.02
46.70	D	C14.39	H1.11	C107.04	H2.13
43.66	D			C107.04	H5.29
26.27	T			C116.22	H3.76
19.49	Q	$^3\text{J}_{\text{HH}}$ Correlations		C116.22	H5.29
18.92	Q	H4.05	H3.76	C116.22	H6.02
14.39	Q	H3.19	H1.11	C116.22	H6.79
11.45	Q	H3.19	H1.81	C143.30	H3.76
				C145.52	H6.02
				C145.52	H6.79

Table 3: A list of NMR signals from 1D ^{13}C NMR as well as 2D HMQC, HH COSY and HMBC experiments used for the structure elucidation of the fungal *Monochaetia*.

Atom:	DEP	δ_c [ppm]	CH COSY	CH COLOC	HH COSY
	T				
C-1	C	146.0		6.0, 2.0/2.26, 1.01	
C-2	C	142.0		5.3, 1.04	
C-3	C	131.0		1.66, 1.6	
C-4	CH	125.5	5.2	1.66, 1.6	
C-5	CH	122.0	6.0	2.05/ 2.14	2.05/ 2.14
C-6	CH	116.3	5.3	2.0/ 2.26	2.0/ 2.26
C-7	CH	79.0	3.15		1.68
C-8	CH	74.8	4.3	1.04	1.83/ 1.93
C-9	C	52.5		1.04, 0.64	
C-10	CH	49.7	1.14	1.0, 1.01, 0.91	
C-11	CH	49.3	1.65	0.87, 0.64	1.83/ 1.93
C-12	C	44.4		5.3, 1.65, 1.04, 0.64	
C-13	CH ₂	39.8	1.83/ 1.93		4.3, 1.65
C-14	C	39.0		1.14, 1, 1.68, .91	
C-15	CH ₂	38.8	2.0/ 2.26	0.64	5.3
C-16	C	37.7		1.14, 1.68, 1.01	
C-17	CH ₂	36.8	1.07/1.45	0.87	
C-18	CH	36.4	1.32	0.87	0.87
C-19	CH ₂	36.3	1.38/1.92	1.01	1.68
C-20	CH ₃	28.3	1.0	0.91	
C-21	CH ₂	27.9	1.68		3.15, 1.38/ 1.92
C-22	CH ₃	25.9	1.66	1.6	
C-23	CH ₂	25.3	2.0		
C-24	CH ₂	23.3	2.05/2.14		6
C-25	CH ₃	23.0	1.01	1.14	
C-26	CH ₃	18.7	0.87		1.32
C-27	CH ₃	17.8	1.6	1.66	

Atom:	DEP	δ_c [ppm]	CH COSY	CH COLOC	HH COSY
	T				
C-28	CH ₃	17.5	1.04		
C-29	CH ₃	16.1	0.64	1.65, 2.0/ 2.26	
C-30	CH ₃	16.0	0.91	1.14, 1.0	

Table 4: NMR data used for the structure elucidation of Polycarpol (C₃₀H₄₈O₂).

References

- (1) Munk, M. E. Computer-Based Structure Determination: Then and Now. *Journal of Chemical Information and Computer Sciences* **1998**, *38*, 997-1009.
- (2) Steinbeck, C. Computer-Assisted Structure Elucidation. *Handbook on Chemoinformatics.*; Wiley-VCH: Weinheim, 2003; pp 1378-1406.
- (3) Steinbeck, C. Correlations between Chemical Structures and NMR Data. *Handbook on Chemoinformatics.*; Wiley-VCH: Weinheim, 2003; pp 1368-1377.
- (4) Steinbeck, C. The Automation of Natural Product Structure Elucidation. *Current Opinion in Drug Discovery and Development* **2001**, *4*, 338-342.
- (5) Steinbeck, C. SENECA: A platform-independent, distributed, and parallel system for computer-assisted structure elucidation in organic chemistry. *Journal of Chemical Information & Computer Sciences* **2001**, *41*, 1500-1507.
- (6) Kirkpatrick, S.; Gerlatt, C. D. J.; Vecchi, M. P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671-680.
- (7) Faulon, J. L. Stochastic generator of chemical structure .2. Using simulated annealing to search the space of constitutional isomers. *Journal of Chemical Information and Computer Sciences* **1996**, *36*, 731-740.
- (8) Steinbeck, C. Lucy - A Program For Structure Elucidation From NMR Correlation Experiments. *Angewandte Chemie. International Ed. in English* **1996**, *35*, 1984-1986.
- (9) Bremser, W. HOSE - A Novel Substructure Code. *Analytica Chimica Acta* **1978**, *103*, 355-365.
- (10) Meiler, J.; Will, M. Automated structure elucidation of organic molecules from C-13 NMR spectra using genetic algorithms and neural networks. *Journal of Chemical Information and Computer Sciences* **2001**, *41*, 1535-1546.

