PRIORITY RESEARCH CENTRE FOR

BIOINFORMATICS, BIOMARKER DISCOVERY & INFORMATION-BASED MEDICINE

Using Evolution and L-systems for Intelligent Design and other attempts to narrow the gap between Theory and Practice in Computer Science"

Chief Investigator
ARC Centre of Excellence
in Bioinformatics
(2006-2015)

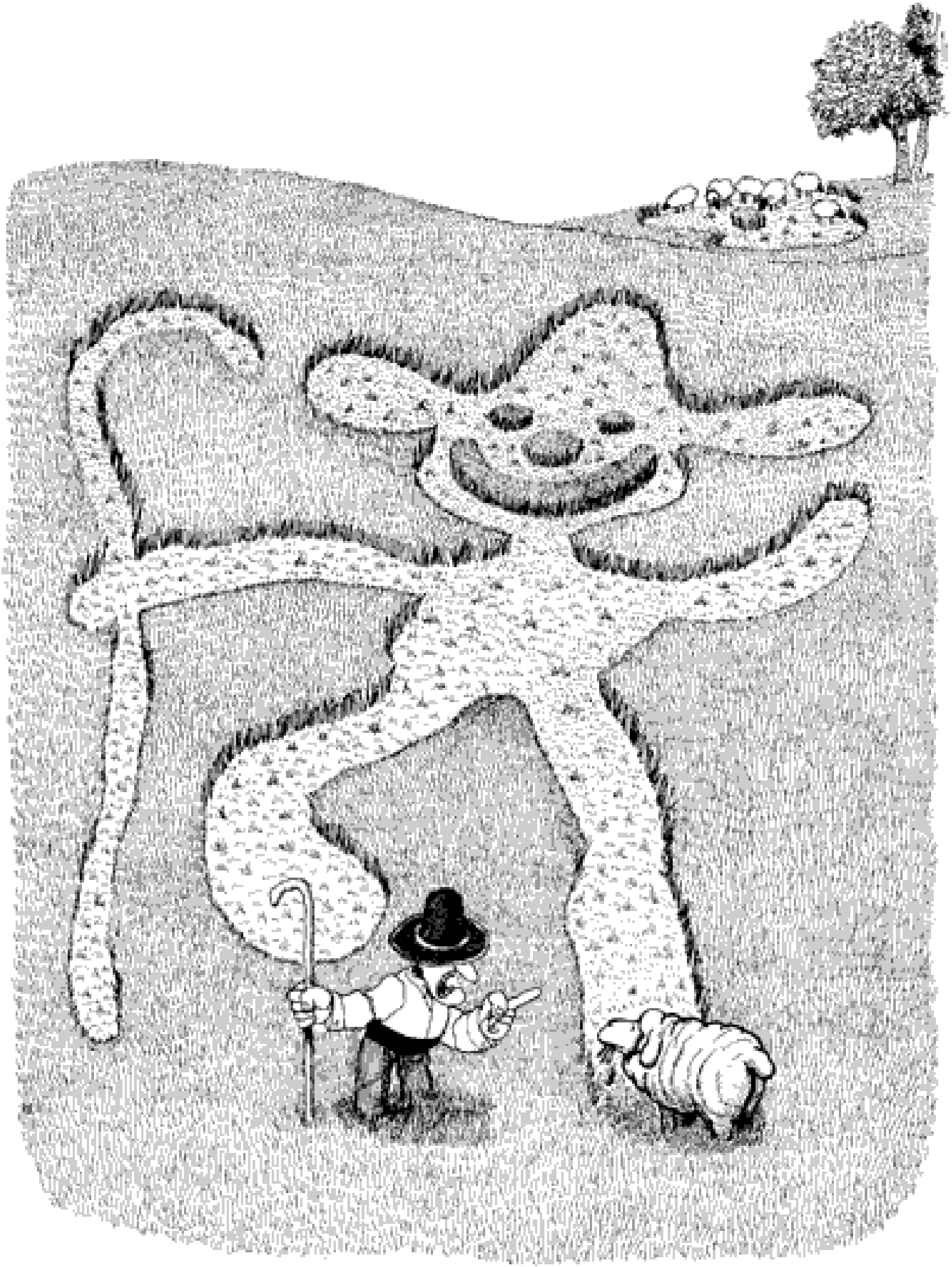
Australian Research Council
Future Fellow
(2012-2016)

Australian Government
Australian Research Council

Founding Director
Centre for Bioinformatics,
Biomarker Discovery and
Information-based Medicine
(2006-current)

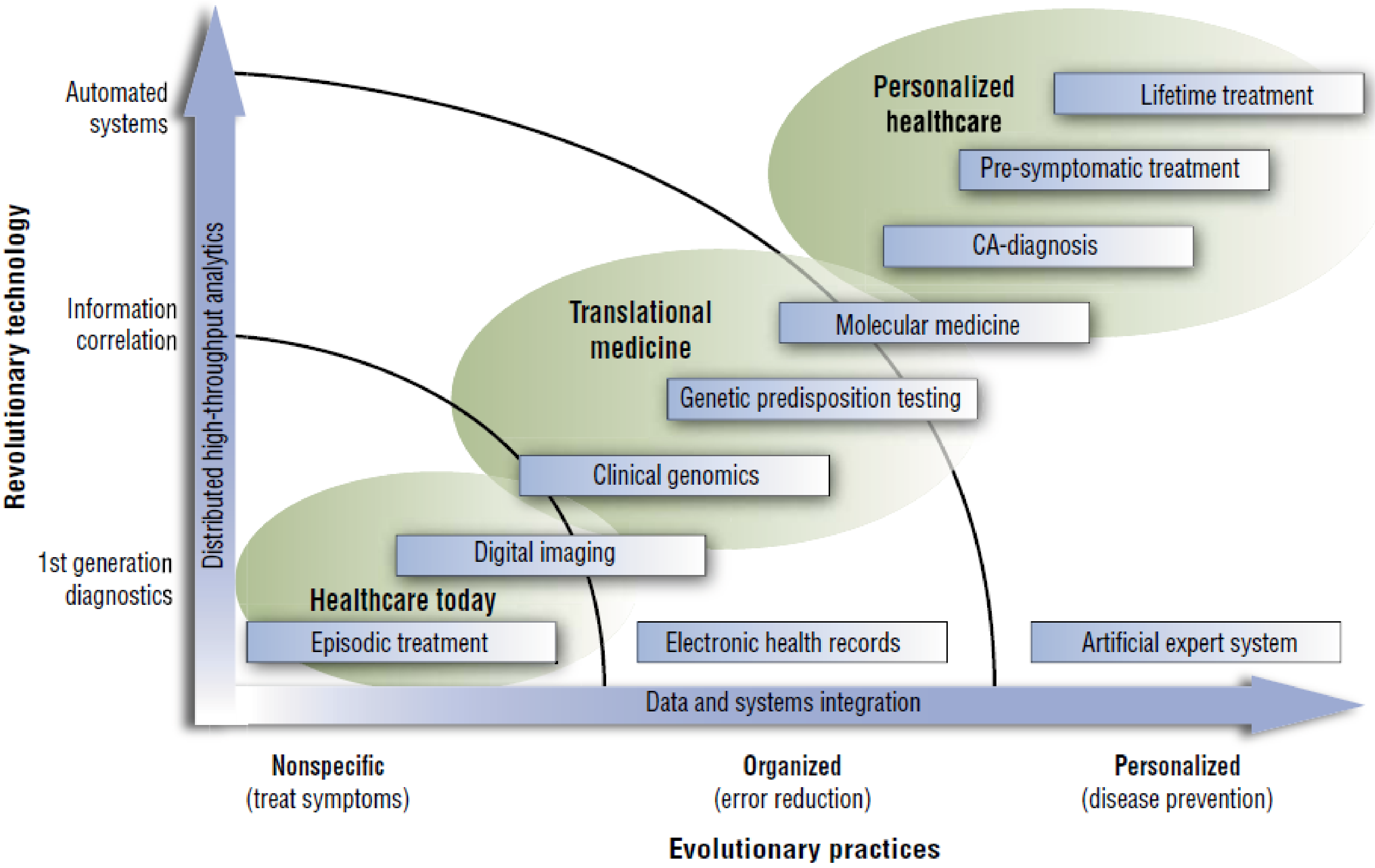
Follow us

Pablo Moscato (on LinkedIn) and **Prof. Pablo Moscato** (on Facebook)



Are you ready for Information-based Medicine ?

Figure 1: Market trends and drivers: revolutionary technologies and evolutionary practices.

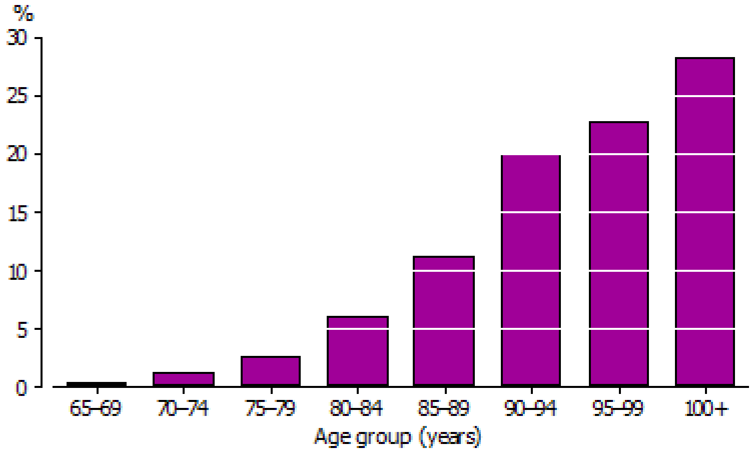


Source: IBM Life Sciences Solutions

Picture from: Personalized Healthcare 2010: Are you ready for information-based Medicine?

http://www-935.ibm.com/services/in/index.wss/ibvstudy/igs/x1022583?cntxt=x1022520

Dementia and Alzheimer's Disease (AD)



Proportion of each age group identified as having dementia or AD.

(ABS 2009, Survey of Disability, Ageing and Careres).

One new case of dementia every 6 minutes.

1.2 million Australians are caring with somebody with dementia.

By the 2060s, spending on dementia is set to outstrip that of any other health condition.

Our approach is largely based on Combinatorial Optimization and Mathematical Modelling

Make *sense* of a deluge of data/information.

Find *hidden patterns* (e.g. those only present in samples from patients that have a disease).

Make *predictions* (e.g. who can get a disease).

It could inform *particular* lifestyle changes

development of new models

modify clinical practice

translational research

Our problems are big and challenging

SNPs (Single Nucleotide Polymorphisms)

Differences in DNA bases common to a certain small percentage of the population

Combinations of them can lead to increased disease risks

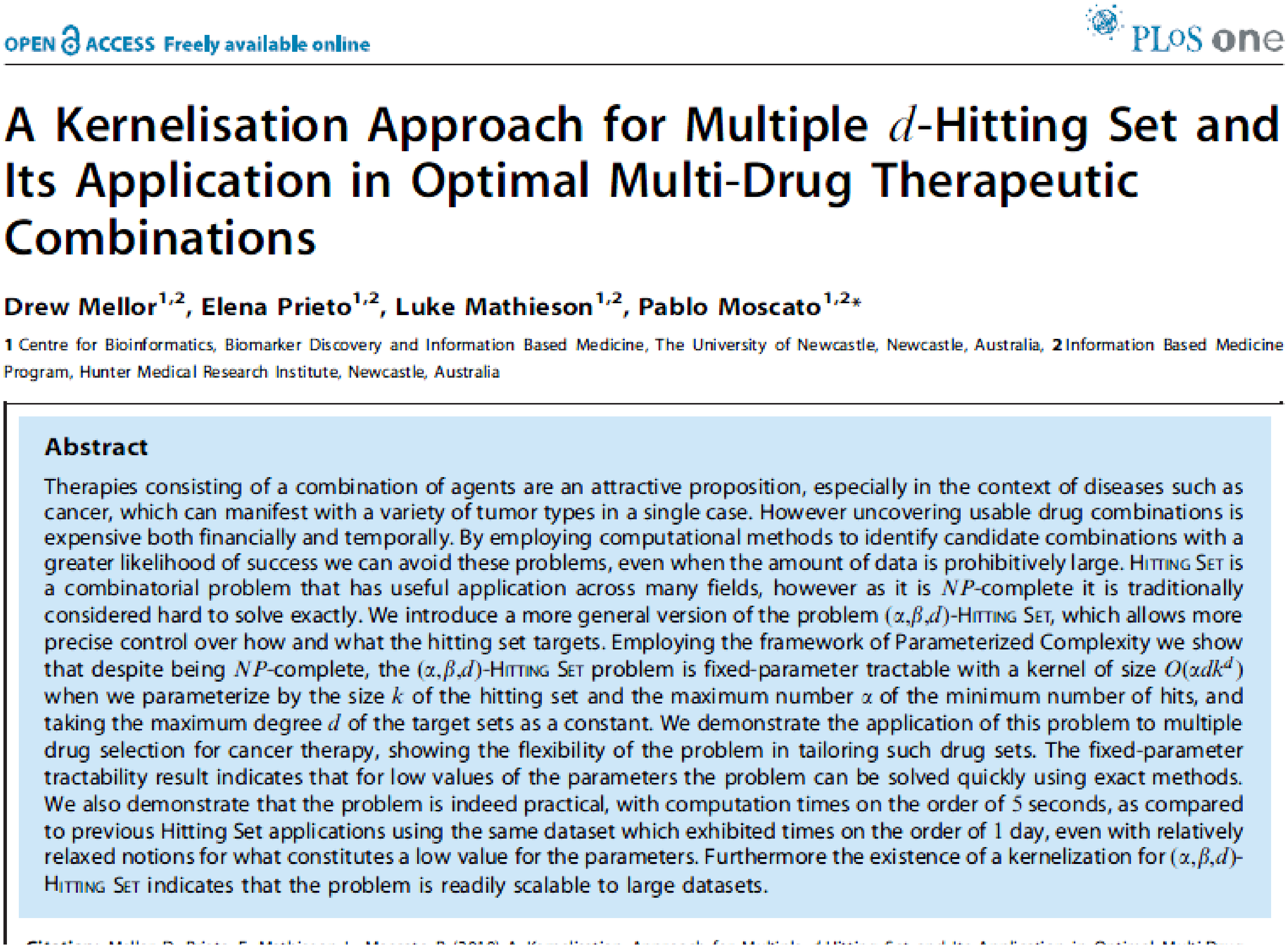
Typically 4,000 to 100,000 samples, ~ 500,000 SNPs

Gene Expression (Microarrays)

Typically 200 to 2,000 samples, ~ 50,000 probe sets

"Extended" Gene Expression datasets

Typically <2000 samples, ~ 2,500,000,000 pairs of probe sets



OPEN ACCESS Freely available online

PLoS one

A Kernelisation Approach for Multiple d-Hitting Set and Its Application in Optimal Multi-Drug Therapeutic Combinations

Drew Mellor[1,2], Elena Prieto[1,2], Luke Mathieson[1,2], Pablo Moscato[1,2]*

[1] Centre for Bioinformatics, Biomarker Discovery and Information Based Medicine, The University of Newcastle, Newcastle, Australia, [2] Information Based Medicine Program, Hunter Medical Research Institute, Newcastle, Australia

Abstract

Therapies consisting of a combination of agents are an attractive proposition, especially in the context of diseases such as cancer, which can manifest with a variety of tumor types in a single case. However uncovering usable drug combinations is expensive both financially and temporally. By employing computational methods to identify candidate combinations with a greater likelihood of success we can avoid these problems, even when the amount of data is prohibitively large. Hitting Set is a combinatorial problem that has useful application across many fields, however as it is NP-complete it is traditionally considered hard to solve exactly. We introduce a more general version of the problem (α,β,d)-Hitting Set, which allows more precise control over how and what the hitting set targets. Employing the framework of Parameterized Complexity we show that despite being NP-complete, the (α,β,d)-Hitting Set problem is fixed-parameter tractable with a kernel of size $O(\alpha dk^d)$ when we parameterize by the size k of the hitting set and the maximum number α of the minimum number of hits, and taking the maximum degree d of the target sets as a constant. We demonstrate the application of this problem to multiple drug selection for cancer therapy, showing the flexibility of the problem in tailoring such drug sets. The fixed-parameter tractability result indicates that for low values of the parameters the problem can be solved quickly using exact methods. We also demonstrate that the problem is indeed practical, with computation times on the order of 5 seconds, as compared to previous Hitting Set applications using the same dataset which exhibited times on the order of 1 day, even with relatively relaxed notions for what constitutes a low value for the parameters. Furthermore the existence of a kernelization for (α,β,d)-Hitting Set indicates that the problem is readily scalable to large datasets.

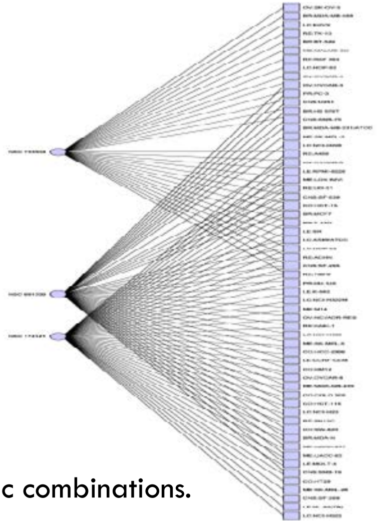
Citation: Mellor D, Prieto E, Mathieson L, Moscato P (2010) A Kernelisation Approach for Multiple d-Hitting Set and Its Application in Optimal Multi-Drug

Selection of optimal multi-drug therapeutic combinations

A Princeton researcher found in 2010 that a combination of three drugs (out of tens of thousands tested by the National Cancer Institute) was able to "target" efficiently all cell lines of the NCI60 panel.

