Towards the Automatic Design of More Efficient Digital Circuits

Vesselin K. Vassilev

South Bank University

London

Dominic Job

Napier University

Edinburgh

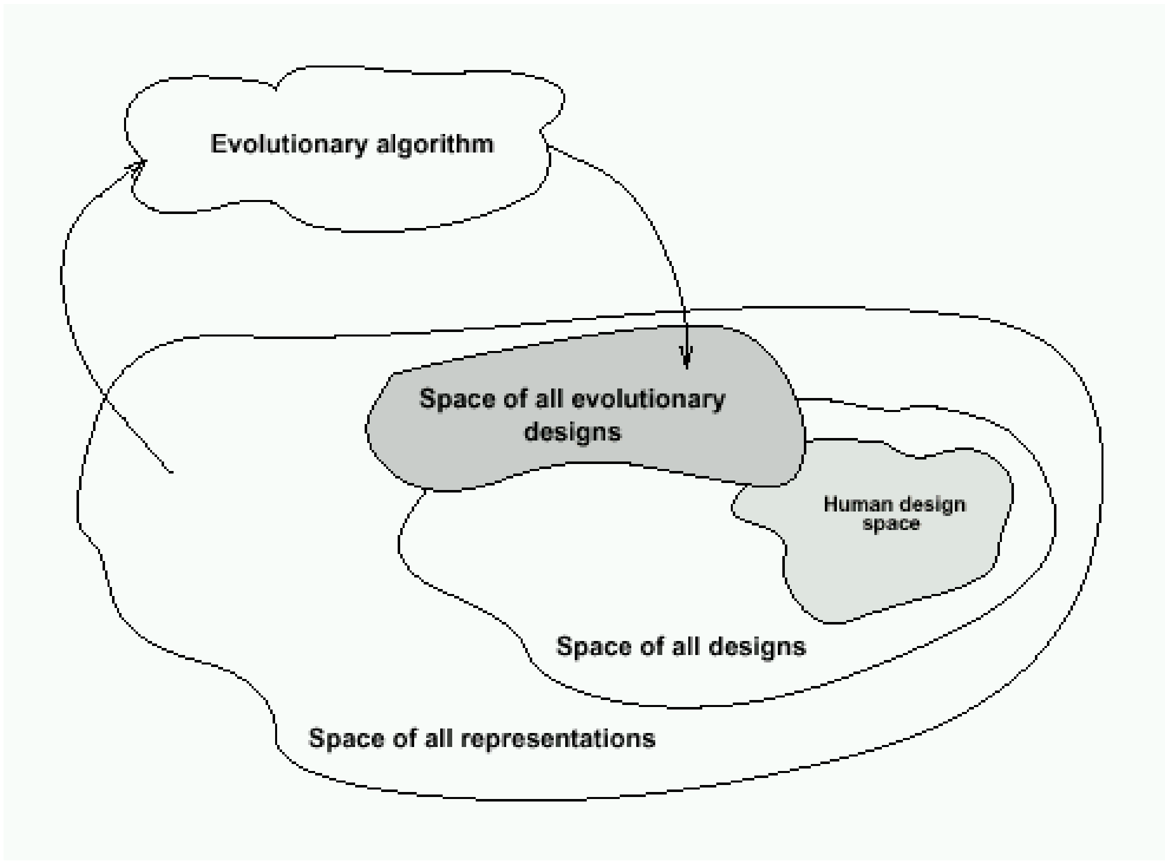
Julian F. Miller

The University of Birmingham

Birmingham

July 4, 2000

Evolving digital circuits ...

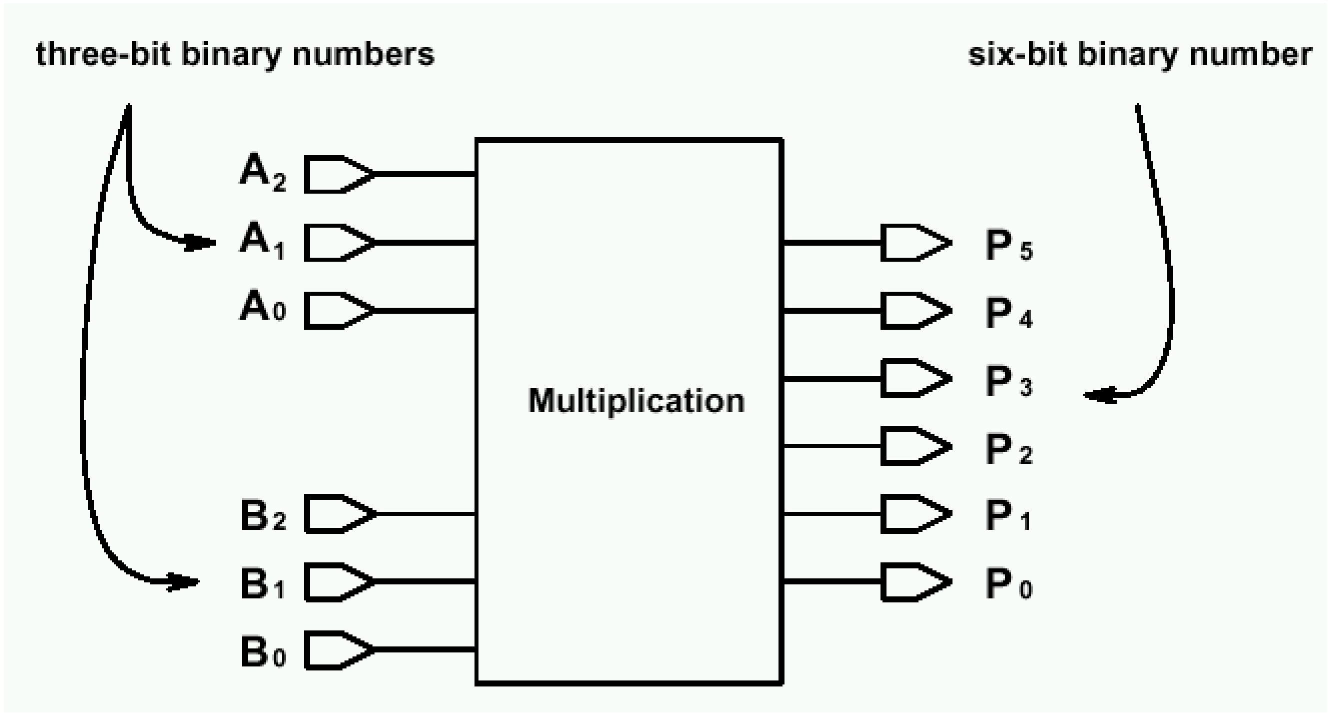

Evolutionary algorithm
Space of all evolutionary designs
Human design space
Space of all designs
Space of all representations

- ... does not necessarily mean automatic design, because evolution may not produce a functionally correct circuit!

n * m -Bit Binary Multiplier

- Implements multiplication of two binary numbers of n and m bits.