- (11) Steinbeck, C. SENECA: A Platform-Independent, Distributed and Parallel System for Computer-Assisted Structure Elucidation in Organic Chemistry. *J. Chem. Inf. Comput. Sci.*, accepted for publication on June 22, 2001 **2001**.
- (12) Elyashberg, M. E.; Blinov, K. A.; Williams, A. J.; Martirosian, E. R.; Molodtsov, S. G. Application of a new expert system for the structure elucidation of natural products from their 1D and 2D NMR data. *Journal of Natural Products* **2002**, *65*, 693-703.
- (13) Williams, A. Recent advances in NMR prediction and automated structure elucidation software. *Current Opinion in Drug Discovery and Development* **2000**, *3*, 298-305.
- (14) Blinov, K. A.; Elyashberg, M. E.; Molodtsov, S. G.; Williams, A. J.; Martirosian, E. R. An expert system for automated structure elucidation utilizing H-1-H-1, C-13-H-1 and N-15-H-1 2D NMR correlations. *Fresenius Journal of Analytical Chemistry* **2001**, *369*, 709-714.
- (15) Hao, J. F.; Xu, L.; Hu, C. Y. Expert system for elucidation of structures of organic compounds (ESESOC) - Algorithm on stereoisomer generation. *Science in China Series B, Chemistry, Life Sciences & Earth Sciences* **2000**, *43*, 503-515.
- (16) Stokov, II; Lebedev, K. S. Computer aided method for chemical structure elucidation using spectral databases and C-13 NMR correlation tables. *Journal of Chemical Information & Computer Sciences* **1999**, *39*, 659-665.
- (17) Steinbeck, C. Recent advancements in the development of SENECA, a computer program for Computer Assisted Structure Elucidation based on a stochastic algorithm. *Abstracts of Papers of the American Chemical Society* **1999**, *218*, 34-CINF.
- (18) Jaspars, M. Computer assisted structure elucidation of natural products using two-dimensional NMR spectroscopy. *Natural Product Reports* **1999**, *16*, 241-247.
- (19) Lindel, T.; Junker, J.; Kock, M. 2D-NMR-guided constitutional analysis of organic compounds employing the computer program COCON. *European Journal of Organic Chemistry* **1999**, *3*, 573-577.

- (20) Elyashberg, T. E.; Karasev, Y. Z.; Martirosian, E. R. Spectroscopic determination of elemental composition of organic with the aid of the X-PERT system. *Analytica Chimica Acta* **1999**, *388*, 353-363.
- (21) Williams, A. J.; Shilay, V.; Mityushev, D. Developments in nmr chemical-shift prediction and utilization of user databases to improve possibilities for structure elucidation. *Abstracts of Papers of the American Chemical Society* **1998**, *216*, 20-COMP.
- (22) Peng, C.; Bodenhausen, G.; Qiu, S. X.; Fong, H. H. S.; Farnsworth, N. R. et al. Computer-assisted structure elucidation: Application of CISOC- SES to the resonance assignment and structure generation of betulinic acid. *Magnetic Resonance in Chemistry* **1998**, *36*, 267-278.
- (23) Nuzillard, J. M. Computer-assisted structure determination of organic-molecules. *Journal de Chimie Physique et de Physico-Chimie Biologique* **1998**, *95*, 169-177.
- (24) Molchanova, M. S.; Zefirov, N. S. Irredundant generation of isomeric molecular-structures with some known fragments. *Journal of Chemical Information and Computer Sciences* **1998**, *38*, 8-22.
- (25) Elyashberg, M. E.; Karasev, Y. Z.; Martirosian, E. R.; Thiele, H.; Somberg, H. Expert systems as a tool for the molecular structure elucidation by spectral methods - strategies of solution to the problems. *Analytica Chimica Acta* **1997**, *348*, 443-463.
- (26) Wieland, T.; Kerber, A.; Laue, R. Principles of the generation of constitutional and configurational isomers. *Journal of Chemical Information & Computer Sciences* **1996**, *36*, 413-419.
- (27) Will, M.; Fachinger, W.; Richert, J. R. Fully automated structure elucidation - A spectroscopist's dream comes true. *Journal of Chemical Information and Computer Sciences* **1996**, *36*, 221-227.
- (28) Sasaki, K. F. a. S.-i. Recent Advances in the Automated Structure Elucidation System, CHEMICS. Utilization of Two-Dimensional NMR Spectral Information and Development of Peripheral Functions for Examination of Candidates. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 190 -204.