The problem: Takes 24 hours, and does not have a performance guarantee (IP could be lost by working with suboptimal solutions). Is there a better solution?

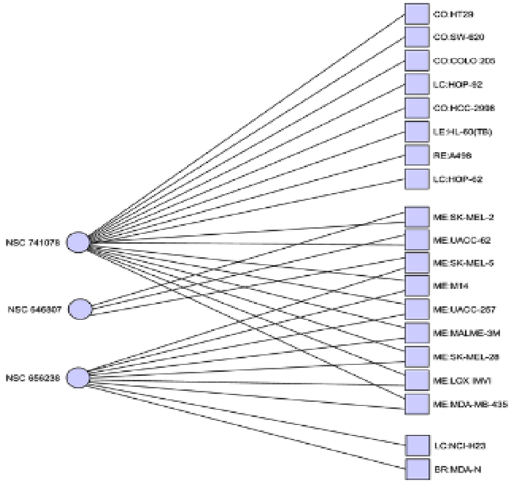
Our Solution: In less than two months, we showed (top right corner) that indeed three is the optimal number, but that there are other solutions, and that the best can be found in five seconds with our methods.



A kernelisation approach for multiple d-Hitting Set and its application in optimal multi-drug therapeutic combinations.
Mellor D, Prieto E, Mathieson L, **Moscato P**.
PLoS One. 2010 Oct 18;5(10):e13055

Potential: Guide the design of therapeutic combination approaches with biologists "in the loop" as the fast turnaround of our methods would allow them to iterate between in silico and wet lab experimentation.

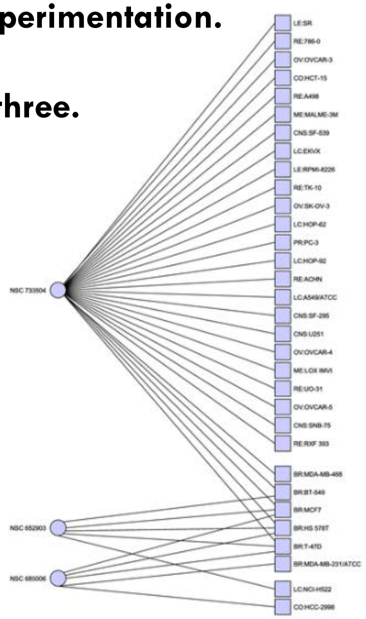
Note: Everolimus/Afinitor (Novartis) appeared as part of the group of selected three.



(Left) A Minimal set of drugs (three) that hit melanoma cell lines at least two times and all other cell lines zero (not shown) or one time.

(Right) A Minimal set of drugs (three) that breast cancer cell lines (excluding the disputed MDA-N cell line)

Relaxing the restriction on hitting non-breast cancer cell lines it is possible to hit more BC cell lines repeatedly.



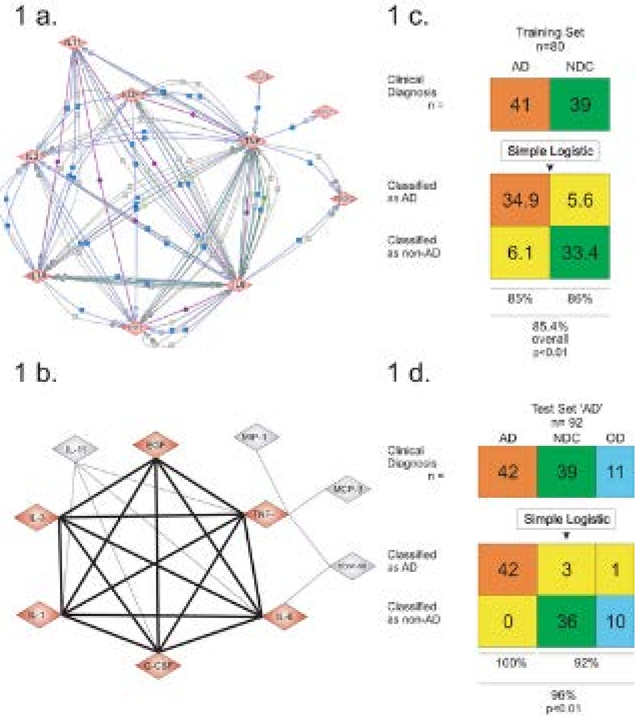
Alzheimer's Disease diagnosis

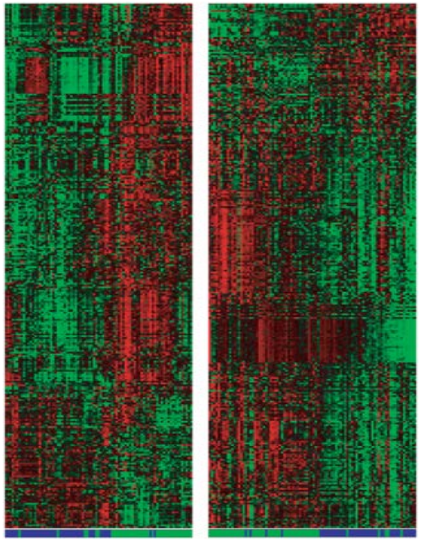
Stanford researchers found that a panel of 18 proteins could predict clinical AD five years in advance (*Nature Medicine*, 2007)

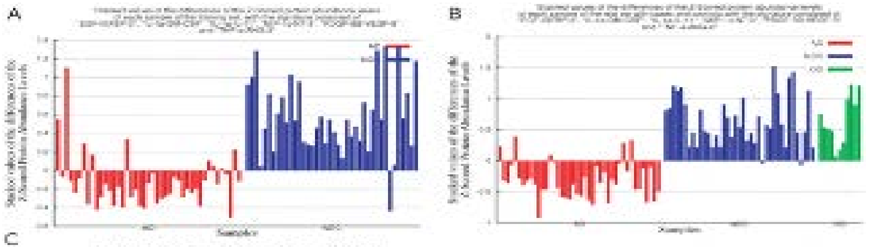
In less than two months, we showed that we could reduce the panel to only 5, maintaining the performance quite independently of the classifier

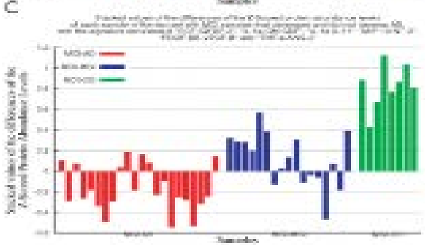
Gómez Ravetti M, **Moscato P** (2008) Identification of a 5-Protein Biomarker Molecular Signature for Predicting Alzheimer's Disease. PLoS ONE 3(9): e3111. doi:10.1371/journal.pone.0003111

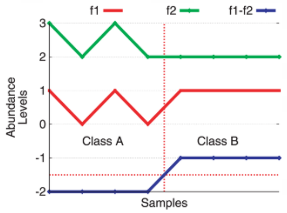
Differences in Abundances of Cell-Signalling Proteins in Blood Reveal Novel Biomarkers for Early Detection Of Clinical Alzheimer's Disease











There is hope to the possibility of separating Mild Cognitive Impaired individuals that progress to AD from does that progress to other dementias (in green)

Identification of Genome-Wide SNP–SNP and SNP–Clinical Boolean Interactions in Age-Related Macular Degeneration

Or "How I Learned to Stop Worrying… about spending one million dollars"

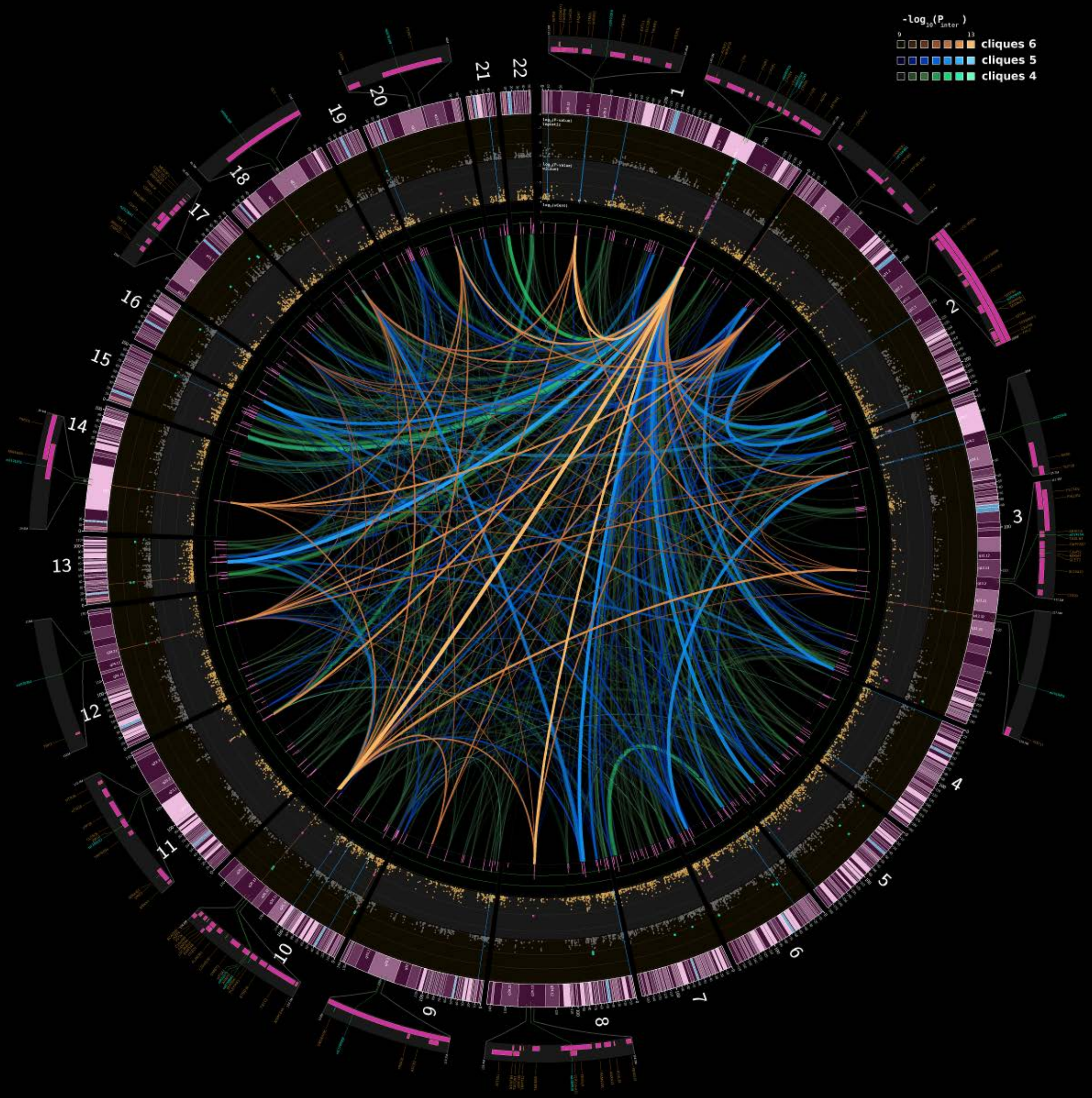
The Problem: Computing all possible SNP-SNP pairs of interactions in several genome-wide association studies. The need is to identify a large number of significant interactions to then correlate with statistically relevant "hits" on pathways and gene ontology associations.

x

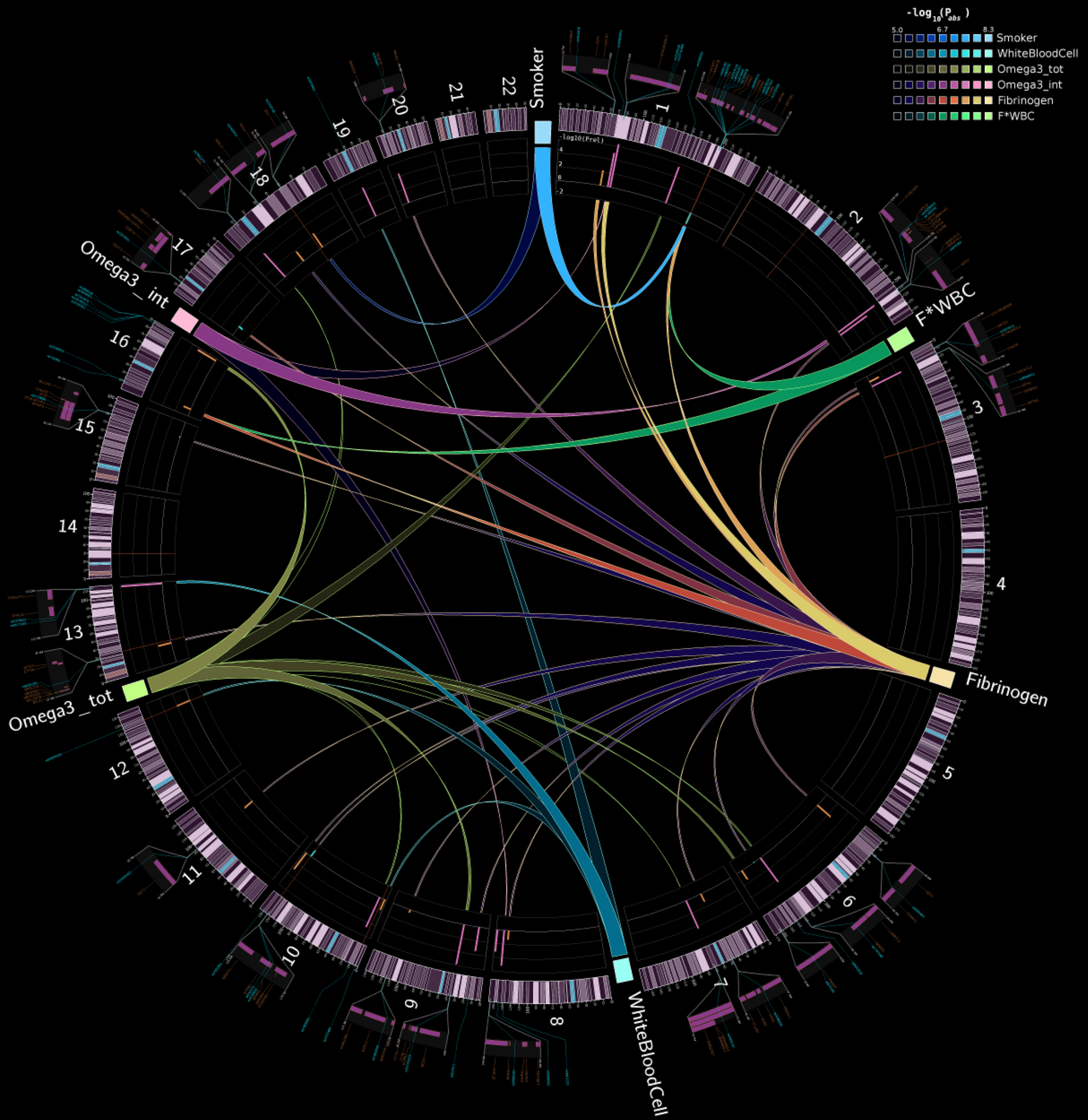
The Solution: Statistical analysis of SNP-SNP pairs of interactions in several genome-wide association studies via **GPU computing**. Followed by **combinatorial optimisation techniques** to identify **cliques in weighted graphs.** Identification of commonalities in clique via Gene Ontology **statistical associations**, leading to working hypotheses and predictive analytics.