Three Three-Bit Binary Multiplier



- ... for instance, the three three-bit multiplier (also known as the three-bit multiplier) implements binary multiplication of two three-bit numbers to produce a possible six-bit number.

… evolving Binary Multipliers is an interesting problem

- The multiplier is a fundamental building block.
- There is a well established conventional methodology for design of binary multipliers.
- There are no conventional methods for building multiplication, using other gates apart from AND, OR, NOT, and XOR.

Circuit evolution can be considered as a search on a fitness landscape

Features of Fitness Landscapes

- A fitness landscape is uniquely characterized by its:
  - smoothness,
  - ruggedness,
  - and neutrality.

- Ruggedness
  - Number and distribution of local optima
- Smoothness
  - Size of the basins of attraction
- Neutrality
  - Neighboring configurations with equal fitnesses

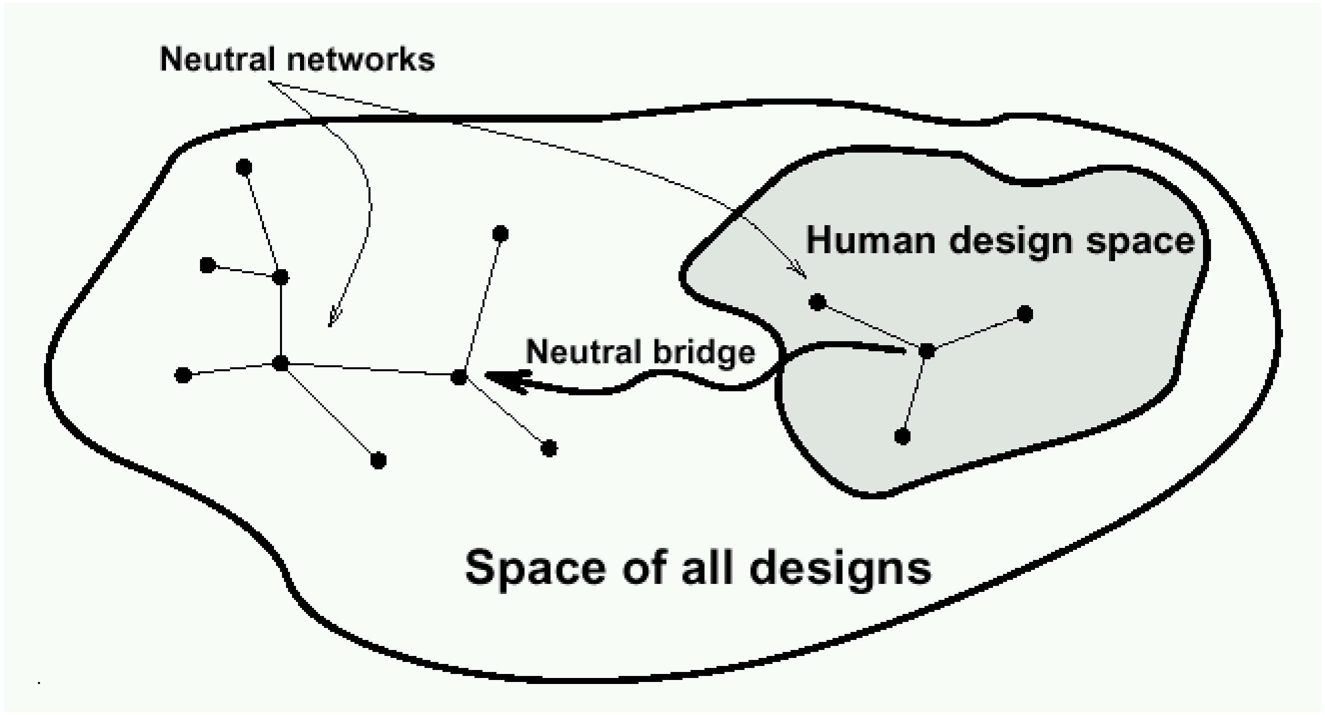
... these define the structure of landscapes

What do we know about landscape neutrality and evolutionary search?

- 1) A connected subgraph of genotypes with equal fitness values is referred to as neutral network.
- 2) Landscapes neutrality: neutral networks or neutral clusters?
- 3) The neutral networks allow the population to cross large landscape regions with lower fitness, and thus, to investigate the space of all designs!!!

... hmmm!?! If this is true ...

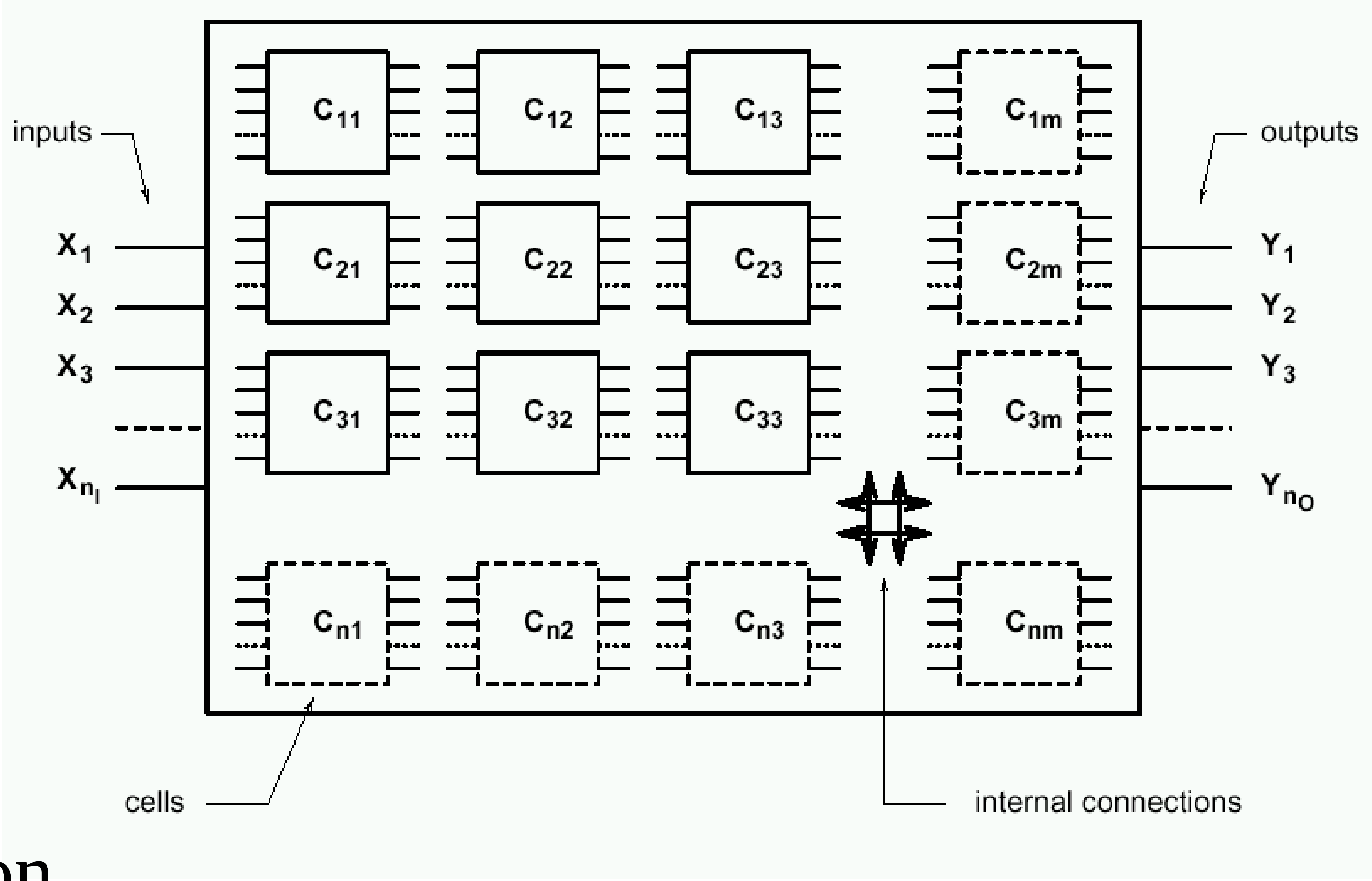
- Why don't we define a neutral network that will connect “all" functionally correct designs of an arithmetic (or any other) function?