- (29) Neudert, R.; Penk, M. Enhanced structure elucidation. *Journal of Chemical Information & Computer Sciences* **1996**, *36*, 244-248.
- (30) Funatsu, K.; Sasaki, S.-i. Recent Advances in the Automated Structure Elucidation System, CHEMICS. Utilization of Two-Dimensional NMR Spectral Information and Development of Peripheral Functions for Examination of Candidates. *Journal of Chemical Information and Computer Sciences* **1996**, *36*, 190-204.
- (31) Peng, C.; Yuan, S. G.; Zheng, C. Z.; Shi, Z. S.; Wu, H. M. Practical Computer-Assisted Structure Elucidation for Complex Natural-Products - Efficient Use of Ambiguous 2d Nmr Correlation Information. *Journal of Chemical Information and Computer Sciences* **1995**, *35*, 539-546.
- (32) Jones, G.; Willett, P.; Glen, R. C. A genetic algorithm for flexible molecular overlay and pharmacophore elucidation. *Journal of Computer-Aided Molecular Design* **1995**, *9*, 532-549.
- (33) Peng, C.; Yuan, S. G.; Zheng, C. Z.; Hui, Y. Z.; Wu, H. M. et al. Application of Expert-System Cisoc-Ses to the Structure Elucidation of Complex Natural-Products. *Journal of Chemical Information and Computer Sciences* **1994**, *34*, 814-819.
- (34) Faulon, J. L. Stochastic Generator of Chemical-Structure .1. Application to the Structure Elucidation of Large Molecules. *Journal of Chemical Information and Computer Sciences* **1994**, *34*, 1204-1218.
- (35) Funatsu, K.; Nishizaki, M.; Sasaki, S. Introduction of Noe Data to an Automated Structure Elucidation System, Chemics - 3-Dimensional Structure Elucidation Using the Distance Geometry Method. *Journal of Chemical Information and Computer Sciences* **1994**, *34*, 745-751.
- (36) Warr, W. A. Computer-Assisted Structure Elucidation .1. Library Search and Spectral Data Collections. *Analytical Chemistry* **1993**, *65*, A1045-A1050.
- (37) Yuan, S. G.; Peng, C.; Zheng, C. Z. Application of a C-13 Nmr Topological Model to the Structure Elucidation of Organic-Compounds. *Science in China Series a-Mathematics Physics Astronomy* **1992**, *35*, 1136-1143.