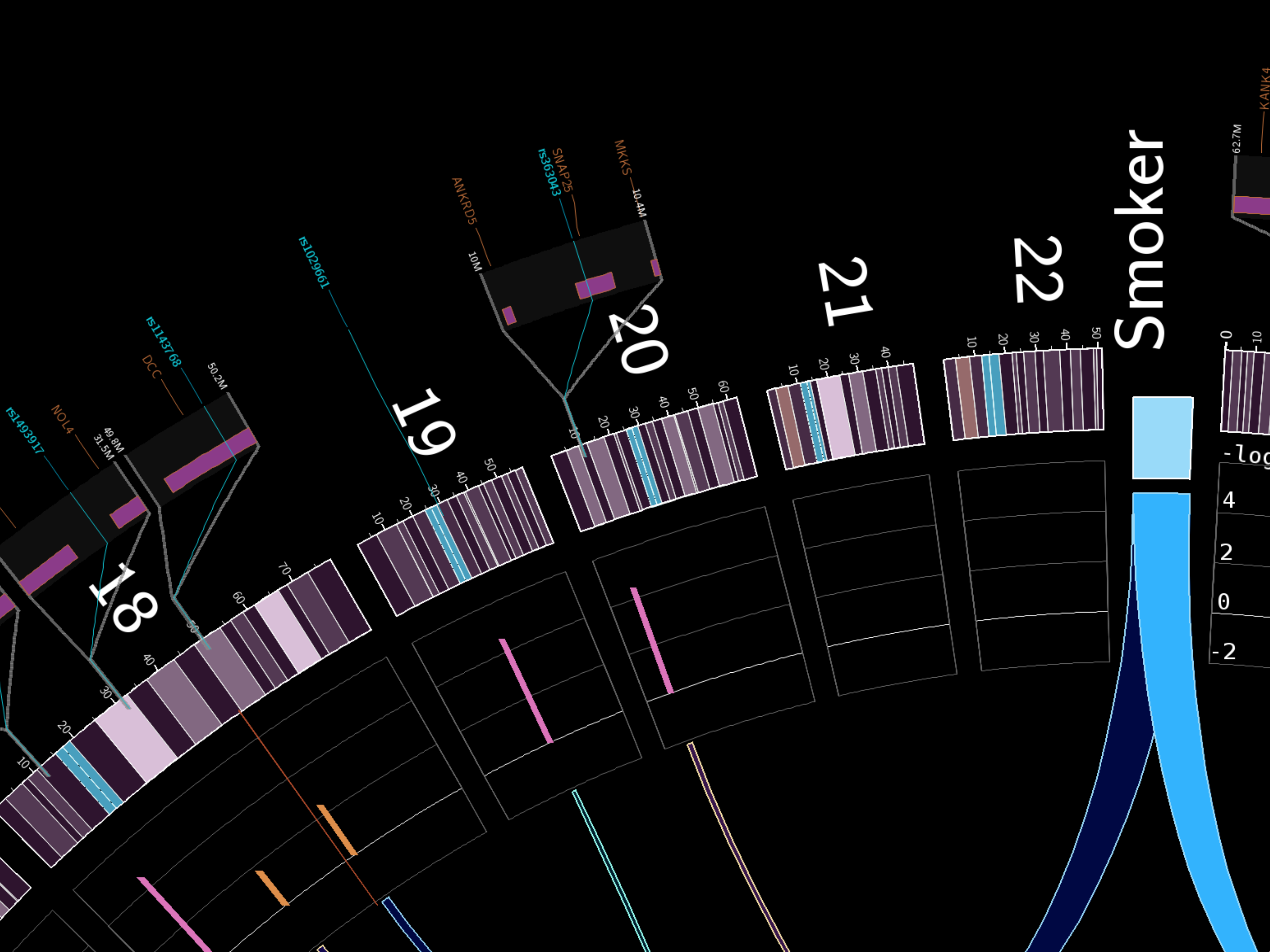
Prediction now confirmed: Netrin-DCC axis in 77,000 samples.

Our solution: We prioritized the number of SNPs/loci for further investigation using highly sophisticated supercomputer-based algorithms. We focused on identifying highly connected groups (these are called cliques, quasi-cliques in Mathematics). We illustrate here how we developed whole genome visualization methods which are employed to reveal the most important interactions in the largest cliques. The observed topologies can be analysed and they are shown to be highly complex but not random. This gives novel insights in multifactorial diseases.

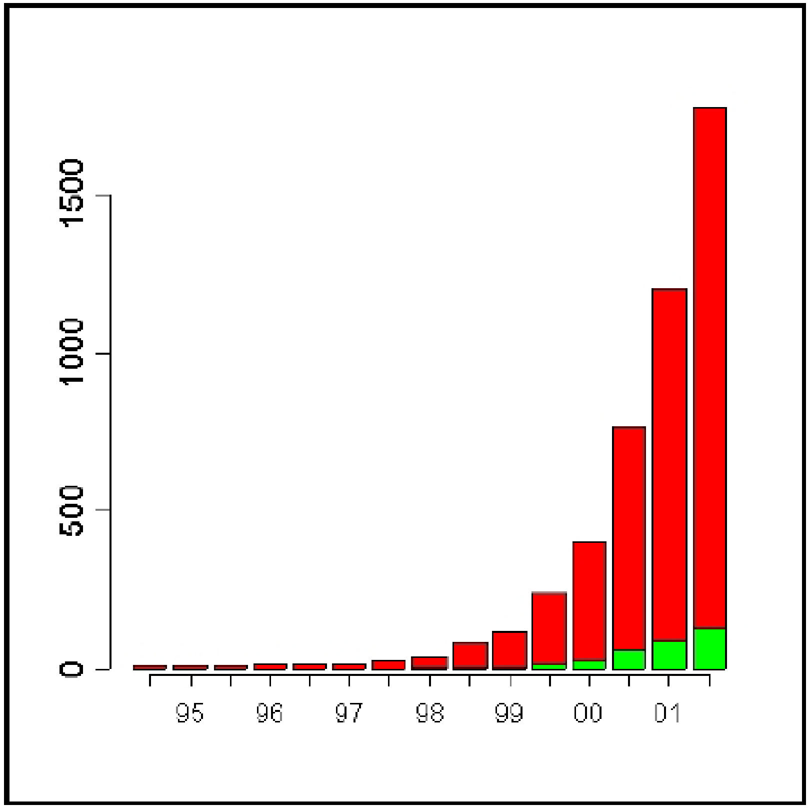


Novel whole genome visualization techniques have also been developed to uncover interactions of SNPs with environmental variables. The integration of the information of these two types of maps helps life scientists by providing new working hypotheses for wet lab validations.




KANK4
62.7M
MKKS
10.4M
SNAP25
rs363043
ANKRD5
10M
rs1029661
rs1143768
DCC
50.2M
rs1493917
NOL4
49.8M
31.5M
18
19
20
21
22
Smoker
10
20
30
40
50
60
70
0
-log
4
2
0
-2

Impact of microarray methods in the Life Sciences and Medical Research



Cumulative number of papers in PubMed with keywords:

“Microarray”

“Microarray+clustering”

TODAY (June 4, 2003):

3557 !

From 10 to 1000:

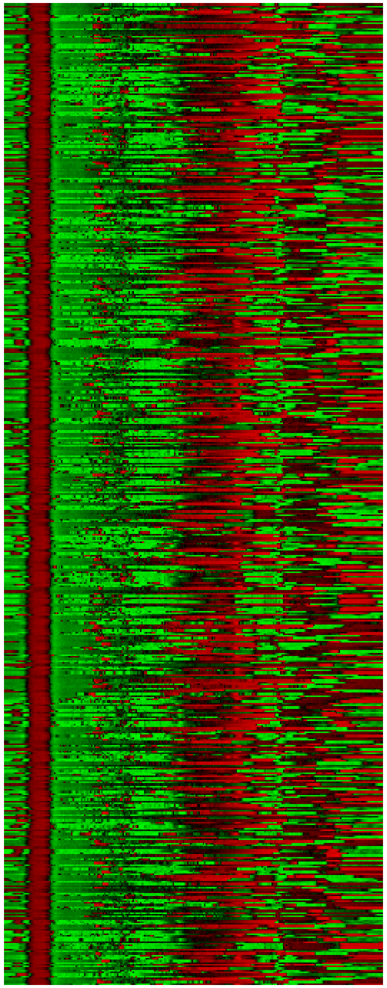
“Sequence analysis”: 1967 - 1982

“Microarray”: 1996 - 2001

Jan. 17, 2005: 8,592 June 13, 2015: **67,758** !!!

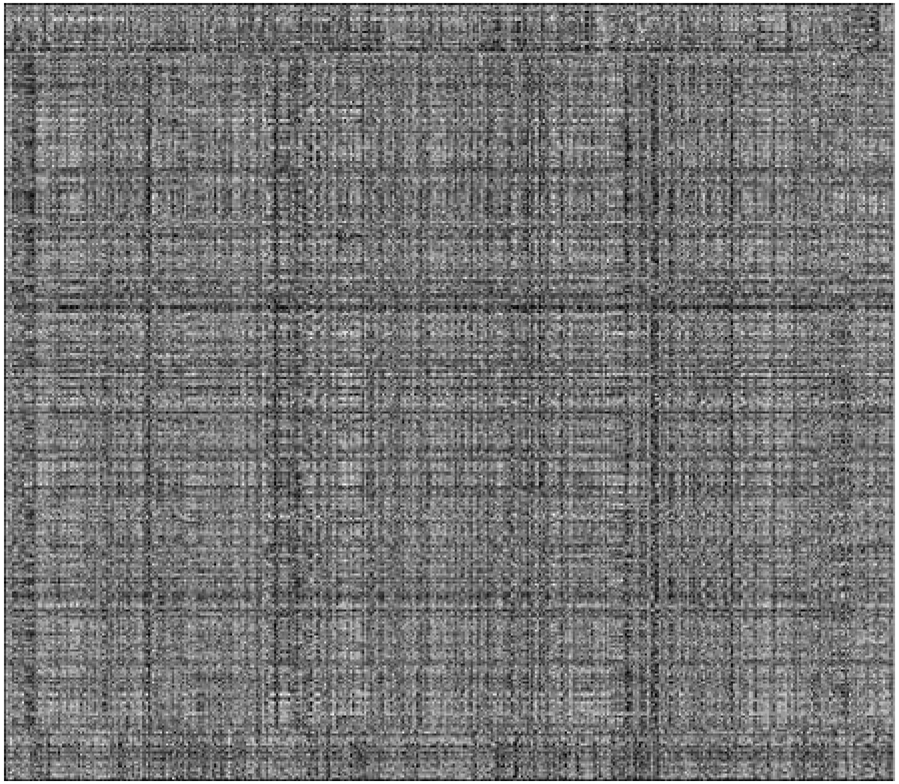
Microarray Clustering
(Deceivingly similar to the TSP problem –
a trap for countless computer scientists)

Instance size: 5120 genes x 512 experiments



Data Clustering a 930x930 matrix
(again deceivingly similar to the TSP problem,
this is of great relevance for the field of Systems Biology)

TSP algorithms have been suggested,
but on 930x930 cities…



Memetic Algorithms
27 years is a long time…

- I proposed the denomination in 1989 for work we started a year before.

- Since then several papers appeared, including:
  - Formal Memetic Algorithms
  - Competent Memetic Algoritms
  - …..   … so I am probably the pioneer of:
  - Informal and Incompetent Memetic Algorithms

Memetic Algorithms (Moscato, 1989 @ Caltech)

- Thousands of papers in all fields of science and technology
- A dedicated journal (Memetic Computing, Springer)
- IEEE dedicated a Task Force to the subject
- Many international Workshops, etc.

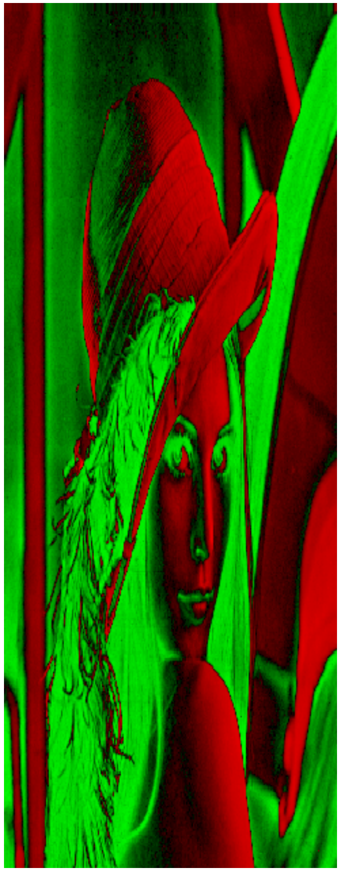
- In 2013, the IP & Science division of Thomson Reuters identified "Memetic Computing" as **one of the world's top ten research fronts** of the combined areas **of Mathematics, Computer Science and Engineering**. The selection was done from approximately "8,000 research fronts currently identified".

http://sciencewatch.com/sites/sw/files/sw-article/media/research-fronts-2013.pdf

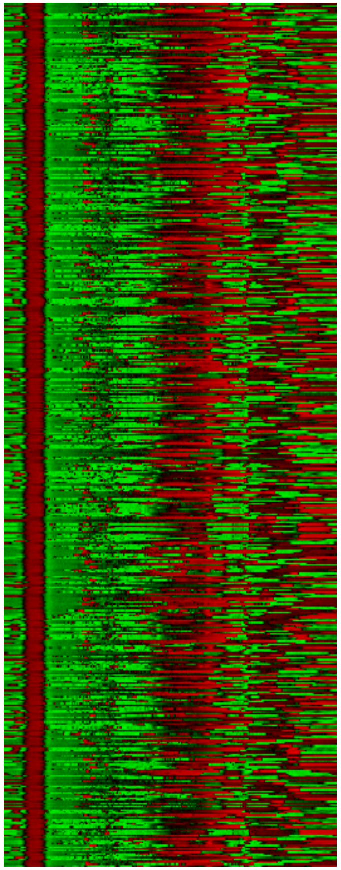
Building a “solved case” of Microarray Clustering to test performance of our methods

Instance size: 5120 genes x 512 conditions

Original Data



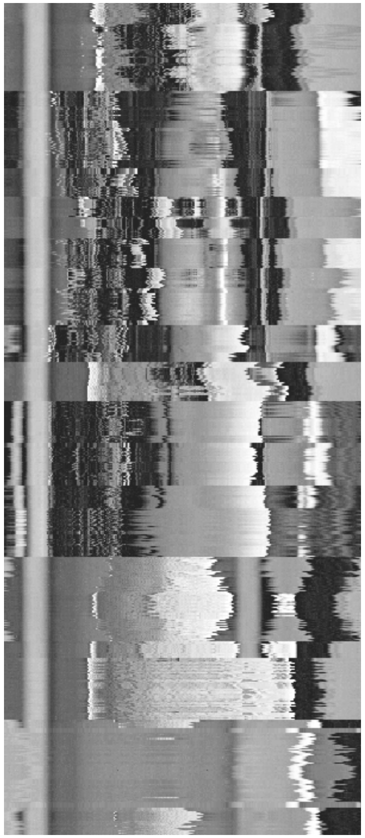
After randomization



Ordering microarray data

Lenna image – 5120 x 512 – No noise

Eisen (1998)



Cluster analysis and display of genome-wide expression patterns
www.pnas.org/content/95/25/14863.long
by MB Eisen - 1998 - Cited by 13,412

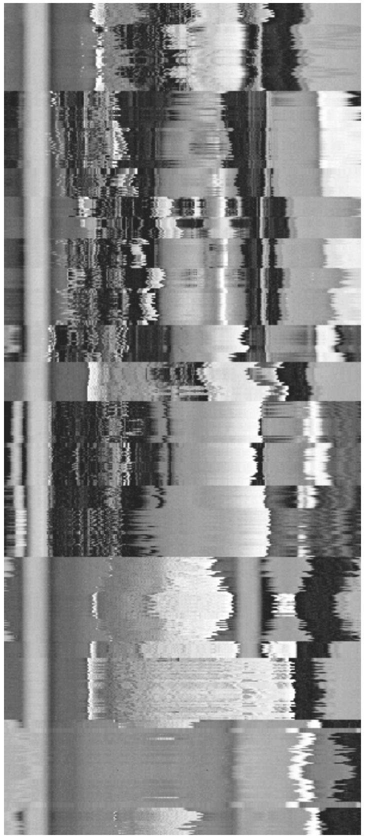
(from Google Scholar, August 1, 2013).

Cited by 15,001 papers available online!!