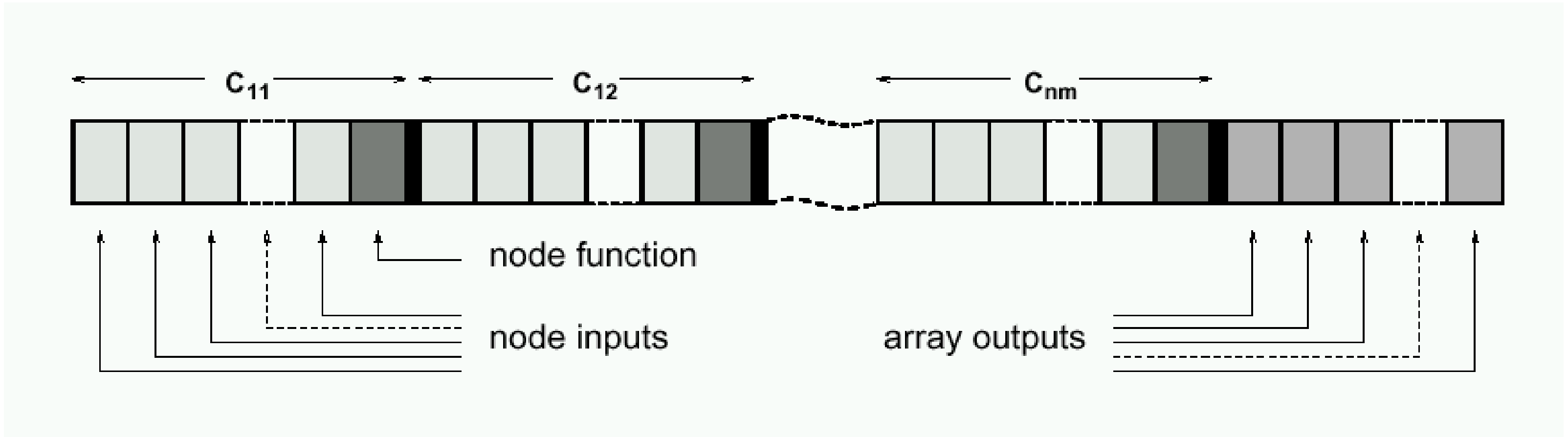
Cartesian Genetic Programming

- Evolution of programs represented by rectangular arrays (graphs) rather than trees.
- This is done via genotype- phenotype mapping that allows landscape neutrality.
- ***An array of cells***