- (38) Funatsu, K.; Susuta, Y.; Sasaki, S. Application of the Automated Structure Elucidation System (Chemics) to the Chemistry of Natural-Products. *Pure and Applied Chemistry* **1989**, *61*, 609-612.
- (39) Robien, W. Computer-Assisted Structure Elucidation of Organic-Compounds .3. Automatic Fragment Generation from C-13-Nmr Spectra. *Mikrochimica Acta* **1986**, *2*, 271-279.
- (40) Funatsu, K.; Delcarpio, C. A.; Sasaki, S. Automated Structure Elucidation System - Chemics. *Fresenius Zeitschrift Fur Analytische Chemie* **1986**, *324*, 750-759.
- (41) Munk, M. Computer-Assisted Structure Elucidation. *Fresenius Zeitschrift Fur Analytische Chemie* **1982**, *311*, 317-317.
- (42) Schwenzer, G. M. Computer-Assisted Structure Elucidation Using Automatically Acquired C-13 Nmr Rules. *Abstracts of Papers of the American Chemical Society* **1977**, *173*, 24-24.
- (43) Smith, D. H.; Carhart, R. E. Computer-Assisted Structure Elucidation. *Abstracts of Papers of the American Chemical Society* **1975**, 3-3.
- (44) Djerassi, C.; Smith, D. H.; Crandell, C. W.; Gray, N. A. B.; Nourse, J. G. et al. Applications of Artificial-Intelligence for Chemical Inference .42. The Dendral Project - Computational Aids to Natural- Products Structure Elucidation. *Pure and Applied Chemistry* **1982**, *54*, 2425-2442.
- (45) Smith, D. H.; Gray, N. A. B.; Nourse, J. G.; Crandell, C. W. Applications of Artificial-Intelligence for Chemical Inference .38. The Dendral Project - Recent Advances in Computer-Assisted Structure Elucidation. *Analytica Chimica Acta-Computer Techniques and Optimization* **1981**, *5*, 471-497.
- (46) Abe, H.; Yamasaki, T.; Sasaki, S. Chemics - Computer-Assisted Structure Elucidation of Organic- Compounds by Spectrometric Data and Chemical Evidences. *Abstracts of Papers of the American Chemical Society* **1979**, 34-34.
- (47) Munk, M. E. Computer-based structure determination - then and now. *Journal of Chemical Information & Computer Sciences* **1998**, *38*, 997-1009.

- (48) Griffiths, L.; Bright, J. D. Towards the automatic analysis of H-1 NMR spectra: Part 3. Confirmation of postulated chemical structure. *Magnetic Resonance in Chemistry* **2002**, *40*, 623-634.
- (49) Steinbeck, C.; Kuhn, S.; Krause, S. NMRShiftDB - An Open-Submission, Open-Retrieval Database for Organic Structures and their NMR Data (<http://www.nmrshiftdb.org>, visited on June 2003).
- (50) Elyashberg, M. E.; Karasev, Y. Z.; Martirosian, E. R.; Thiele, H.; Somberg, H. Expert systems as a tool for the molecular structure elucidation by spectral methods. Strategies of solution to the problems. *Analytica Chimica Acta* **1997**, *348*, 443-463.
- (51) Wieland, T. Construction algorithms in molecular graphs and their applications. *Match-Communications in Mathematical and in Computer Chemistry* **1997**, *7*.
- (52) Aires-de-Sousa, J.; Hemmer, M. C.; Gasteiger, J. Prediction of H-1 NMR chemical shifts using neural networks. *Analytical Chemistry* **2002**, *74*, 80-90.
- (53) Doucet, J. P.; Panaye, A.; Feuilleaubois, E.; Ladd, P. Neural Networks and C-13 Nmr Shift Prediction. *Journal of Chemical Information and Computer Sciences* **1993**, *33*, 320-324.
- (54) Magri, F. M. M.; Militao, J. S. L.; Ferreira, M. J. P.; Brant, A. J. C.; Emerenciano, V. P. Applications of artificial intelligence in organic chemistry part XXIV - A new program to C-13 NMR spectrum prediction based on tridimensional models. *Spectroscopy-an International Journal* **2001**, *15*, 99-117.
- (55) Maier, W. New Approaches to Computer-Aided NMR Interpretation and Structure Prediction. *Computer-Enhanced Analytical Spectroscopy*; Plenum Press: New York, London, 1993; pp 37-55.
- (56) Meiler, J.; Meusinger, R.; Will, M. Fast determination of C-13 NMR chemical shifts using artificial neural networks. *Journal of Chemical Information & Computer Sciences* **2000**, *40*, 1169-1176.
- (57) Smith, S. K.; Cobleigh, J.; Svetnik, V. Evaluation of a H-1-C-13 NMR spectral library. *Journal of Chemical Information & Computer Sciences* **2001**, *41*, 1463-1469.