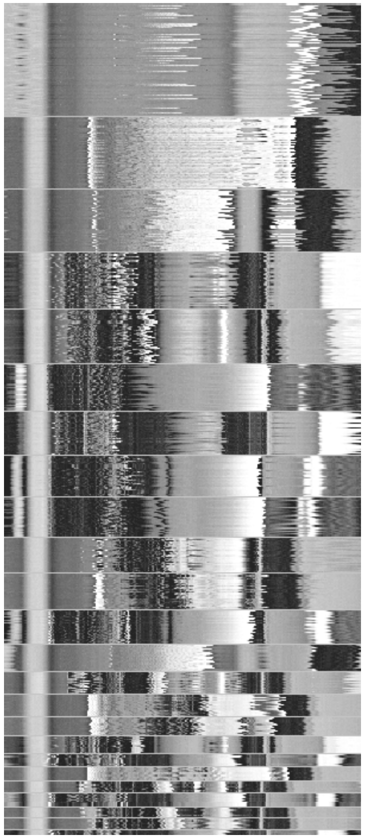
Ordering microarray data

Lenna image – 5120 x 512 – No noise

Eisen (1998)



CLICK(2001)

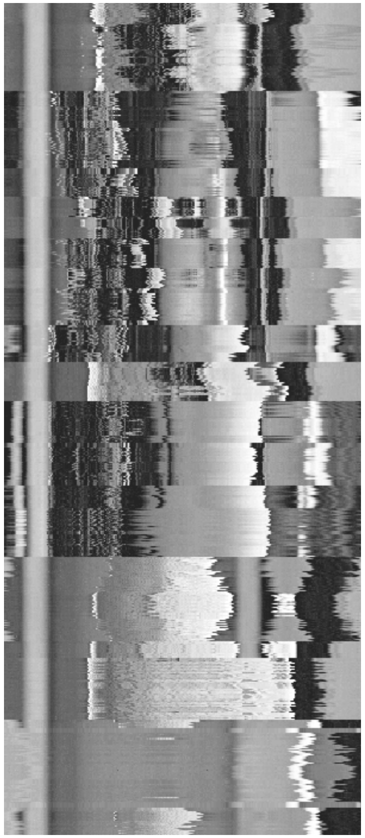


CLICK and EXPANDER: a system for clustering and visualizing gene expression data
R Sharan, A Maron-Katz, R **Shamir** -
Bioinformatics, 2003 - Oxford Univ Press
Cited by 289

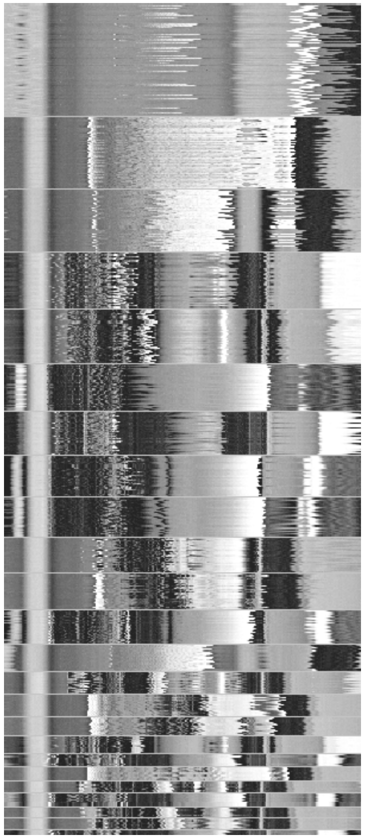
Ordering microarray data

Lenna image – 5120 x 512 – No noise

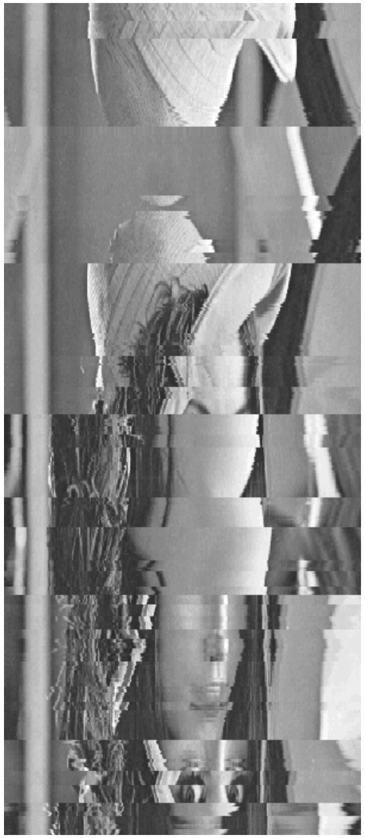
Eisen (1998)



CLICK(2001)



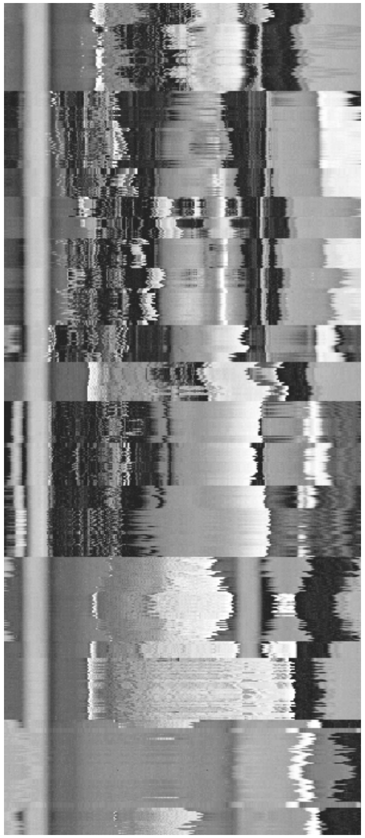
EBI (2003)



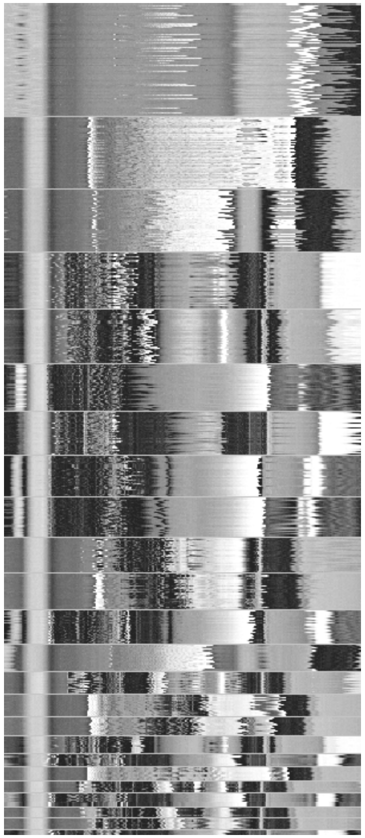
Ordering microarray data

Lenna image – 5120 x 512 – No noise

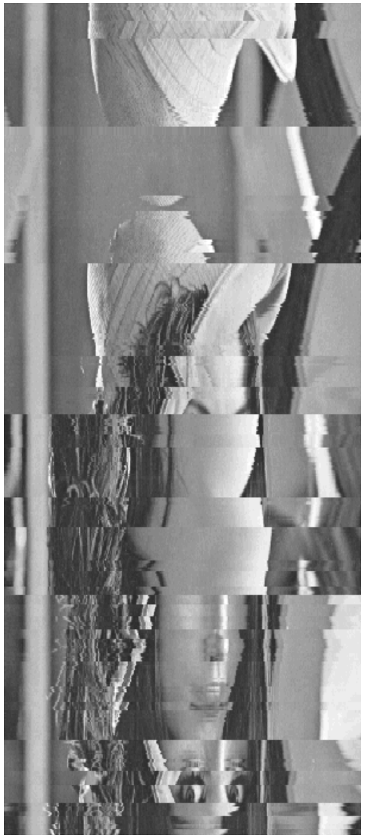
Eisen (1998)



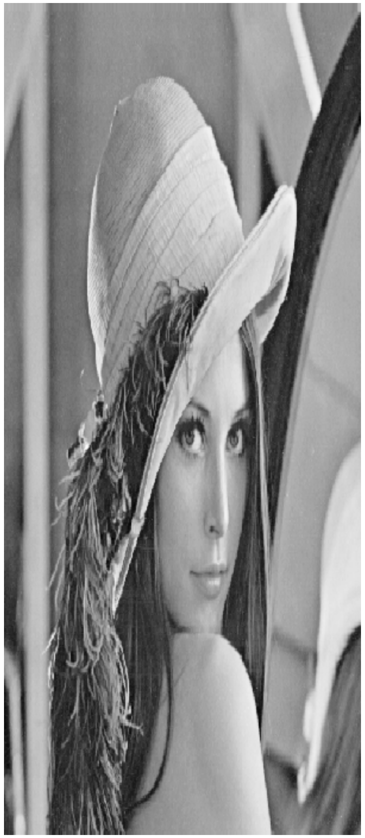
CLICK(2001)



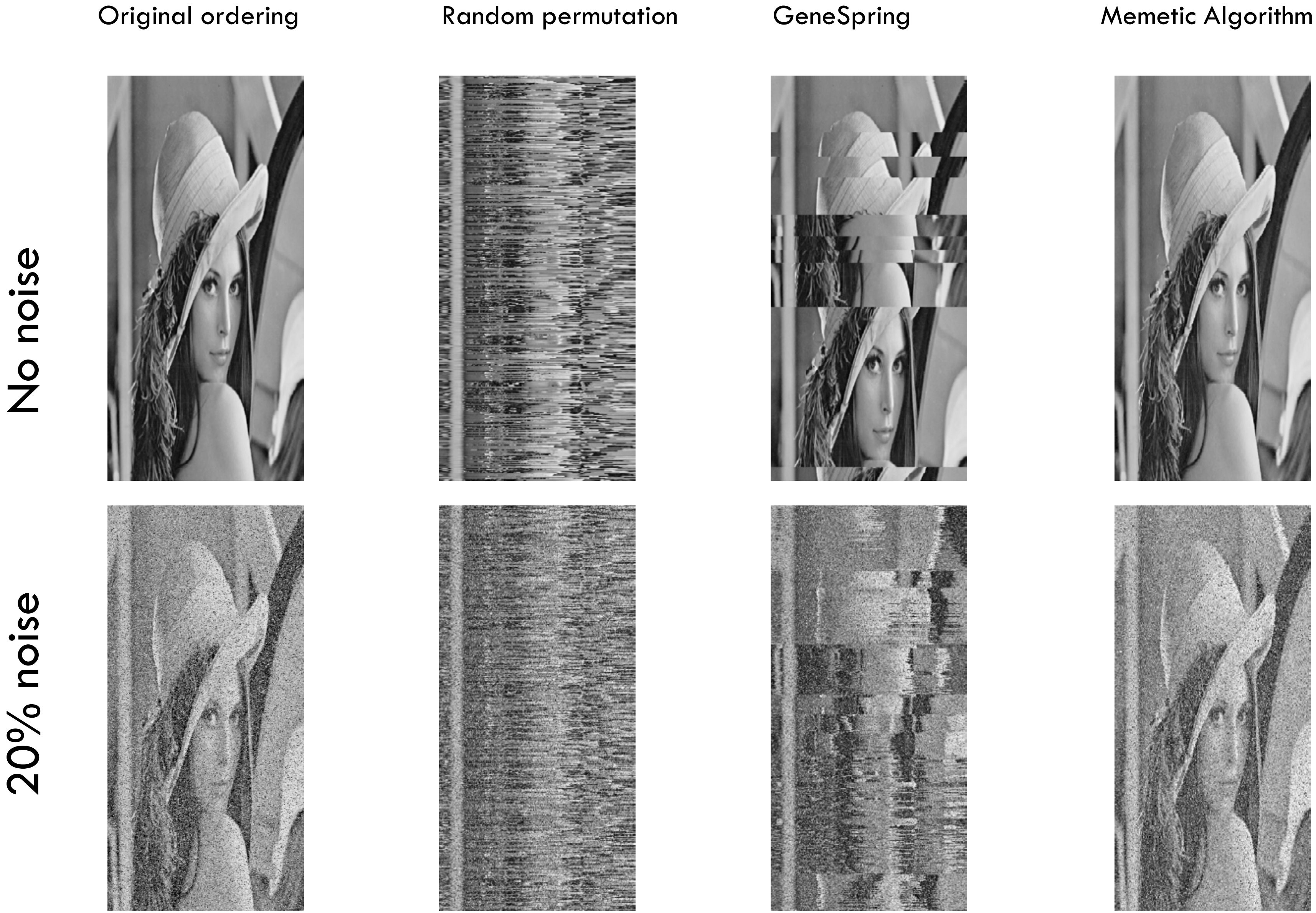
EBI (2003)



Our Memetic
Algorithm
(2007)



Top-off-the-shelf solutions vs. creating new algorithms



Original ordering

Random permutation

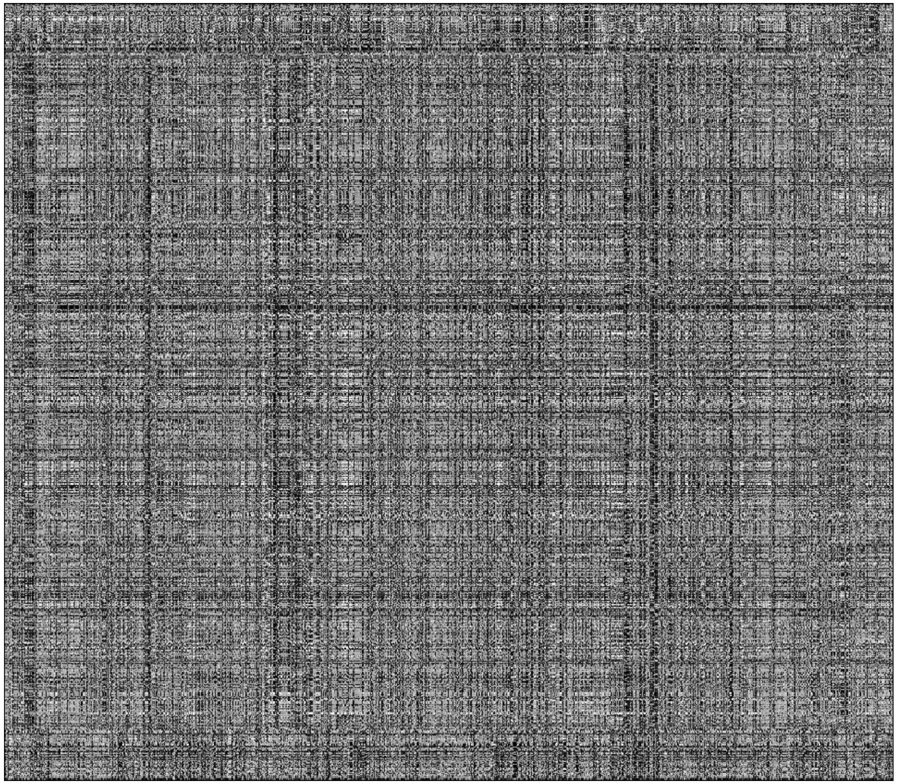
GeneSpring

Memetic Algorithm

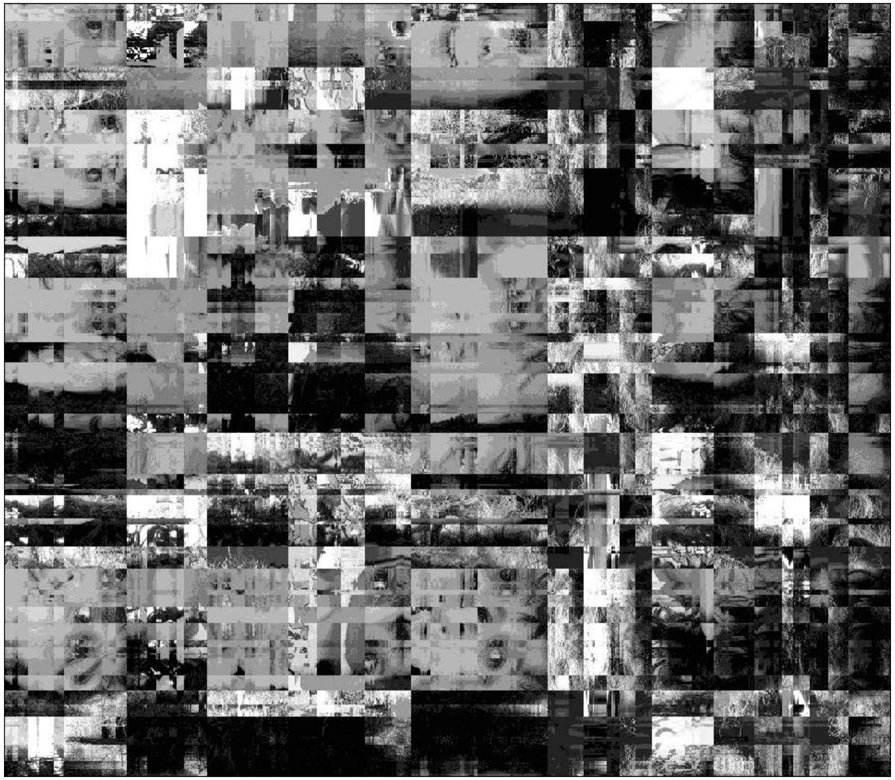
No noise

20% noise

Ordering samples and genes: A need for personalized medicine

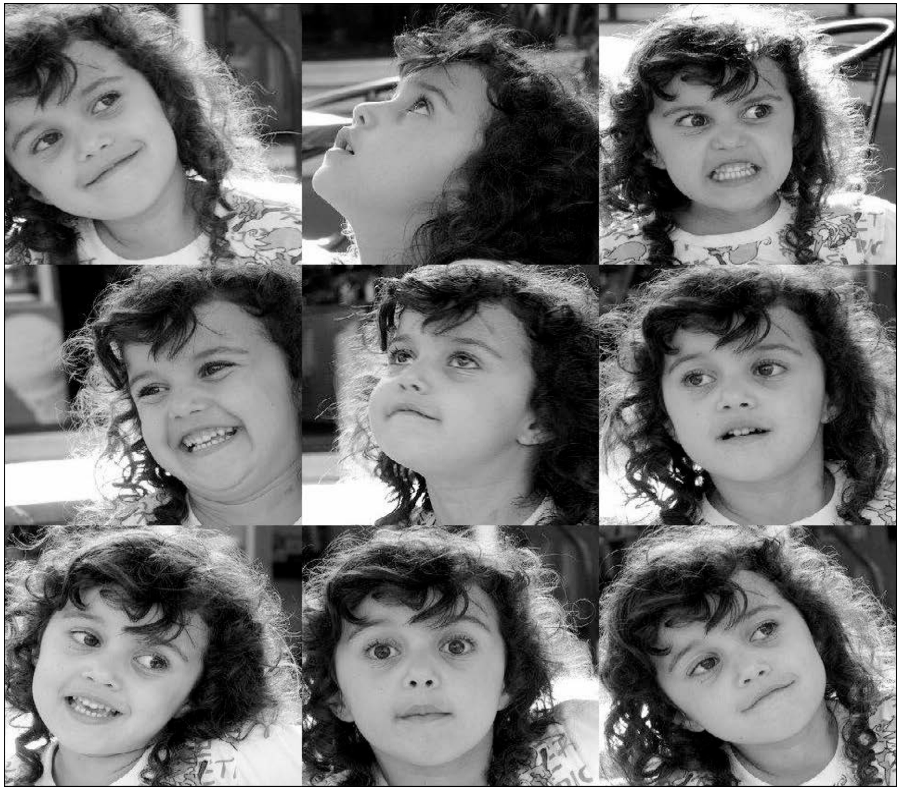


Input dataset (930 rows & 930 columns)