Array of cells

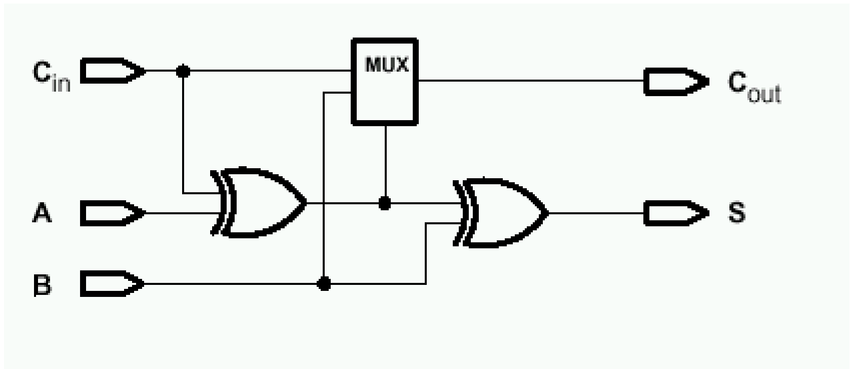


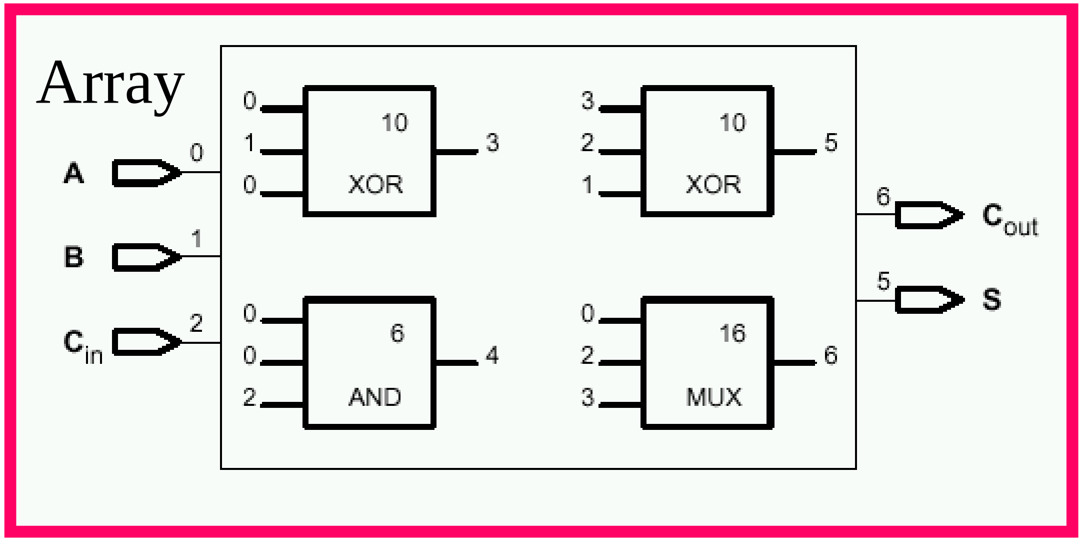
representation



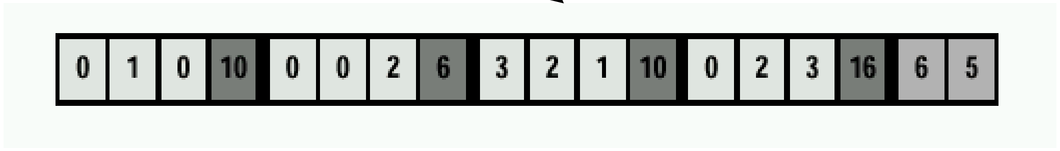
Genotype-Phenotype Mapping (example)

Phenotype





Genotype



... the mapping introduces five sources of landscape neutrality

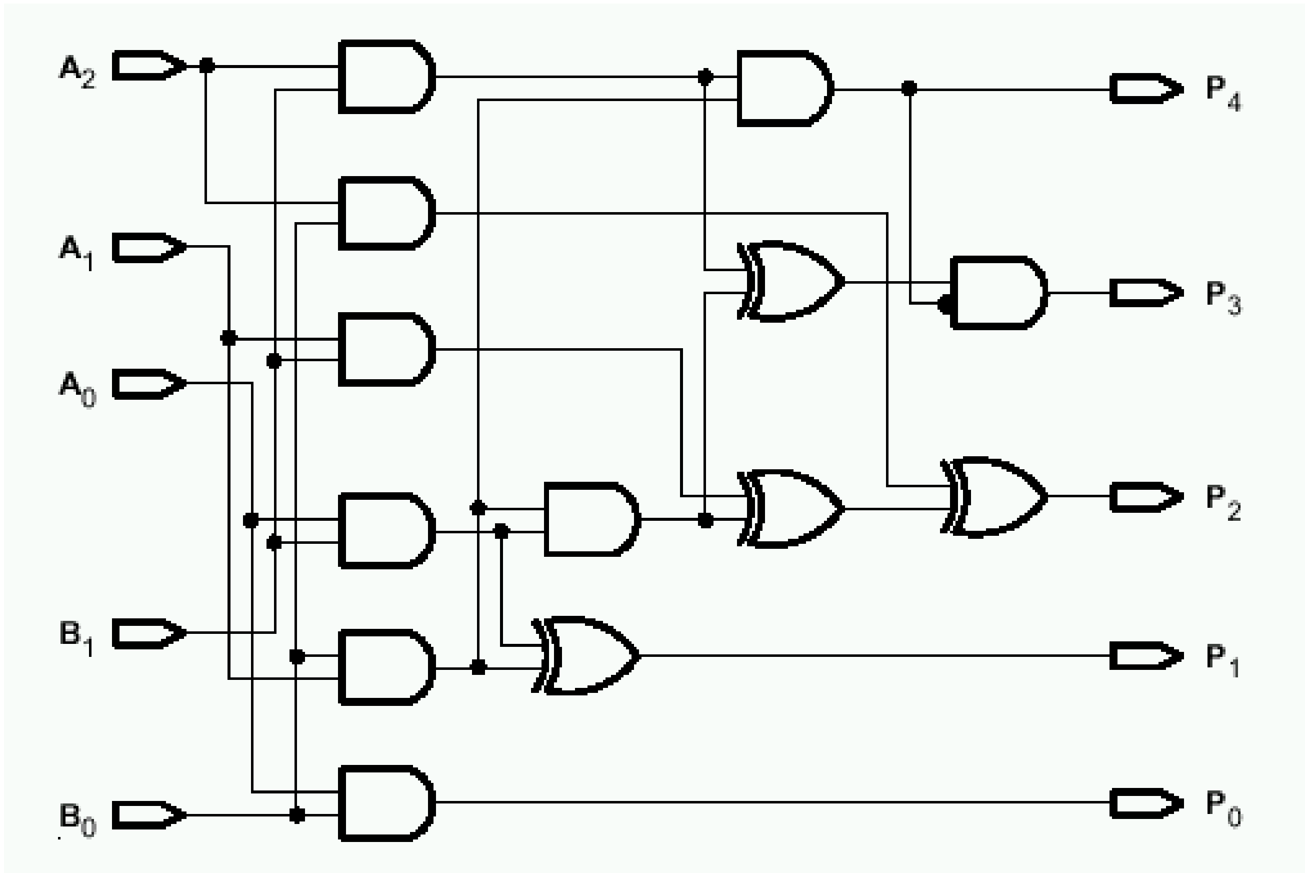
- 1) **Input redundancy**
  - Inputs of cells that are not used in the operating circuit.
- 2) **Cell redundancy**
  - Cells whose outputs are not connected in the operating circuits.
- 3) **Functional redundancy**
  - The case in which the number of cells of a digital circuit is higher than the optimal number needed to implement the circuit.
- 4) **Logic equivalency**
  - The case in which a (sub-)circuit can be substituted with another “logically" equivalent (sub-)circuit that has the same number of gates.
- 5) **Phenotype equivalency**
  - The possibility to encode a digital circuit in dierent ways.