- (58) Trepalin, S. V.; Yarkov, A. V.; Dolmatova, L. M.; Zefirov, N. S.; Finch, S. A. E. Windat - an nmr database compilation tool, user interface, and libraries for personal computers. *Journal of Chemical Information & Computer Sciences* **1995**, *35*, 405-411.
- (59) Xu, J. C-13 NMR spectral prediction by means of generalized atom center fragment method. *Molecules* **1997**, *2*, 114-128.
- (60) Meiler, J.; Sanli, E.; Junker, J.; Meusinger, R.; Lindel, T. et al. Validation of structural proposals by substructure analysis and C-13 NMR chemical shift prediction. *Journal of Chemical Information and Computer Sciences* **2002**, *42*, 241-248.
- (61) Schutz, V.; Purtuc, V.; Felsing, S.; Robien, W. Csearch-stereo - a new generation of nmr database systems allowing three-dimensional spectrum prediction. *Fresenius Journal of Analytical Chemistry* **1997**, *359*, 33-41.
- (62) Meiler, J.; Meringer, M. Ranking MOLGEN structure proposals by C-13 NMR chemical shift prediction with ANALYZE. *Match-Communications in Mathematical and in Computer Chemistry* **2002**, 85-108.
- (63) Lebedev, K. S.; Cabrolbass, D. New computer-aided methods for revealing structural features of unknown compounds using low-resolution mass-spectra. *Journal of Chemical Information and Computer Sciences* **1998**, *38*, 410-419.
- (64) Filimonov, D.; Poroikov, V.; Borodina, Y.; Glorizova, T. Chemical similarity assessment through multilevel neighborhoods of atoms: definition and comparison with the other descriptors. *Journal of Chemical Information and Computer Sciences* **1999**, *39*, 666-670.
- (65) Bohacek, R. S.; McMartin, C. Modern computational chemistry and drug discovery - structure generating programs. *Current Opinion in Chemical Biology* **1997**, *1*, 157-161.
- (66) Faulon, J. L. Isomorphism, automorphism partitioning, and canonical labeling can be solved in polynomial-time for molecular graphs. *Journal of Chemical Information and Computer Sciences* **1998**, *38*, 432-444.
- (67) Steinbeck, C.; Han, Y. Q.; Kuhn, S.; Horlacher, O.; Luttmann, E. et al. The Chemistry Development Kit (CDK): An open-source Java library for chemo- and

- bioinformatics. *Journal of Chemical Information and Computer Sciences* **2003**, *43*, 493-500.
- (68) Bangov, I. P. Fuzzy Logic in Computer-Aided Structure Elucidation. *Fuzzy Logic in Chemistry*; Academic Press: San Diego, 1997; pp 283-320.
- (69) Stadler, P. F. Landscapes and their correlation functions. *Journal of Mathematical Chemistry* **1996**, *20*, 1-45.
- (70) Smith, T.; Husbands, P.; Layzell, P.; O'Shea, M. Fitness landscapes and evolvability. *Evolutionary Computation* **2002**, *10*, 1-34.
- (71) Schuster, P. Landscapes and molecular evolution. *Physica D* **1997**, *107*, 351-365.
- (72) Reidys, C. M.; Stadler, P. F. Combinatorial landscapes. *Siam Review* **2002**, *44*, 3-54.
- (73) Bornholdt, S. Genetic algorithm dynamics on a rugged landscape. *Physical Review E* **1998**, *57*, 3853-3860.
- (74) Albuquerque, P.; Chopard, B.; Mazza, C.; Tomassini, M. On the impact of the representation on fitness landscapes. *Genetic Programming, Proceedings*, 2000; pp 1-15.
- (75) Bak, P.; Flyvbjerg, H.; Lautrup, B. Coevolution in a Rugged Fitness Landscape. *Physical Review A* **1992**, *46*, 6724-6730.
- (76) Han, Y.; Steinbeck, C. An evolutionary algorithm based strategy for computer-assisted molecular structure elucidation. *Journal of Chemical Information and Computer Sciences* **2003**, Accepted for publication.
- (77) Steinbeck, C.; Han, Y. SENECA: A Platform-Independent, Distributed and Parallel System for Computer-Assisted Structure Elucidation in Organic Chemistry (<http://seneca.sourceforge.net>, visited on October 2003).
- (78) Sundaram, A.; Venkatasubramanian, V. Parametric sensitivity and search-space characterization studies of genetic algorithms for computer-aided polymer design. *Journal of Chemical Information and Computer Sciences* **1998**, *38*, 1177-1191.