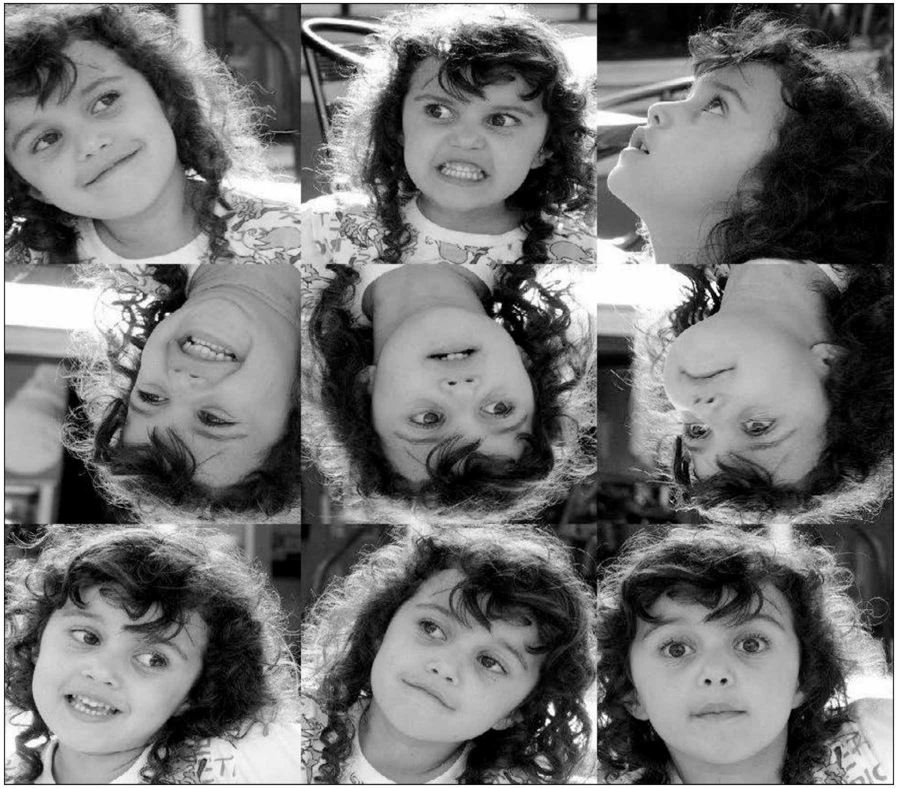


TIGR Multiexperiment Viewer software
developed by
The Institute for Genomic Research

Ordering samples and genes: A need for personalized medicine



Unperturbed dataset



Memetic Algorithm final result

Where the real challenges are…

*“From the point of view of the formal criteria that we have developed in this book, **the local improvement algorithms and their many variants are totally unattractive**. They do not in general return the optimal solution, they tend to have exponential worst-case complexity, and they are not even guaranteed to return solutions that are in any well-defined sense “close” to the optimum. Still, for many NP-complete problems, **in practice they often turn out to be the ones that perform best ! Explaining and predicting the impressive empirical success of some of these algorithms is one of the most challenging frontiers of the theory of computation today.** ”*

H. Lewis and C.H. Papadimitriou,
Elements of the Theory of Computation,
2nd edition…

1988 - MAs as “the next logical step”

- Many good algorithmic approaches exist for a problem.

- Local search methods are generally easy to implement and give good results for many, if not all (?), problems in NP.

- Exact algorithms can sometimes prove optimality, even for large instances. Problem-instance dependent running times.

- Why not hybridize always ?

- When does randomization help?

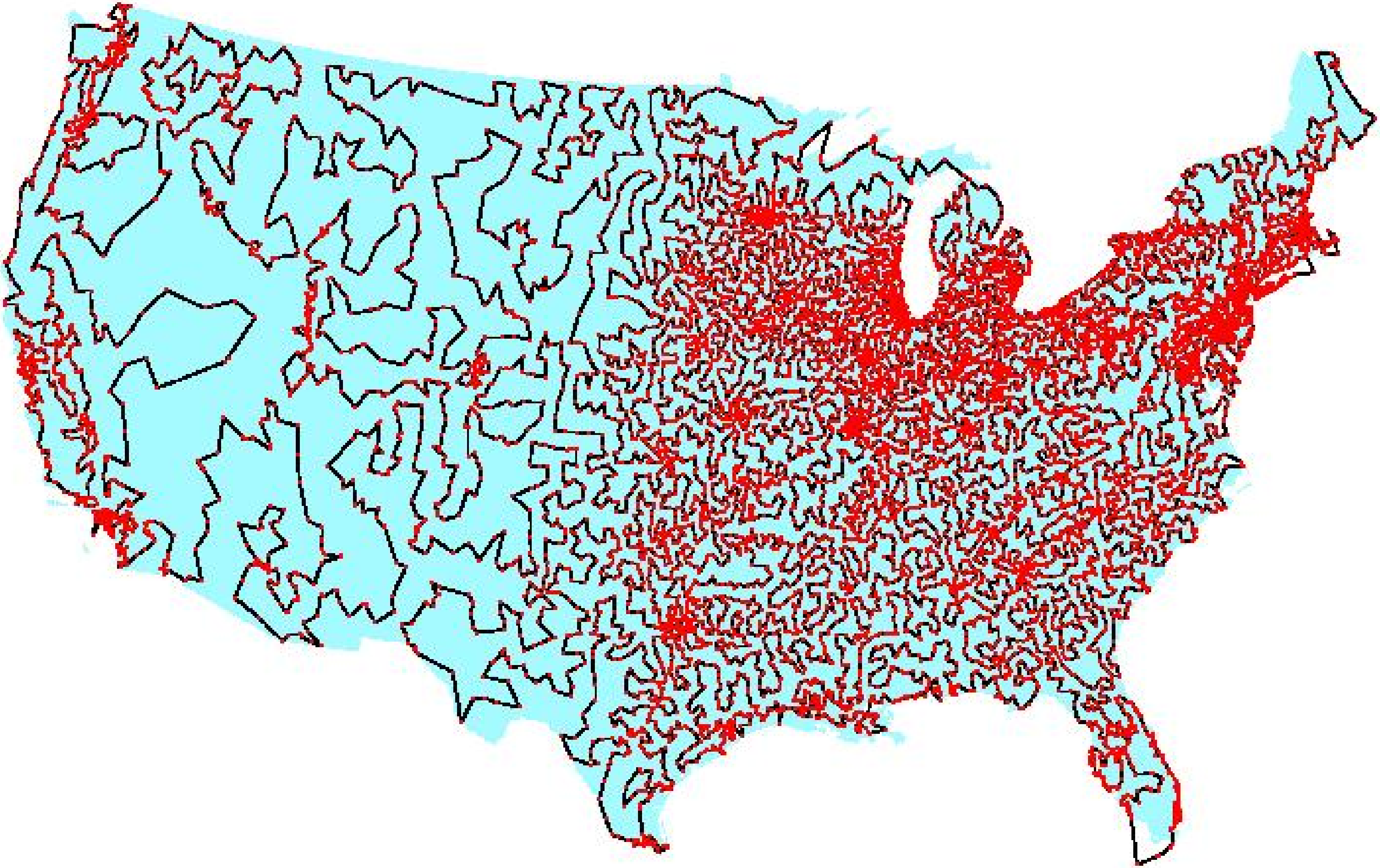
- When “does not pay” to hybridize the algorithms ?

It was not an easy start…

Three periods of MAs

- (1989-94)
  - Work in relative isolation (CALTECH, La Plata, Edinburgh, EPFL): “classical problems” TSP, QAP, Binary Perceptron, Scheduling and Coloring - TS and SA used as optimizers.
- (1994-2000)
  - Fast evolution – MAs Home Page -Timetabling – Applications in Operations Research - Use of exact techniques in complete MAs starts – Great results in TSPs (Merz & CONCORDE).
- (2000 - current)
  - Expanding number of applications and users - Cooperation with exact algorithms is better established - Links with Bioinformatics & Parameterized Complexity.

Reality check: Largest instance solved to optimality (1998): usa13509, CONCORDE group (their slide).



Tour-merging approach (CONCORDE)

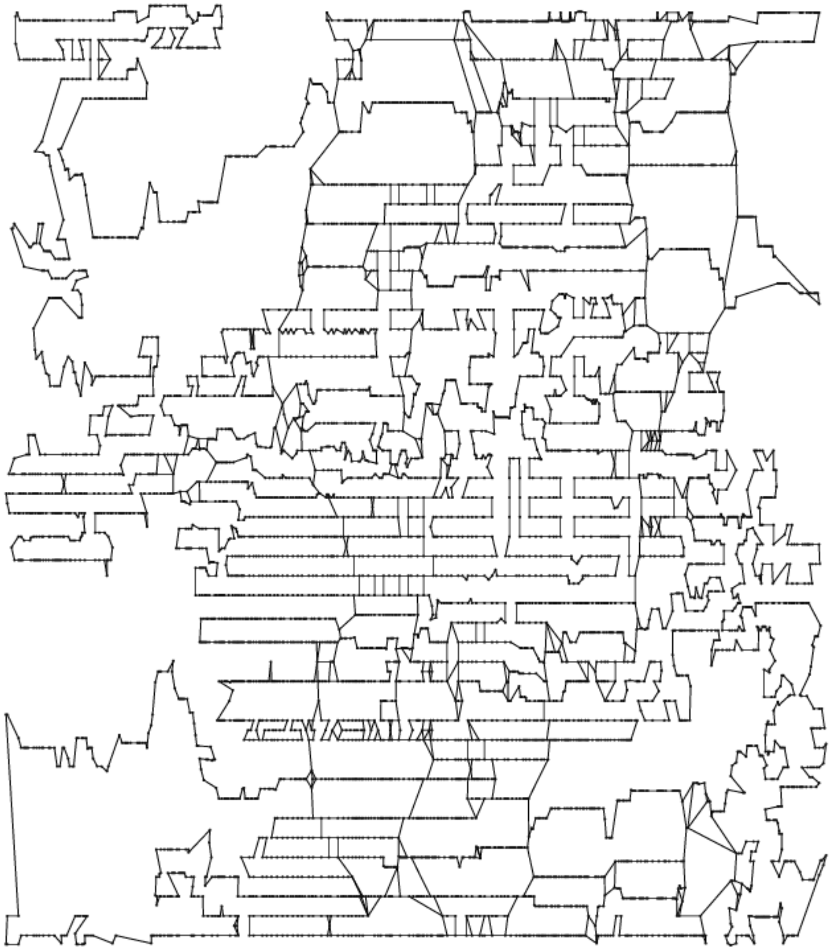
- *"Finding tours in the TSP"* paper (1999) by the CONCORDE team.
- Obtained a tour only 0.00002% more costly than optimal for usa13509.
- Great solutions by an approach that iterates Chained-LK (their basic individual optimizer) with Tour-merging (for recombination).

"it is likely that
ideas drawn from genetic algorithms
can be combined with tour merging
to produce a powerful class of heuristics."

- Of course, the approach they are using is a MAs (first proposed in 1989).
- Interest in multi-parent recombination has been rekindled (check also TSP results by Hisao Tamaki).

Union of 10 LKH tours for r15934

Figure 1 from "Tour Merging via Branch-decomposition",
W. Cook and P. Seymour, *INFORMS Journal on Computing*, 15(3):233-248, 2003.



But what if some edges (for instance the long ones in this figure) are present in all 10 parent solutions but are not in the optimal tour ?

For small *k* this could be serious.

For larger *k* is less of an issue, but the optimal recombination will take more time.

(1996-2000) Is there a systematic way to design efficient MAs ?

To reach the desired goal through three research directions

First direction:

"*Identify NP optimisation problems for which the paradigm of Evolutionary Search has been proved not competitive*

in comparison with the best exact or approximation algorithms and the best known heuristics that use other paradigms"

"Min Number Partitioning"

Weakly NP-complete

(1996-2000) Is there a systematic way to design efficient MAs ?

Second Research direction

“To identify the problems for which the Evolutionary Search strategy has proven to be a good alternative and to try to identify the reasons for the success”

Asymmetric Travelling Salesman Problem

(Strongly NP-complete)

Several applications of memetic algorithms

Class of Polynomial Merger Algorithms

(1996-2000) Is there a systematic way to design efficient MAs ?

Third research direction

"For the problems that have been identified in the second research direction, it is important to find links with the Theory of Computational Complexity, and the complexity classes (regarding approximability) to which these problems belong."

Approximability… very disappointed…

then looked at Parameterized Complexity

Running times of some approximation algorithms

Data from recent top-conferences (STOC, SODA, FOCS…)

PTAS (the jewel of the crown)

Running times for a 20% error

Euclidean TSP	$O(n^{15,000})$
Max Ind. Set. Geom. Graphs	$O(n^{523,804})$
Multiple Knapsack	$O(n^{9,375,000})$
Max Subforest	$O(n^{958,267,391})$
Gen. 4-Proc. Job. Sched.	$>$ $O(n^{10E51})$

Data from: "Some new directions and questions in Parameterized Complexity", Rod Downey and Catherine McCartin,

Lecture Notes in Computer Science 3340, pp 12-26, 2004.

The Parameterized Complexity of Multiparent Recombination

Pablo Moscato and **Carlos Cotta** (*)

Newcastle Bioinformatics Initiative

School of Electrical Engineering and Computer Science
The University of Newcastle, AUSTRALIA
Pablo.Moscato@newcastle.edu.au

(*) University of Malaga, SPAIN
ccottap@lcc.uma.es

There is a long story until we reached this talk… (more during my tutorial this afternoon)

- 1988-89, Moscato and Norman (M&N), memetic algorithms (MAs) (@CALTECH) for the TSP (traveling salesman problem).
- 1990-1, Edinburgh (@EPCC), (M and N), "Strategic Edge Crossover", far superior results than the OX crossover.
- 1991, M&N present MAs and their results to Nicholas Radcliffe (NR).
  - NR presents to M&N his work on *Forma Analysis*
  - Two natural questions arise that from that meeting:
    - *"Why is so hard to come up with good recombination methods?".*
    - *" Is this problem harder with more than a pair of parents ?"*
  - M suggests that "the problem" is "NP-complete"…
  - … and the questions sleeps for ages…

Why does an interesting research question sleep for ages ?

- A *technological* answer:
  - M&N were interested in MAs with linear speed-up in distributed heterogeneous systems and transputers. Synchronization of processes requires 2-parent recombination only (if you just think in terms of speed-up…).
- A *practitioner's* answer:
  - "Why multiparent recombination should be interesting if 2-parent recombination seems to be working well in a variety of fields ?"
- A *(bad) theoretician's* answer:
  - "Why is this problem relevant? Who is studying it?"
- A (peer-reviewed) anonymous answer:
  - "Only Moscato and Cotta are interested in these recombinations."
  - (Not true, among others, Fred Glover was pointing to these issues for particular types of Scatter Search method for many years).

Recombination can be regarded as a heuristic way to address new types of combinatorial optimization problems !!!

- THIRD (BETTER-THAN-WORST) HAMILTONIAN CYCLE

  Instance: Graph G(V,E,W), and two hamiltonian cycles C1 and C2 of G, such that C1≠C2, and w.l.o.g. Length(C1) ≤ Length(C2).

  Question: ∃ another C´, Hamiltonian cycle of G,

  such that C´≠ C1, C2; and Length(C´) < Length(C2) ?

Unknown computational complexity !!! (NP-complete ?)

This means.... Less chances of being fired !

(not only MAs help us to address NP-hard problems...

...they help us to create new ones !!)

A chance to revisit old friends

- Recombination would also help us to revisit some old combinatorial optimization problems.

- RESTRICTED HAMILTONIAN CYCLE (RHC)

  Instance: Graph G(V,E) and a Hamiltonian path P of G.

  Question: ∃ a Hamiltonian cycle in G ?

  Computational complexity known !!!

  NP-complete... (see Papadimitriou & Steiglitz, Combinatorial Optimization, Chapter 19, pp. 477-480).

  This means.... May be there is a way of reducing RHC to our previous problem and prove it NP-Complete. (???)

Parameterized Complexity

A new classification of problems is necessary. Reductions that are normally used to prove that a problem is NP-complete generally do not preserve certain structural properties. **Importance of parameters present in real-world instance of interest.**

Definition: A parameterized problem is a pair <x,k>, where x is an instance and $k > 0$ a constant, is said to be *fixed-parameter tractable* (and in class FPT) if there exists and algorithm that solves the problem in time $O(f(k)\ |x|^{\alpha})$ where $|x|$ is a measure of the size of x, and f(k) an arbitrary function of k only, and α a constant independent of k and n.

k-Vertex Cover

Example of a problem in FPT

Instance: a graph G(V,E) and an integer $k > 0$.

Question: Is there a set $V' \subseteq V$, such that for any edge $(u,v) \in E$, at least u or v is a member of V' and $|V'| \leq k$?

Decision problem is NP-complete in the strong sense…

Approximability of the optimisation problem ? During **at least** **<u>two decades</u>** the best approximation algorithm had been one with $\rho = 1$, (gap of 100 %).

Recently is has been proved that **there is no approximation algorithm with ρ < 1/6** (under the $P \neq NP$ conjecture).

Parameterized Complexity results

Fellows & Langston, 1986: $O(f(k)\ n^3)$.

Johnson, '87: $O(f(k)\ n^2)$.

Fellows, '88: $O(2^k\ n)$.