The Evolutionary Algorithm Used

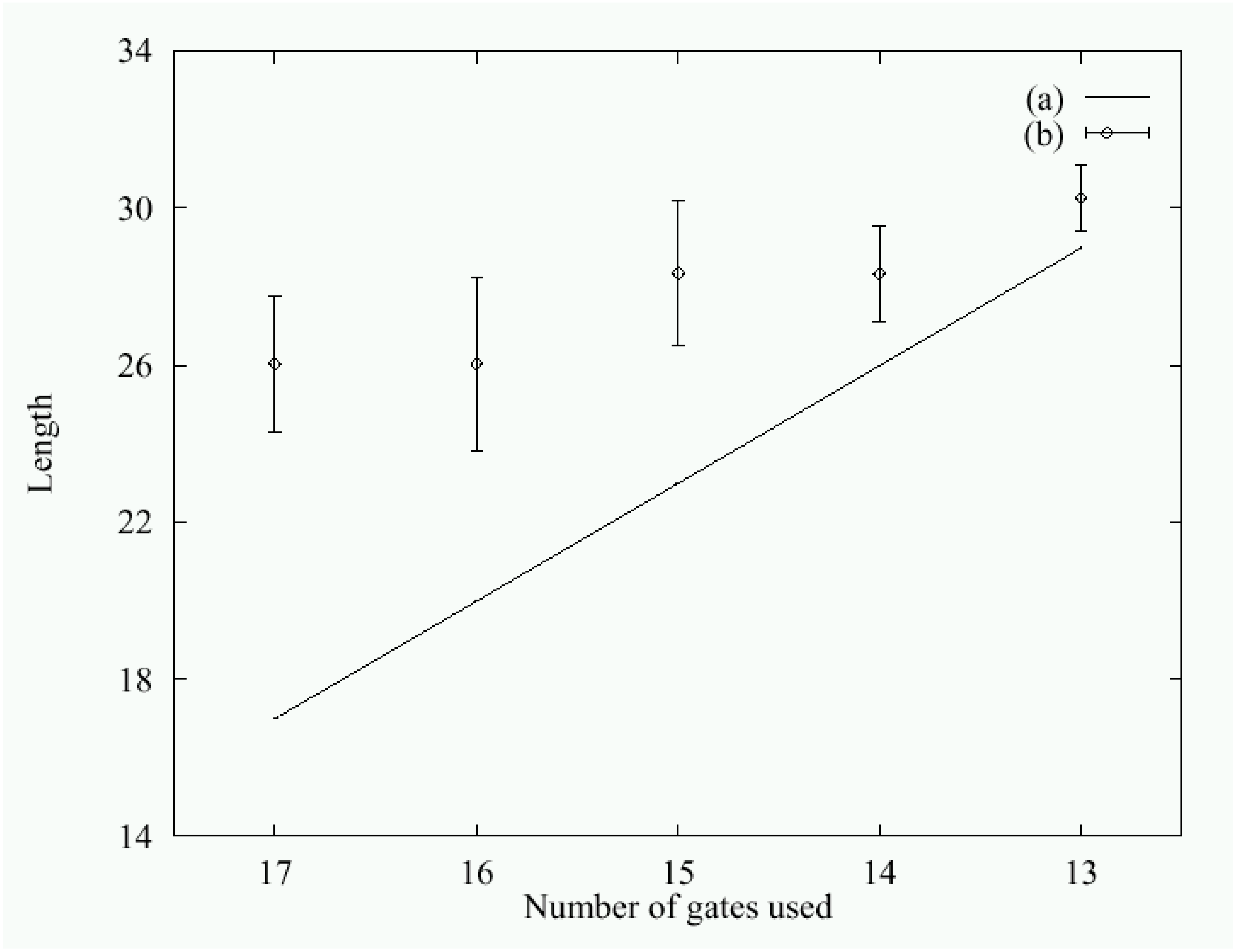
- 1) Initialize the population with a functionally correct circuit and mutated copies of the circuit.
- 2) Evaluate fitness of genotypes, and size of corresponding circuit.
- 3) Copy the genotype of the smallest functionally correct circuit into a new population (offspring).
- ... alternatively, if there are several genotypes of functionally correct circuits with equal size, then select one of these at random and copy it into the new population.
- 4) Fill remaining places in the new population by mutated versions of promoted best genotype.
- 5) Go to 2 until stop criterion is reached.

Results for the Three * Two-Bit Multiplier

- The evolved design ...