- (79) Faulon, J.-L. Stochastic Generator of Chemical Structure. 2. Using Simulated Annealing To Search the Space of Constitutional Isomers. *Journal of Chemical Information and Computer Sciences* **1996**, *36*, 731-740.
- (80) Elyashberg, M. E. Expert systems for the determination of structures of organic molecules by spectral methods. *Uspekhi Khimii* **1999**, *68*, 579-604.
- (81) Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning; Addison-Wesley: Reading, MA,; Addison-Wesley Pub Co, 1989.
- (82) Frey, C. An evolutionary algorithm with local search and classification for conformational searching. *Match-Communications in Mathematical and in Computer Chemistry* **1998**, 137-159.
- (83) Nair, N.; Goodman, J. M. Genetic algorithms in conformational-analysis. *Journal of Chemical Information and Computer Sciences* **1998**, *38*, 317-320.
- (84) Douguet, D.; Thoreau, E.; Grassy, G. A genetic algorithm for the automated generation of small organic molecules: Drug design using an evolutionary algorithm. *Journal of Computer-Aided Molecular Design* **2000**, *14*, 449-466.
- (85) Clark, D. E. An overview of evolutionary algorithm applications in computer-aided molecular design. *Abstracts of Papers of the American Chemical Society* **2001**, *221*, 25-COMP.
- (86) Globus, A.; Lawton, J.; Wipke, T. Automatic molecular design using evolutionary techniques. *Nanotechnology* **1999**, *10*, 290-299.
- (87) Nachbar, R. B. Molecular Evolution: Automated Manipulation of Hierarchical Chemical Topology and Its Application to Average Molecular Structures. *Genetic Programming and Evolvable Machines* **2000**.
- (88) Meiler, J.; Will, M. Genius: A genetic algorithm for automated structure elucidation from C-13 NMR spectra. *Journal of the American Chemical Society* **2002**, *124*, 1868-1870.
- (89) Bremser, W. Expectation Ranges of 13-C NMR Chemical Shifts. *Magnetic Resonance in Chemistry* **1985**, *23*, 271-275.

- (90) Steinbeck, C.; Kuhn, S.; Krause, S. NMRShiftDB - An Open-Submission, Open-Retrieval Database for Organic Structures and their NMR Data. *Magnetic Resonance in Chemistry* **2003**, submitted.
- (91) Steinbeck, C.; Kuhn, S.; Krause, S. NMRShiftDB - Constructing a Chemical Information System with Open Source Components. *Journal of Chemical Information & Computer Sciences* **2003**, Accepted for publication.
- (92) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *Journal of Chemical Information and Computer Sciences* **1988**, *28*, 31-36.
- (93) Weininger, D.; Weininger, A.; Weininger, J. L. SMILES 2. Algorithm for Generation of Unique SMILES Notation. *Journal of Chemical Information and Computer Sciences* **1989**, *29*, 97-101.
- (94) Mitchell, M. *An Introduction to Genetic Algorithms*; The MIT Press, 1996.
- (95) Tsipouras, A.; Ondeyka, J.; Dufresne, C.; Lee, S.; Salituro, G. et al. Using similarity searches over databases of estimated c-13 nmr spectra for structure identification of natural product compounds. *Analytica Chimica Acta* **1995**, *316*, 161-171.
- (96) Back, T. *Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms*; Oxford University Press: New York, 1996.
- (97) Willett, P. Chemoinformatics - similarity and diversity in chemical libraries. *Current Opinion in Biotechnology* **2000**, *11*, 85-88.
- (98) Christie, B. D. The Role of Two-Dimensional Nuclear Magnetic Resonance Spectroscopy in Computer-Enhanced Structure Elucidation. *Journal of the American Chemical Society* **1991**, *113*, 3750-3757.