Buss, '89: $O(kn + 2^k k^{2k+2})$.

Balasubramanian *et al.*, '92: $O(kn + 2^k k^2)$.

Papadimitriou & Yannakakis, '93: $O(3^k\ n)$.

Balasubramanian *et al.* '98: $O(kn + 1.32472^k k^2)$.

Downey, Fellows, & Stege, '99: $O(kn + 1.31951^k k^2)$.

Niedermeier and Rossmanith, '99: $O(kn + 1.29175^k k^2)$.

Stege and Fellows, '99: $O(kn + \max\{1.25542^k k^2, 1.2906^k k\})$.

Niedermeier and Rossmanith, '00: $O(kn + 1.2906^k)$.

Chen, Kanj, & Jia, '99: $O(kn + 1.286^k)$.

Chen, Kanj and Xia, '05: $O(kn + 1.2738^k)$.

Evolving L-Systems as an intelligent design approach to find classes of difficult-to-solve Traveling Salesman Problem instances

Farhan Ahammed and Pablo Moscato

In Applications of Evolutionary Computation
Lecture Notes in Computer Science Volume 6624, 2011, pp 1-11

Introduction

- Suppose we have:
  - A program (that implements a particular algorithm)
  - An instance that the algorithm solves "really fast".

- Question: *Can we modify this instance slightly so that it becomes "difficult-to-solve" for the given program?*
  - The size of the instance does not change
  - Structure is somewhat "preserved" – we're not creating a completely new/different instance

Introduction Cont’d

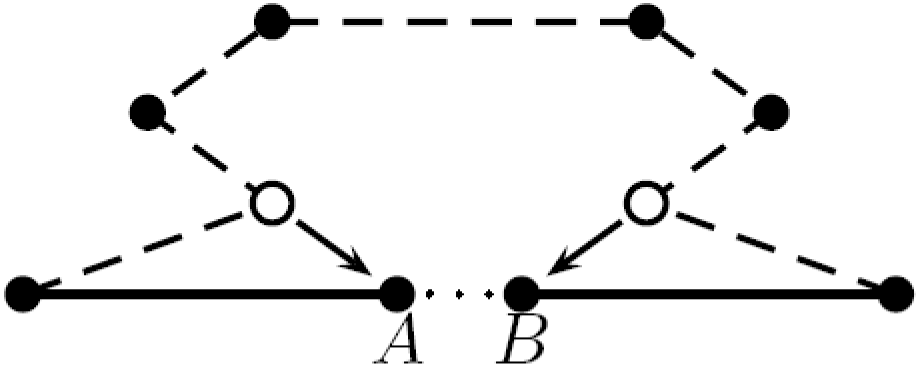
- Build an *instance generator* to create many (easy-to-solve) instances

- Use the instance generator to create similar instances that are now much more difficult to solve.

- Success with evolutionary algorithms.

On the Performance of Heuristics
on Finite and Infinite Fractal Instances of the Euclidean Traveling Salesman Problem,
Pablo Moscato, Michael G. Norman, 10(2), pp. 121–132, Published Online: May 1, 1998.

- Idea: A TSP instance can be made more difficult for an exact algorithm if some cities are "moved" (or *perturbed*) a small distance



- Cities *A* and *B* have been moved

- An exact algorithm might now consider the new inferior option (connecting A to B)

Testing our idea

- *Concorde (best exact algorithm in 2007)*
- Instance generators based on iterated-function systems and L-systems
  - For each instance generated, certain cities are chosen to be *perturbed* slightly
- A local search technique is used to find new modified fractals that produce difficult-to-solve instances

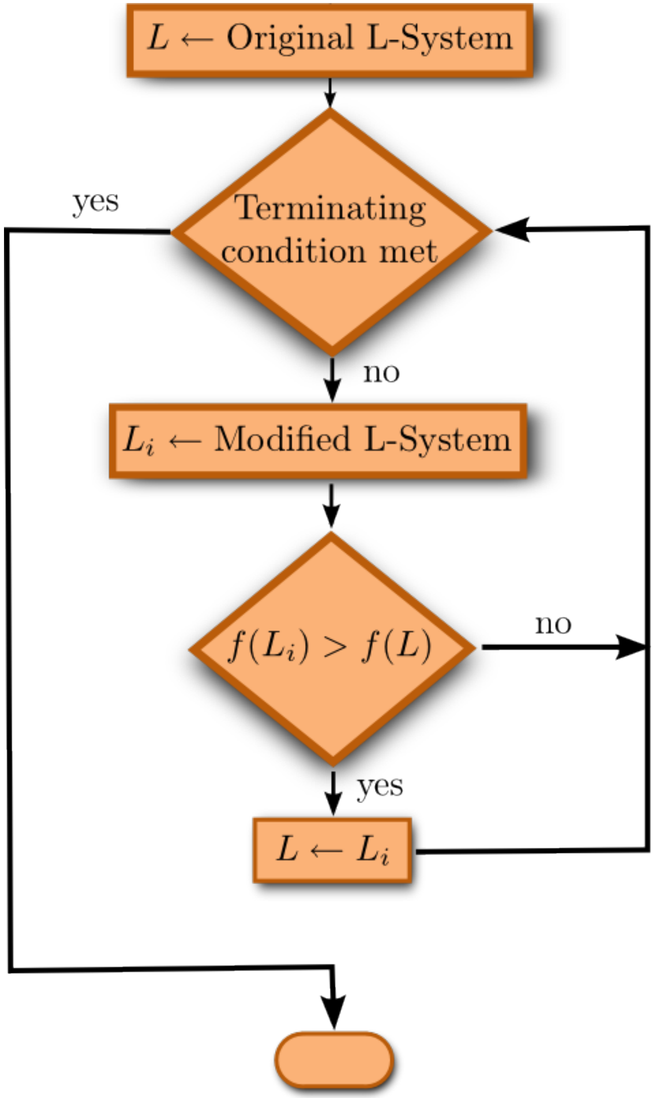
Our contribution

- A framework for finding new instance generators

- Input:
  - A program
  - An instance generator
  - A method of modifying an instance generator

- Output:
  - A new instance generator, which creates instances of the same size and similar structure to the original, but more difficult to solve for the given program

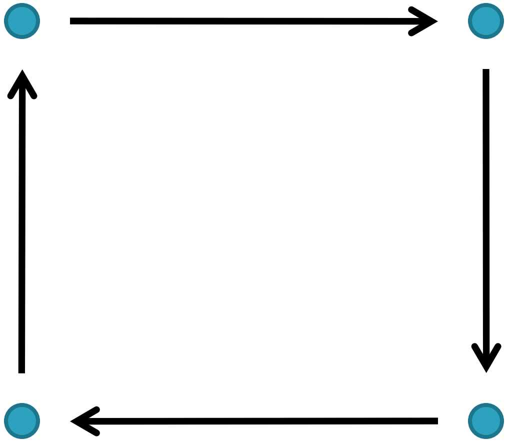
Finding new instance generators


$L \leftarrow$ Original L-System
yes
Terminating condition met
no
$L_i \leftarrow$ Modified L-System
$f(L_i) > f(L)$
no
yes
$L \leftarrow L_i$

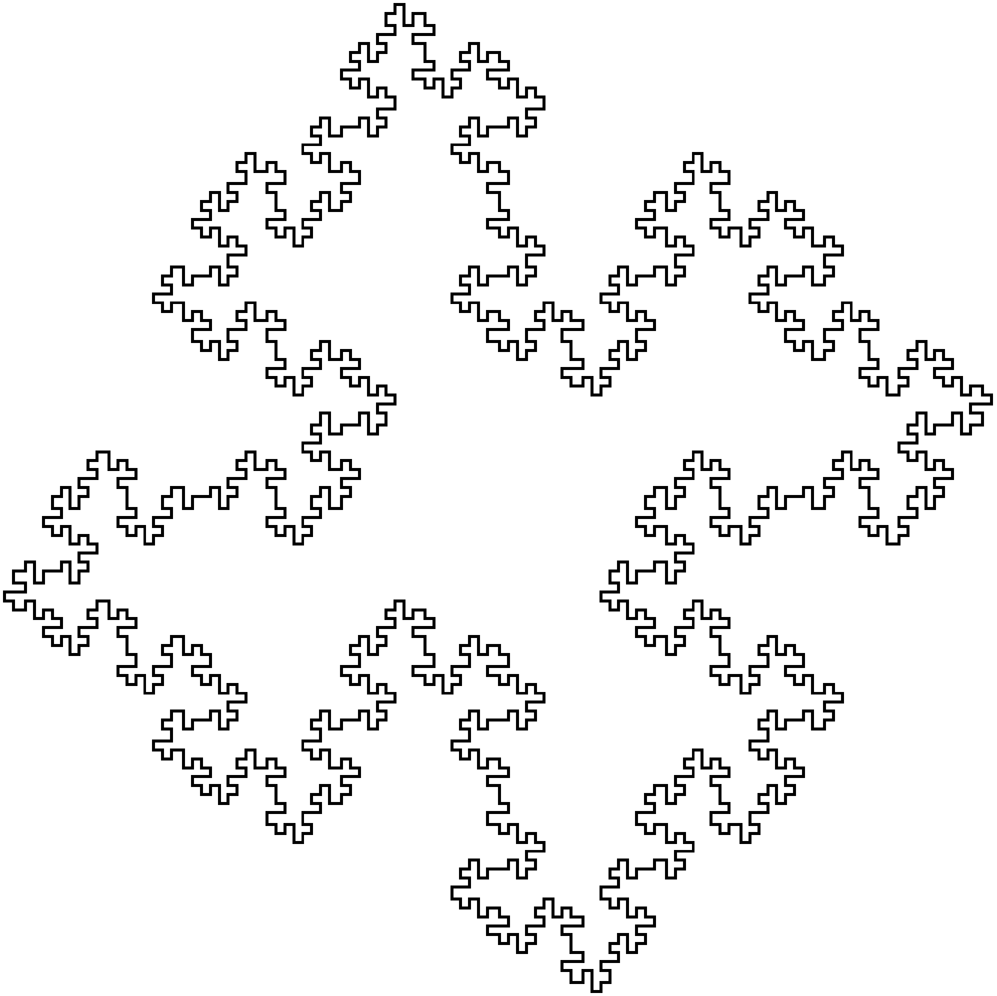
Lindenmayer Systems (L-Systems)

- Grammar-based method of describing

- Example: F\90 F\90 F\90 F



Quadratic Koch Island



Quadratic Koch island

Order n = 3

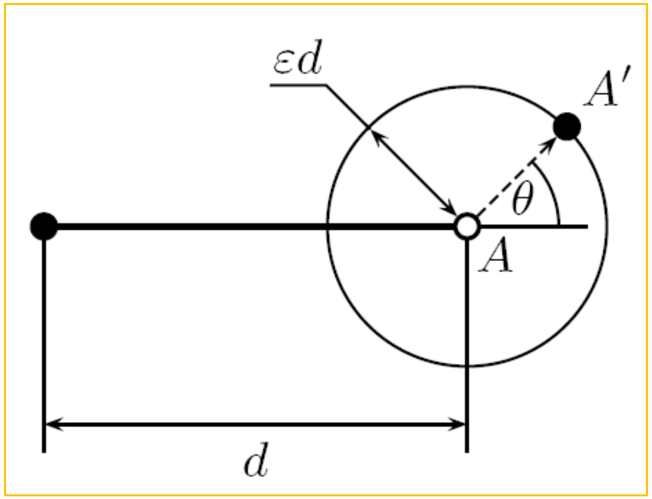
Axiom = F+F+F+F

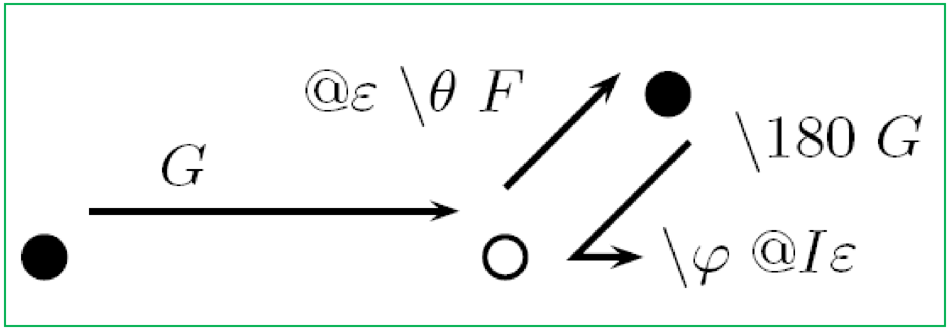
Rule = F → F+F-F-FF+F+F-F

$\delta = 90°$

http://greengene.republika.pl

Perturbation of a city position





$$F \leftarrow G @ \varepsilon \backslash \theta \, F \backslash 180 \, G \backslash \varphi @ I \varepsilon$$

Comparing L-Systems

- Each L-System can generate multiple TSP instances of various sizes
- Given two L-Systems L and L', we can generate instances of sizes $\{i : i=s_1, s_2,\ldots,s_n\}$
- Compute the running times required by *Concorde*
  - *Average:* t_i, t_i'
  - *Std Dev:* σ_i, σ_i'
- The fitness of L', relative to L is:

$$f_L(L') = \sum_i i\left[\left(t_i' - 2\sigma_i'\right) - \left(t_i - 2\sigma_i\right)\right]$$

Evolutionary "attack" of algorithms

- Moscato and Norman ('98) showed that Peano/Hilbert-like iterative process that can be used to generate arbitrary large TSP instances with known optimal solutions (to use as testbeds).

- Cotta and Moscato ('03)* proposed an evolutionary computation-based attack on algorithms
  - Sorting algorithms: bubblesort, quicksort, shellsort, etc.
  - Lower bounds for the worst-case scenarios

* *Applied Mathematics Letters*, 16:41–47, 2003.

Evolutionary "attack" of algorithms

- Langdon and Poli ('05) used genetic programming to find fitness landscapes which highlight strengths and weaknesses of different Particle Swarm Optimisation algorithms

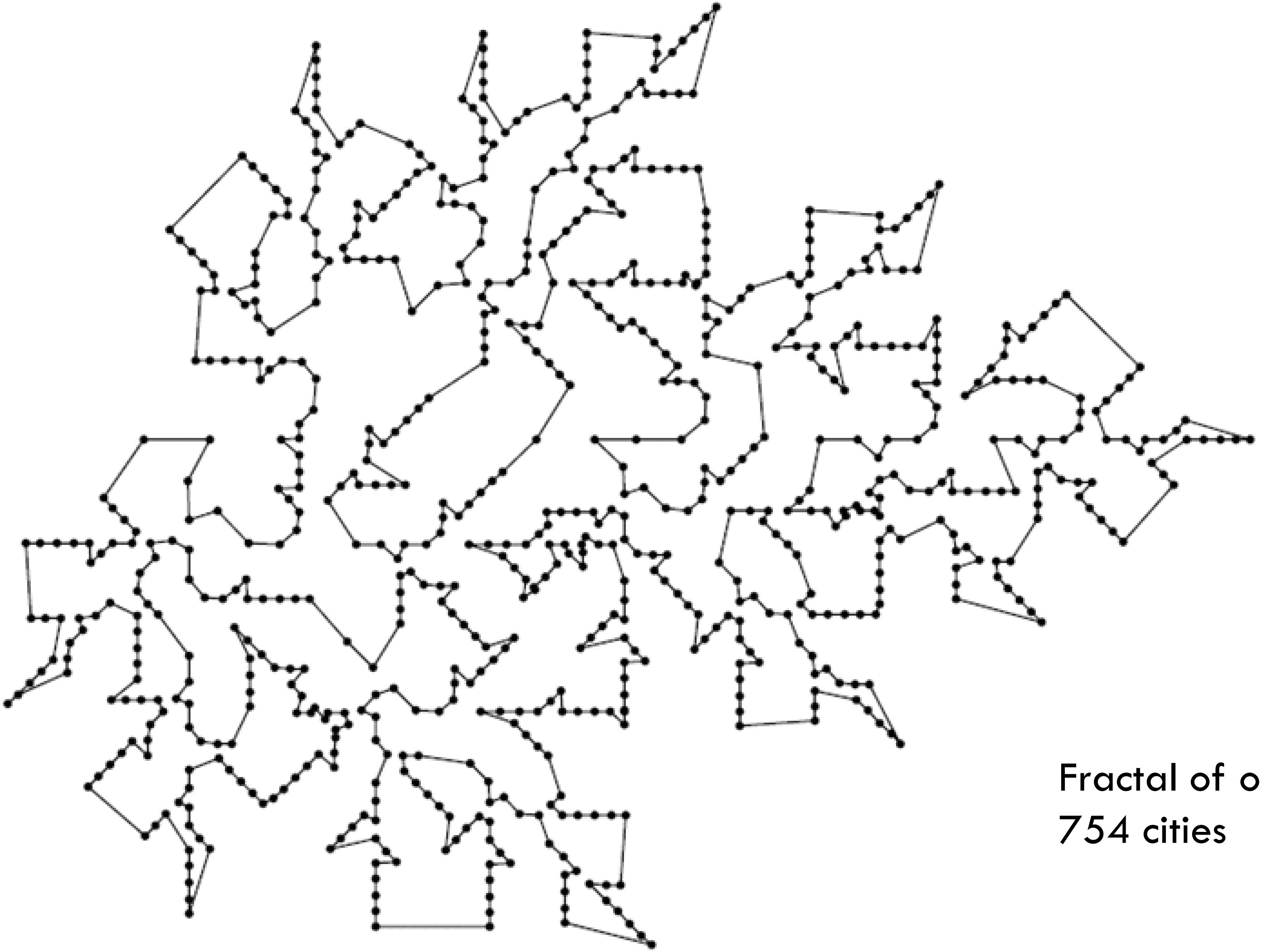
- van Hermert ('06) used EAs to find weaknesses in combinatorial optimisation algorithms
  - Binary constraint satisfaction, boolean satisfiability and traveling salesperson problems
  - Shared patterns among "difficult-to-solve" instances