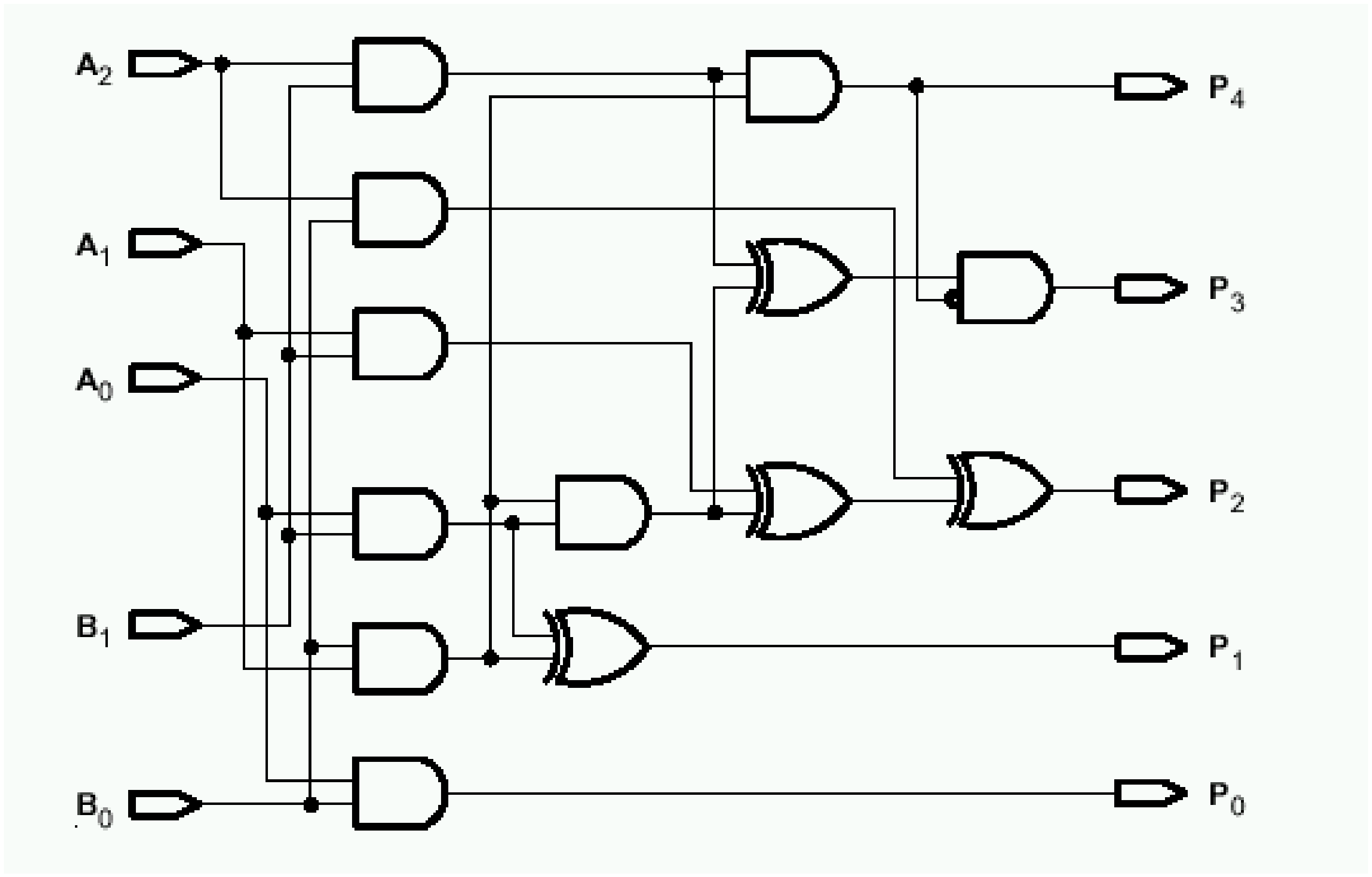


... and its optimality

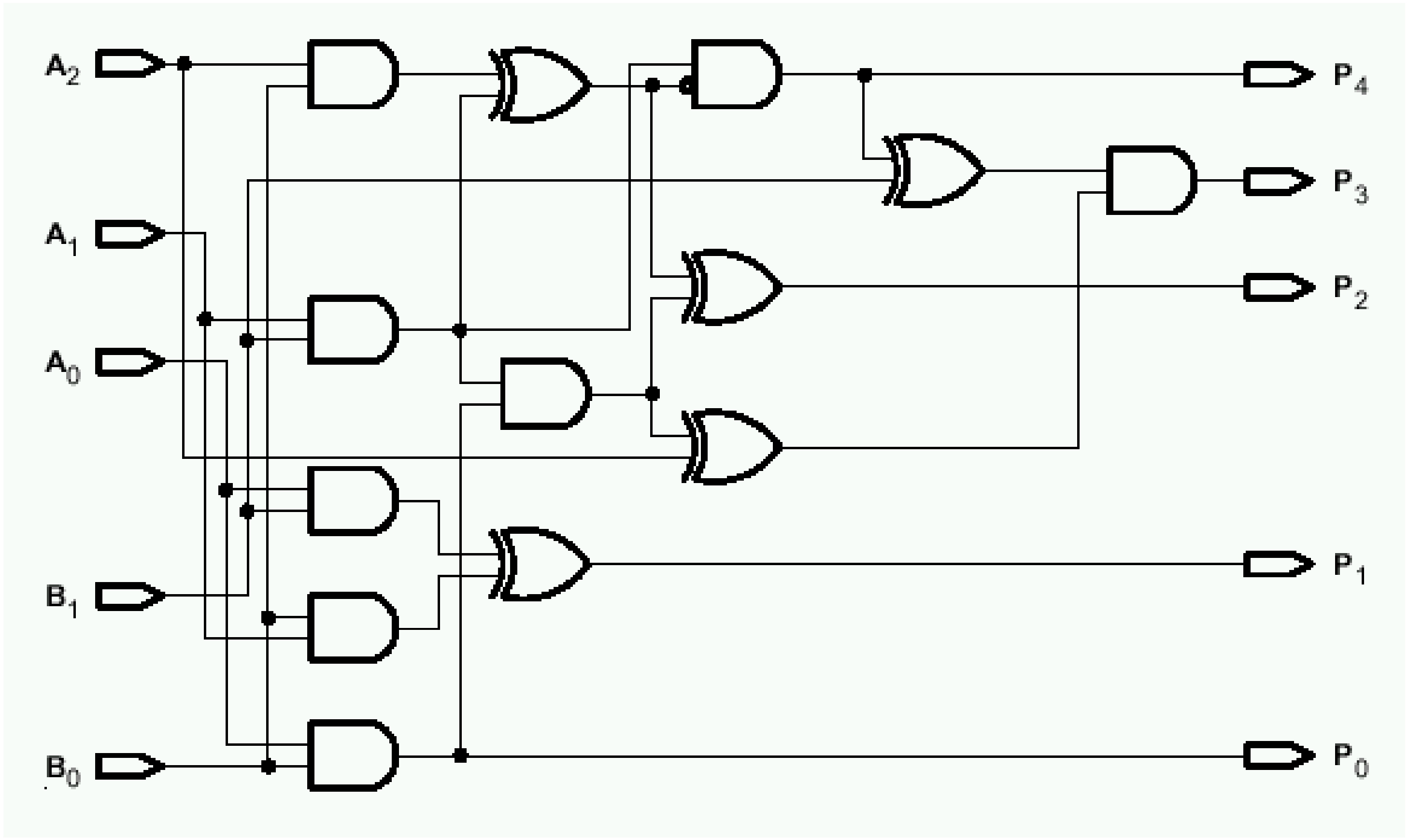


... and some more results for the Three * Two-Bit Multiplier

- The usual evolved design

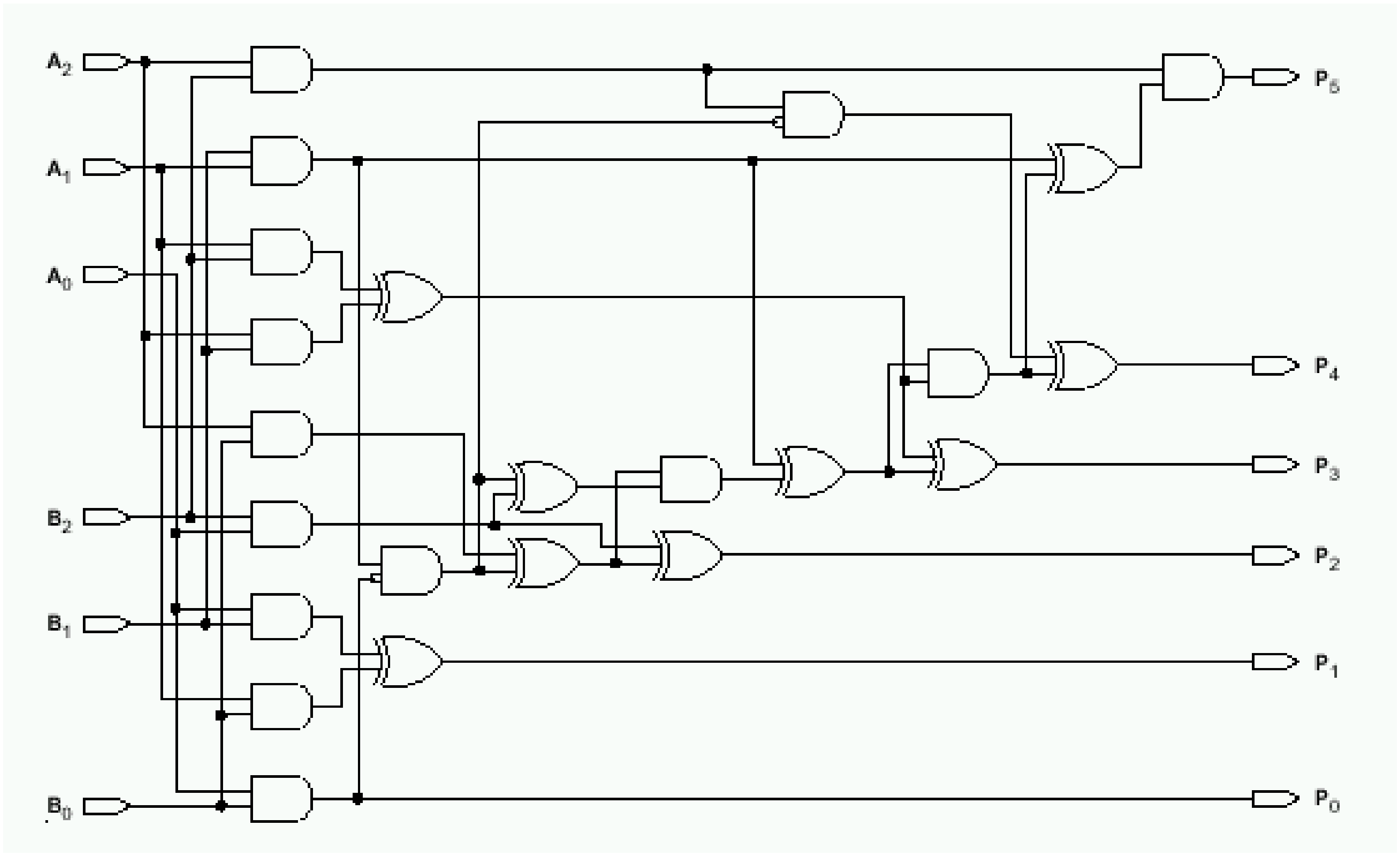


... and the unusual evolved design.