Curriculum Vitae

Personal Data

Name: Yongquan Han
Title: Master of Science (Material Science)
Nationality: Chinese
Date of birth: 25.09.1970
Place of birth: Shanxi, China
Marital status: Married (Weiwei Zeng)

Colleges

09/1986 - 07/1990 B.Sc. in Material Science
University of Electronic Science and Technology of China (UESTC), Chengdu, China

09/1993 - 05/1996 M.Sc. in Material Science
University of Electronic Science and Technology of China (UESTC), Chengdu, China
Thesis: Experimental investigation and theoretical simulation of LISCON-based solid state ionic conductor with 3D network channels
Supervisor: Prof. Ai Chen

04/2000 – present Doctorate Candidate in Chemistry
Max-Planck-Institute for Chemical Ecology and Friedrich-Schiller-University
Thesis: Evolutionary Algorithm as an Approach for Computer Assisted Structure Elucidation of Organic and Bioorganic Compounds
Supervisors: Dr. Christoph Steinbeck (MPI) and Prof. Dr. Ernst Anders (FSU)

Publications

Papers

- [1] Han, Y.; Steinbeck, C. An evolutionary algorithm based strategy for computer-assisted molecular structure elucidation. *Journal of Chemical Information and Computer Sciences* 2003, Accepted for publication.
- [2] Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E. et al. The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *Journal of Chemical Information & Computer Sciences* 2003, 43, 493-500.

Lectures and Posters

- [1] Jährliche Tagung der Fachgruppe Chemie-Information-Computer in der GdCH, Kleinmachnow bei Berlin, 11/2002. Steinbeck, C. und Han, Y., „Werkzeuge für die Computergestützte Strukturaufklärung Kleiner Organischer Moleküle.“
- [2] Virtuelle Hochschule Bioinformatik, Erlangen, Würzburg, Bayreuth, 05/2002: Steinbeck, C. und Han, Y., Stochastic Algorithms for the Computer Assisted Structure Elucidation of Small Organic Molecules.
- [3] Fränkisch-Mitteldeutsches Naturstoffchemiker-Treffen in Leipzig 04/2002 Oral talk: Stochastic Algorithms for Computer Assisted Structure Elucidation of Plant Secondary Metabolites
- [4] Vortrag: Neue N-Acyl-aminosäurekonjugate von herbivoren Insekten
- [5] Analytica Konferenz, München 04/2002: Steinbeck, C. und Han, Y., “Stochastic Algorithms for Computer Assisted Structure Elucidation of Small Organic Molecules.”

- [6] Biochemisches Kolloquium, Institut für Biochemie, Universität Köln, 04/2002; Steinbeck, C. und Han, Y., „The SENECA Structure Elucidation System.“
- [7] Chemisches Kolloquium Universität Darmstadt, 02/2002: Steinbeck, C. und Han, Y., „Stochastische Algorithmen in der Computergestützten Strukturaufklärung Organischer Moleküle.“
- [8] Chemiedozententagung Leipzig 03/2001: Steinbeck, C. und Han, Y., „Neue Verfahren zur Automatischen Strukturaufklärung“.

Selbstständigkeitserklärung

Ich erkläre, dass ich vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Hilfsmittel und Literatur angefertigt habe.

Jena, den 13.10. 2003

Acknowledgement

The thesis describes work carried out, as part of a Deutsche Forschungsgemeinschaft (DFG) project under the guidance of Dr. Christoph Steinbeck. A perfect scientific working environment and administrative support was provided by the Max-Planck-Gesellschaft.

Foremost, among those I would like to thank, are my supervisor and former Group Leader Dr. Christoph Steinbeck at the Max-Planck-Institute for Chemical Ecology (now with the University of Cologne), for his introducing me to the field of chemoinformatics. He was the main source of inspiration for me to develop interests in this promising field, and helped me to grow from being a student, to become a more mature and responsible scientist.

I am also deeply indebted to Prof. Dr. W. Boland, the director of the Max-Planck-Institute for Chemical Ecology, for the support he provided particularly during the second stage of my doctoral research studies. Despite the differences in our background, his patient and confidences were essential in improving the final version of this work.

Sincere thanks are extended in particular to the group colleagues at MPICE, for their heartfelt support and concern and making me feeling like home here in Jena.

My family, my beloved parents and sisters, deserves very special thanks for their never-ending love and persistent encouragement throughout life. And of course, my dear Weiwei, I can not image what will happen without her love. This thesis is dedicated to her, but I know it is nothing compared to those she did for me.