Evolutionary “attack” of algorithms

- Most work is concerned with finding instances which represent the worst-cases scenarios

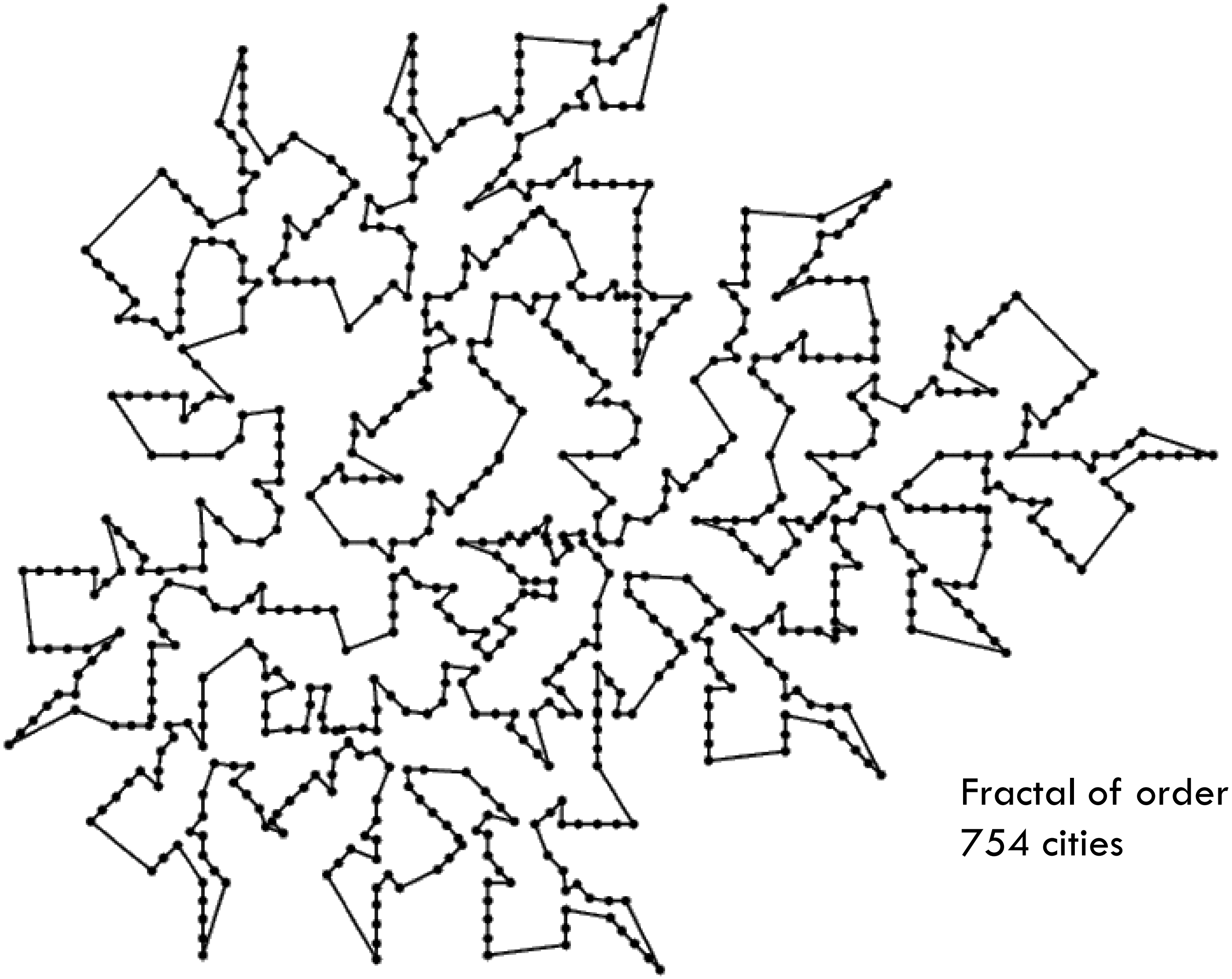
- The situation we’re looking at:
  - Make minor changes to easy-to-solve instances and turn them into difficult-to-solve instances OF THE SAME SIZE
  - Use L-Systems to generate instances that Concorde can solve (quickly)
  - Modify the instances such that the new instances are slightly different, but still have similar “local” characteristics