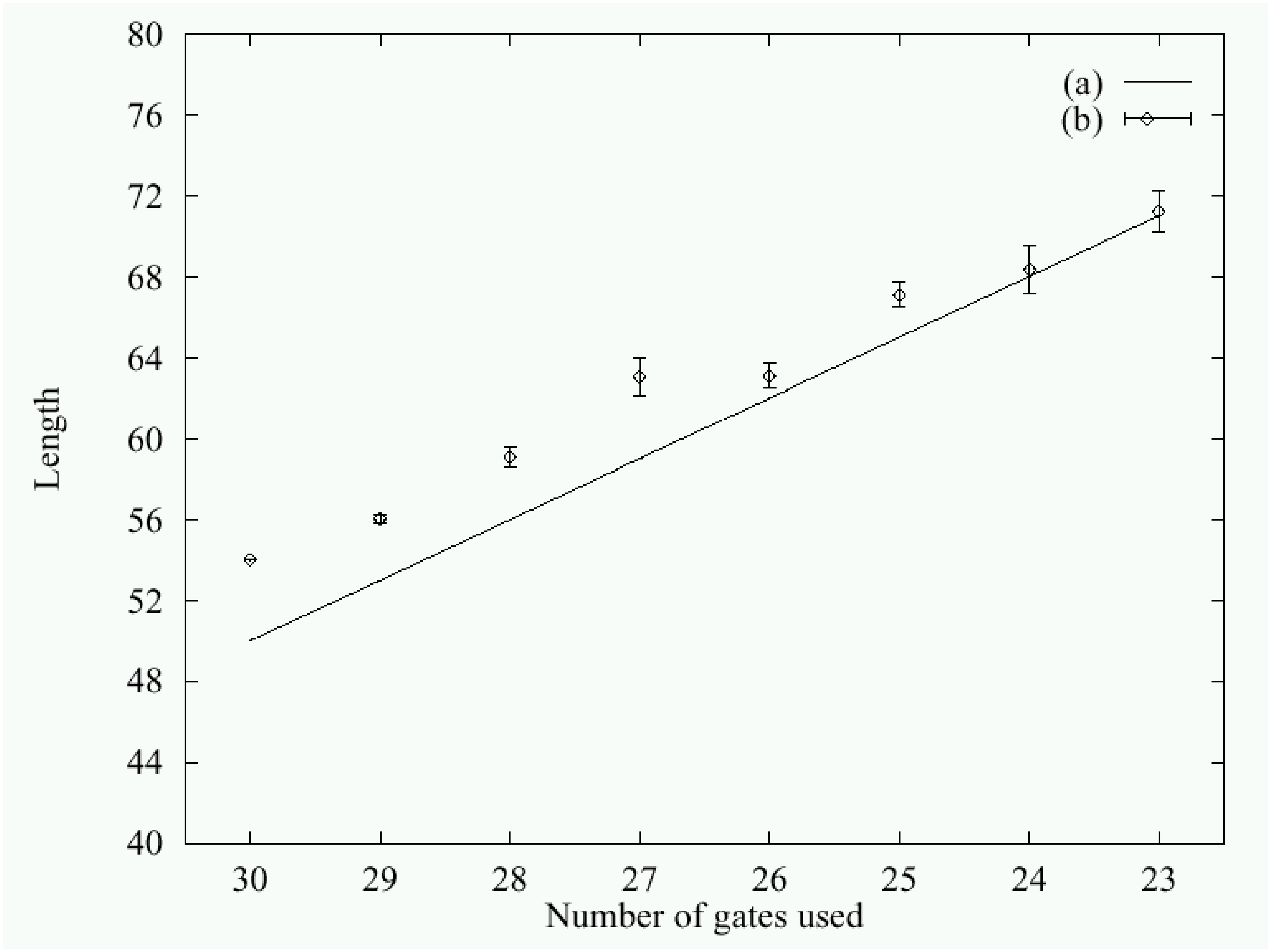


Results for the Three * Three-Bit Multiplier

- The evolved design

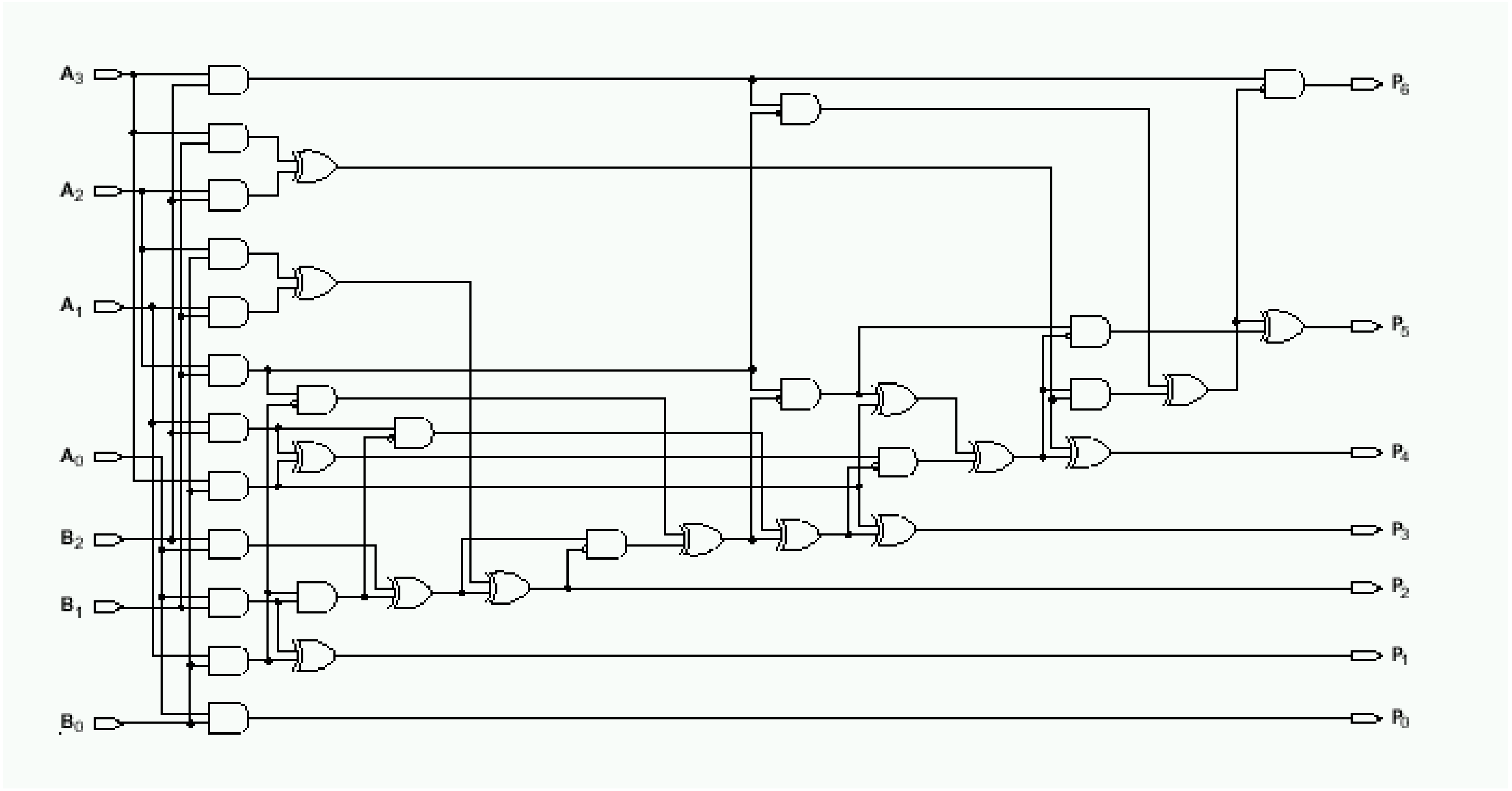