Leaf2 – Original version



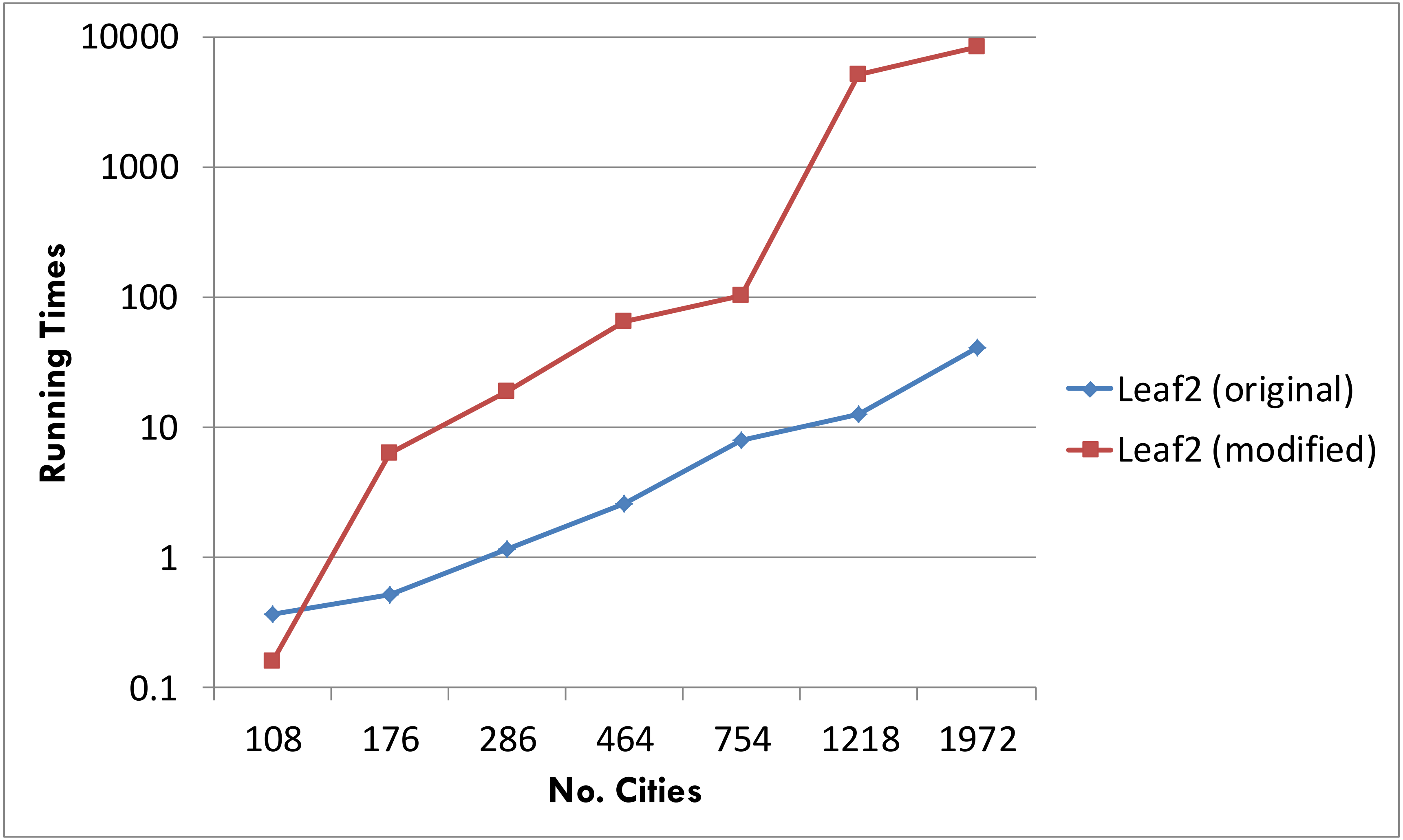
Fractal of order 12
754 cities

Ev-Leaf2 – Evolved instance

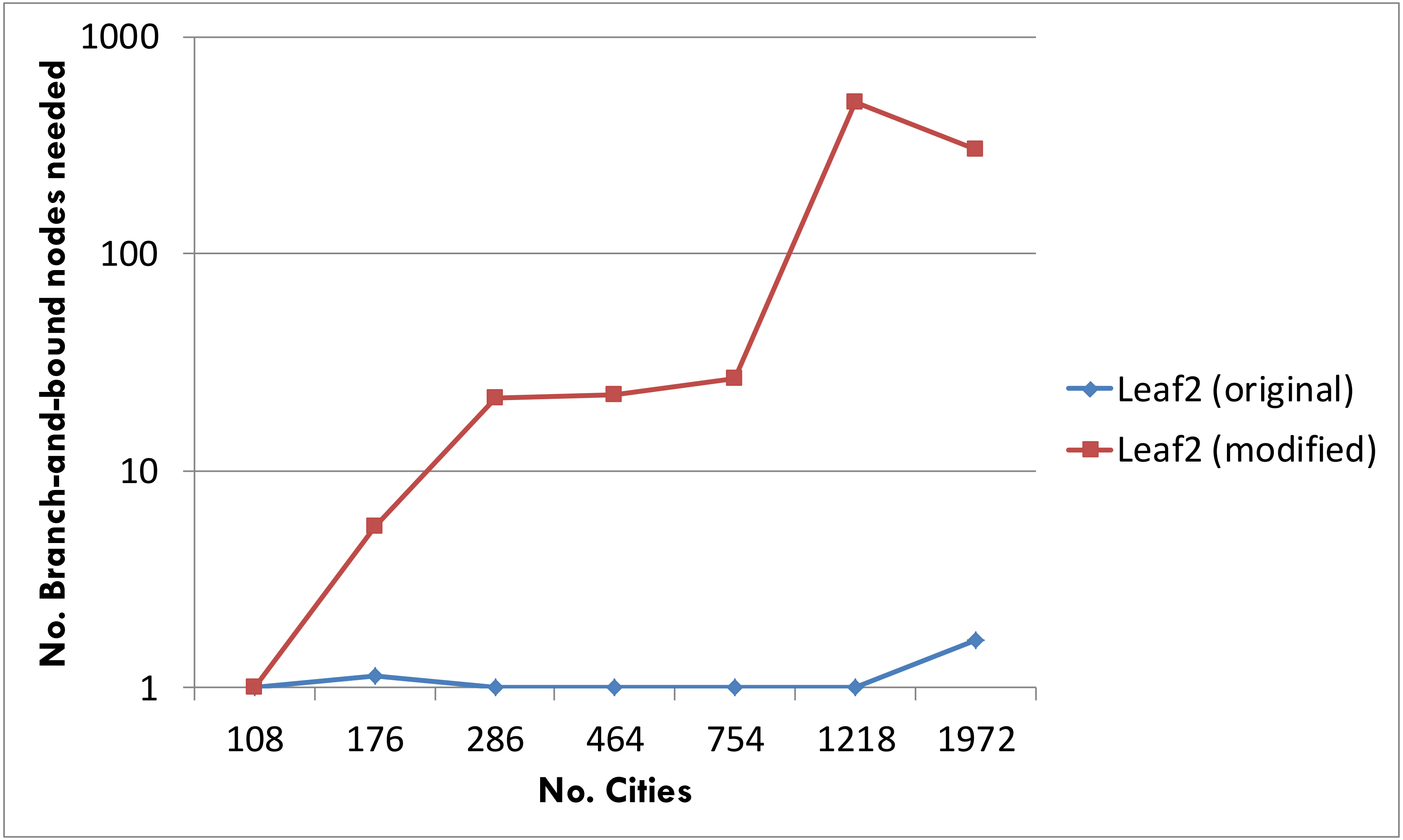


Fractal of order 12
754 cities

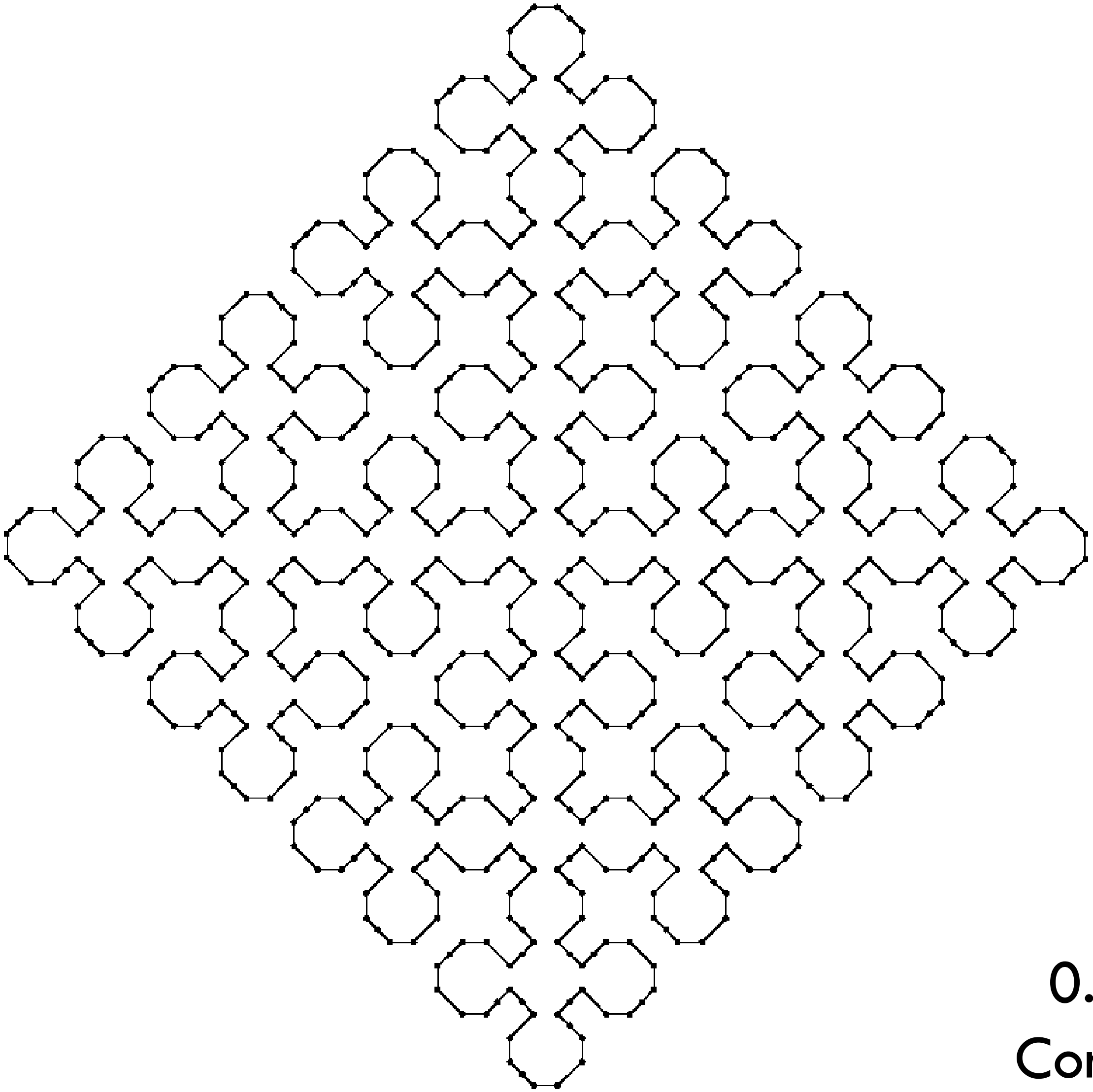
Comparison of running times



Branch-and-bound nodes comparison

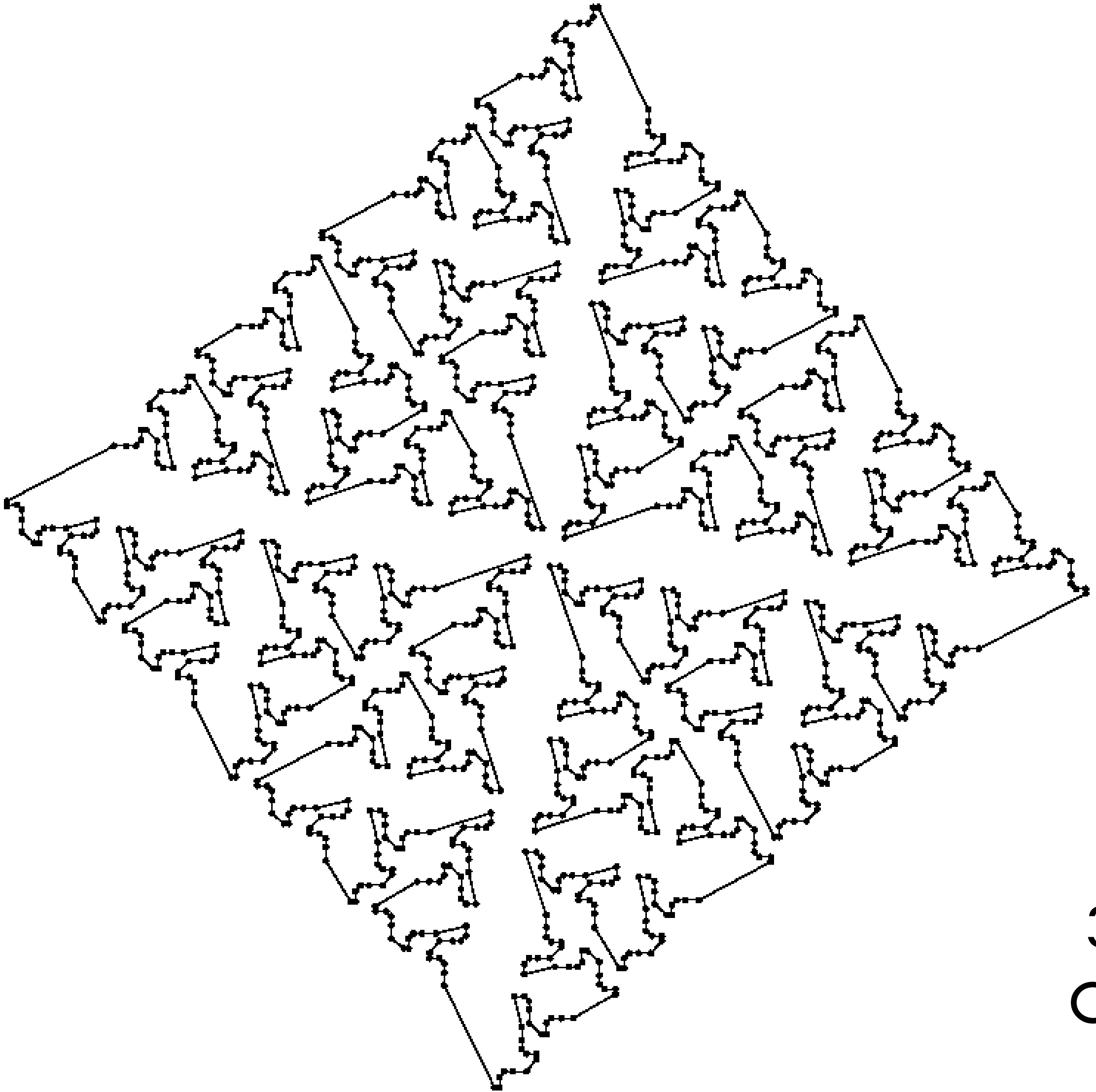


MNPeano – original design



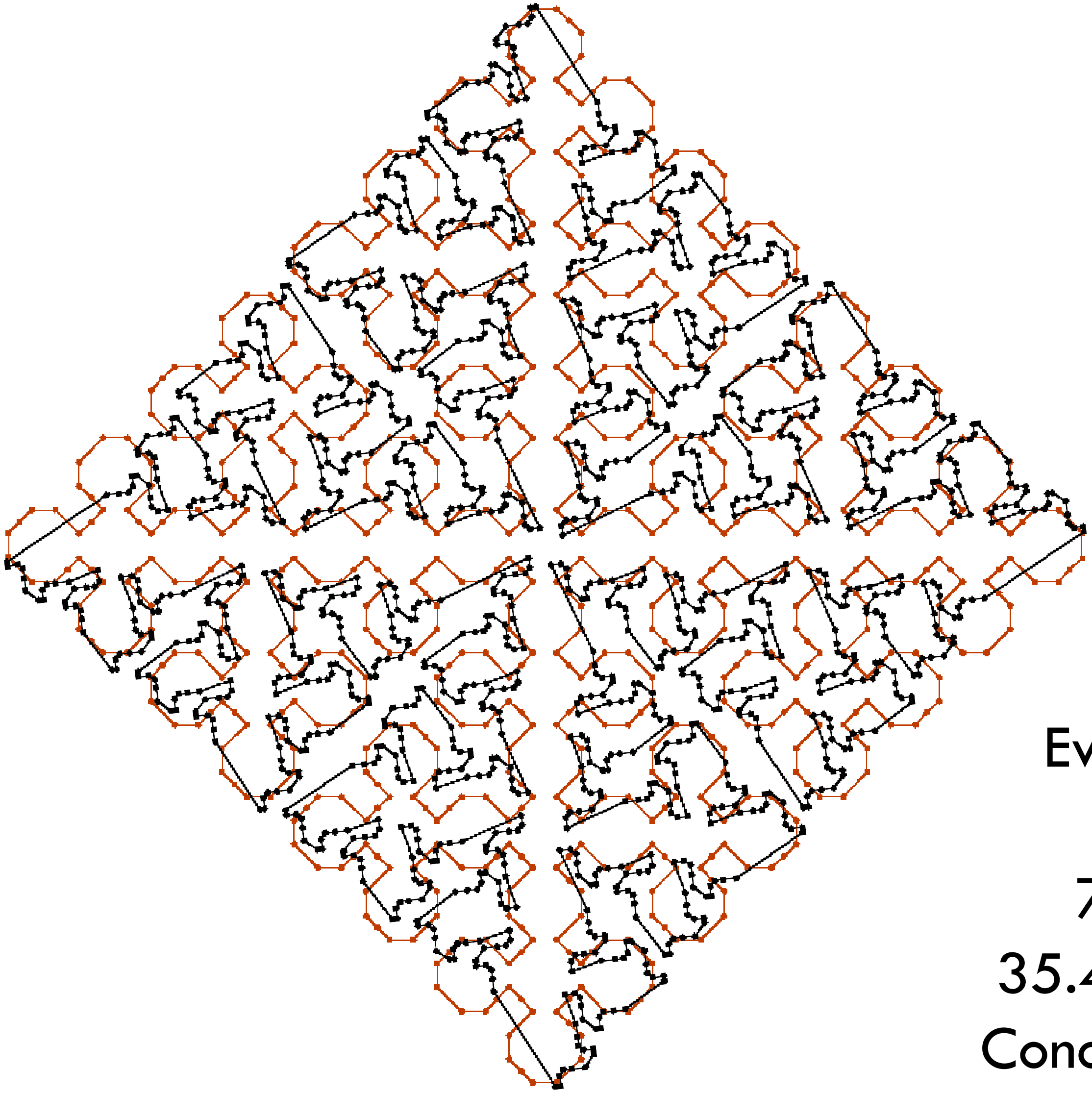
MNPeano
order 7,
724 cities,
0.2 seconds for
Concorde to solve

Ev-MNPeano – Evolved design



Ev-MNPeano
order 7
724 cities,
35.4 seconds for
Concorde to solve

E-MNPeano (superimposed)



Ev-MNPeano
order 7
724 cities,
35.4 seconds for
Concorde to solve

Leaf2 results

Order	Cities	Time(s)		No. BB Nodes	
		Leaf2	Ev-Leaf2	Leaf2	Ev-Leaf2
8	108	0.366	0.16	1.000	1.000
9	176	0.525	6.4	1.133	5.533
10	286	1.152	18.6	1.000	21.800
11	464	2.564	65.8	1.000	22.467
12	754	7.901	103.2	1.000	26.467
13	1218	12.61	5196.3	1.000	497.333
14	1972	41.03	8442.55	1.667	303.245

MNPeano results

Order	Cities	Time(s)		No. BB Nodes	
		MNP	Ev-MNP	MNP	Ev-MNP
5	180	0.045	0.696	1	1
6	364	0.094	21.311	1	4
7	724	0.242	35.380	1	1
8	**1452**	**0.518**	**15278.5**	**1**	**177**
9	2900	1.264	10916.55	1	18

Partial Summary

- Grammars were used to generate instances of a combinatorial optimisation problem on which an algorithm performs “increasingly badly” (proven exponential time?)

- Instances share the same structure design patterns

- A local search optimisation technique was used to find a modification
  - When applied to the instances, more difficult-to-solve instances are found

Partial Summary (cont)

- The results show large increases in time and decisions needed (B&B nodes) when solving the modified instances
  - Same sizes and similar structures to original instances

- This approach can help with investigating the worst-case scenarios of algorithms for which a theoretical analysis is difficult to perform

Final points for discussion

- “P≠NP?” the question is not helpful to develop useful practical solutions

- Theory of Computer Science “does not scale” (15,000 NP-complete problems and counting…). Approximability failed to deliver over four decades!

- The approximability of an NP optimisation problem appears to be uncorrelated with the “hardness” of being solved in practice with evolutionary programming techniques. Parameterized complexity appears to be a better paradigm to study the complexity of recombination problems arising in memetic algorithms (Concorde is a good example)

- The “FPT Toolkit” of Downey & Fellows can be sometimes reused to solve to optimality the “dynastically optimal” problem in recombination and they can be useful in practice.

… And more conclusions…

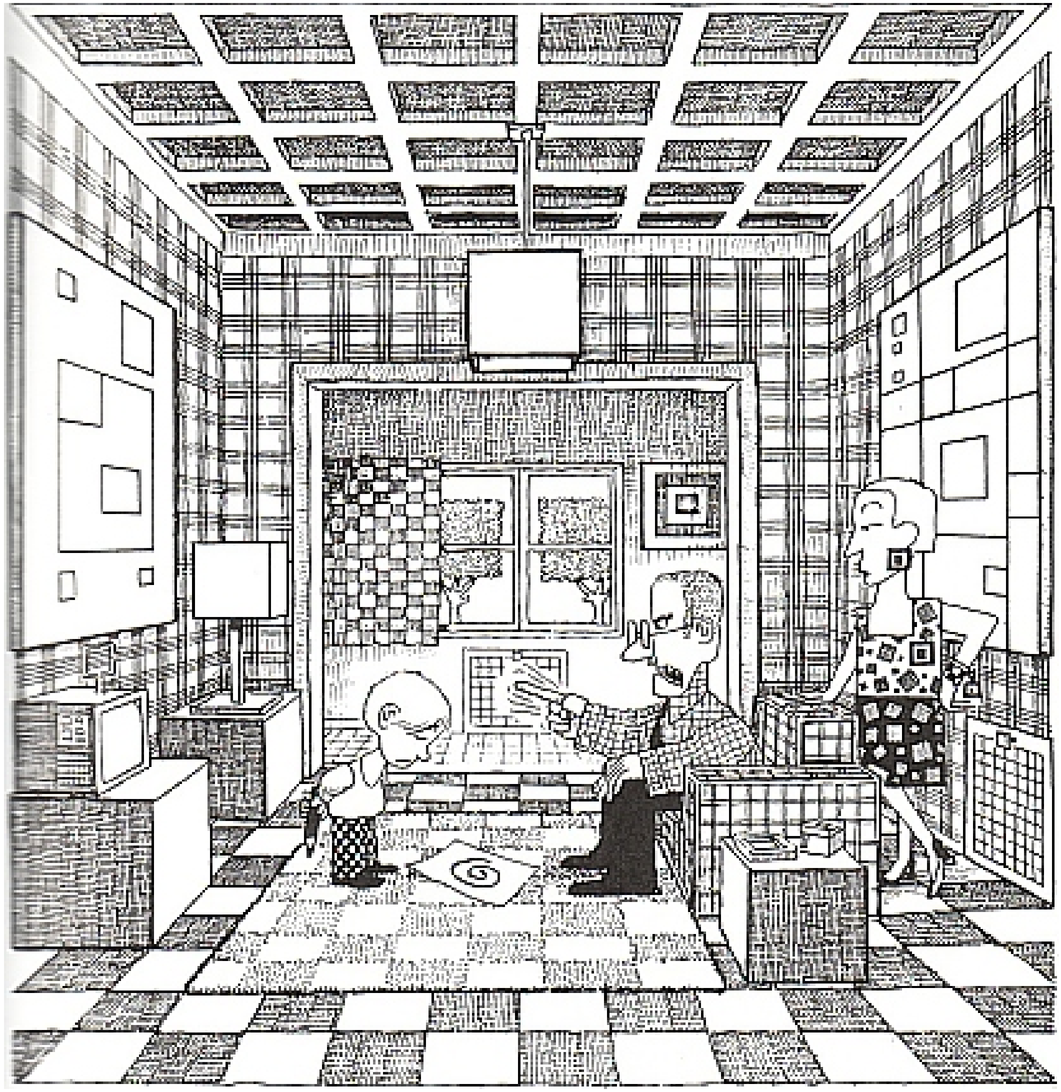
- Possibility of establishing an **algorithmic design framework of recombination problems**

- **"Hard Puzzles Conjecture" (such that HPC implies** P≠NP) (Fellows and Rosamond, in In Computation and Logic in the Real World: Third Conference on Computability in Europe, CiE 2007).

  "there exist intrinsically hard instances of NP-hard problems
  (e.g., GRAPH 3-COLORING)
  that uniformly defeat all algorithms,
  whose only purpose is to efficiently (i.e. in time O(*n*^c))
  find a 3-coloring of a single instance of size *n*,
  when the algorithms are uniformly bounded in size by c."

- **A man-machine approach** for creating these instances, will inform the field on the key obstructions to develop exact algorithm and practical solutions for important problems.

Remember, this is just a personal view



Thank you for listening !

Conclusions obtained from previous work…

- Hybridization in MAs should continue due to the good results in several application areas

- Complete memetic algorithms -> good research direction.

- The approximability of an NP optimisation problem appears to be uncorrelated with the "hardness" of being solved in practice with evolutionary programming techniques.

- Parameterized complexity appears to be a better paradigm to study the complexity of recombination problems arising in memetic algorithms.

- The "FPT Toolkit" of Downey & Fellows can be sometimes reused to solve to optimality the "dynastically optimal" problem in recombination and they can be useful in practice.

… And more conclusions…

- Possibility of establishing a **solid computational complexity theory of recombination problems**

- Classes uPMA and PMA.

- Computational complexity of **multi-parent recombination** (… links to **Genetic Engineering** issues ?).

- Benefits of using methods from Modal Logic and **Multi-Agent Belief Logic** for the introduction of problem (and instance) dependent knowledge.

- Possibility of using the generated "on-line" knowledge to guide exact and GRASP-like algorithms by taking advantage of the **learning processes** in each agent.

- Use of **Logic Programming and belief update methods** to update the population's shared knowledge..

Next steps

- FPT algorithms and Parameterized Complexity
  - As a “tool”
    - At the recombination step: use of exact algorithms to solve subproblems in the metaheuristic.
    - **No need to use randomization when a good exact algorithm can optimally solve a subproblem of interest.**

- Complete MAs
  - Hybridization with mathematical programming commercial solvers
  - Hybridization with logic programming and constraint programming solvers
  - Challenges, challenges, challenges !!!!

Future issues

- MAs with multiple representations
- Frameworks for MAs to exploit code reuse
- MemePool Project
- Polynomial Merger Algorithms
- Problems outside NP
  - Linear Programming ?
  - Maximum Cardinality Matching ?
  - Problems in PSPACE ?
  - GAs vs MAs
- Complete Memetic Algorithms

Next steps

- Role of FPT algorithms and Parameterized Complexity in MA research
  - As a "tool"
    - At the recombination step: use of exact algorithms to solve subproblems in the metaheuristic.
    - **No need to use randomization when a good exact algorithm can optimally solve a subproblem of interest.**
- Complete MAs
  - Hybridization with mathematical programming commercial solvers
  - Hybridization with logic programming and constraint programming solvers
  - Challenges, challenges, challenges !!!

Next steps

- FPT algorithms and Parameterized Complexity
  - As a "tool"
    - At the recombination step: use of exact algorithms to solve subproblems in the metaheuristic.
    - **No need to use randomization when a good exact algorithm can optimally solve a subproblem of interest.**

- Complete MAs
  - Hybridization with mathematical programming commercial solvers
  - Hybridization with logic programming and constraint programming solvers
  - Challenges, challenges, challenges !!!!

Final Conclusions

- Hybridization in MAs should continue due to the good results in several application areas

- Complete memetic algorithms -> good research direction.

- The approximability of an NP optimisation problem appears to be uncorrelated with the "hardness" of being solved in practice with evolutionary programming techniques.

- Parameterized complexity appears to be a better paradigm to study the complexity of recombination problems arising in memetic algorithms.

- The "FPT Toolkit" of Downey & Fellows can be sometimes reused to solve to optimality the "dynastically optimal" problem in recombination and they can be useful in practice.

… And more conclusions…

- Possibility of establishing an **algorithmic design framework of recombination problems**

- **"Hard Puzzles Conjecture" (such that HPC implies** P≠NP) (Fellows and Rosamond, in In Computation and Logic in the Real World: Third Conference on Computability in Europe, CiE 2007).

  "there exist intrinsically hard instances of NP-hard problems
  (e.g., GRAPH 3-COLORING)
  that uniformly defeat all algorithms,
  whose only purpose is to efficiently (i.e. in time O(n^c))
  find a 3-coloring of a single instance of size n,
  when the algorithms are uniformly bounded in size by c."

- **A man-machine meta-evolutionary memetic approach for creating these instances,** will inform the field on the key obstructions to develop exact algorithm and practical solutions for important problems.