... and its optimality

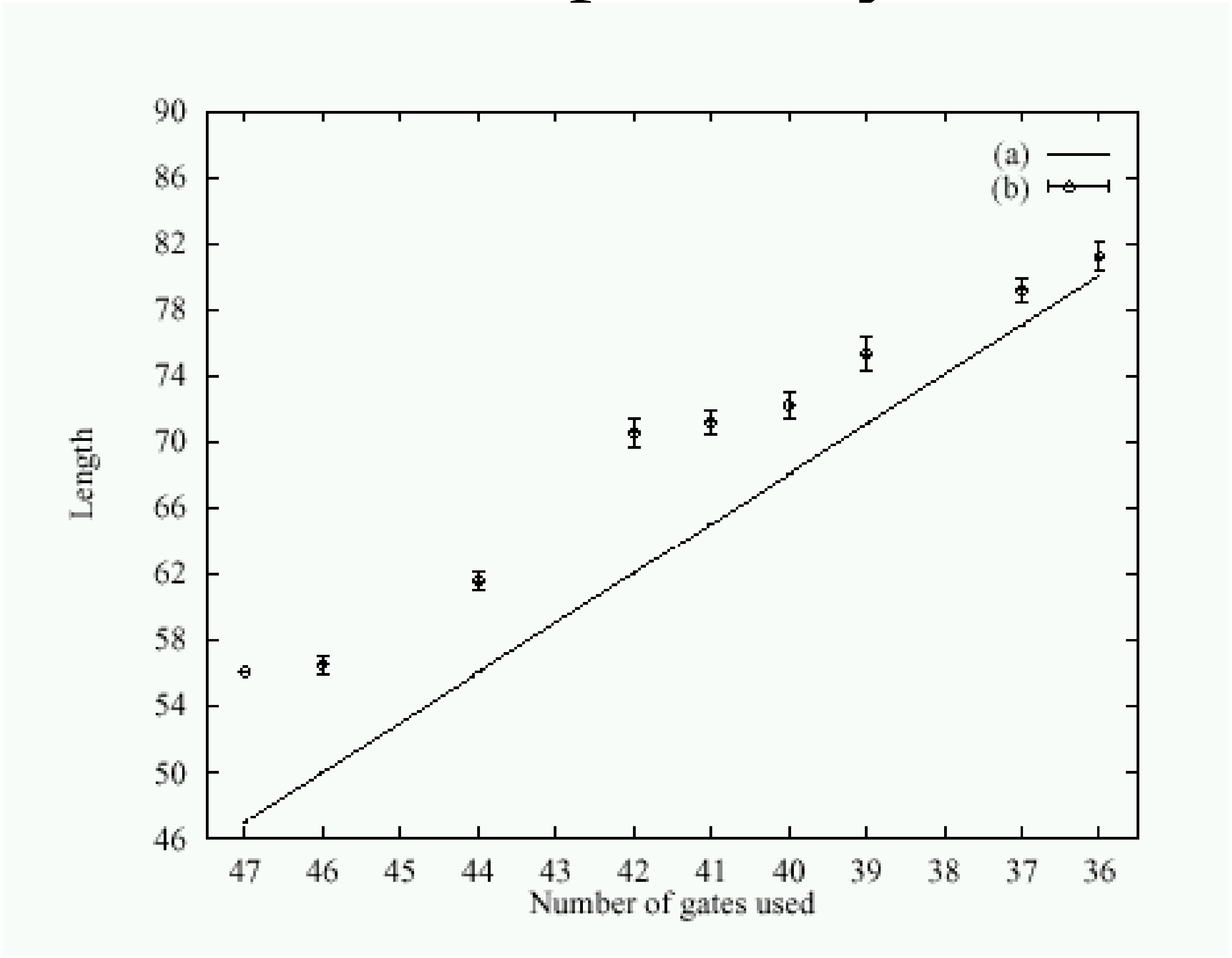


Results for the Four * Three-Bit Multiplier

- The evolved design ...

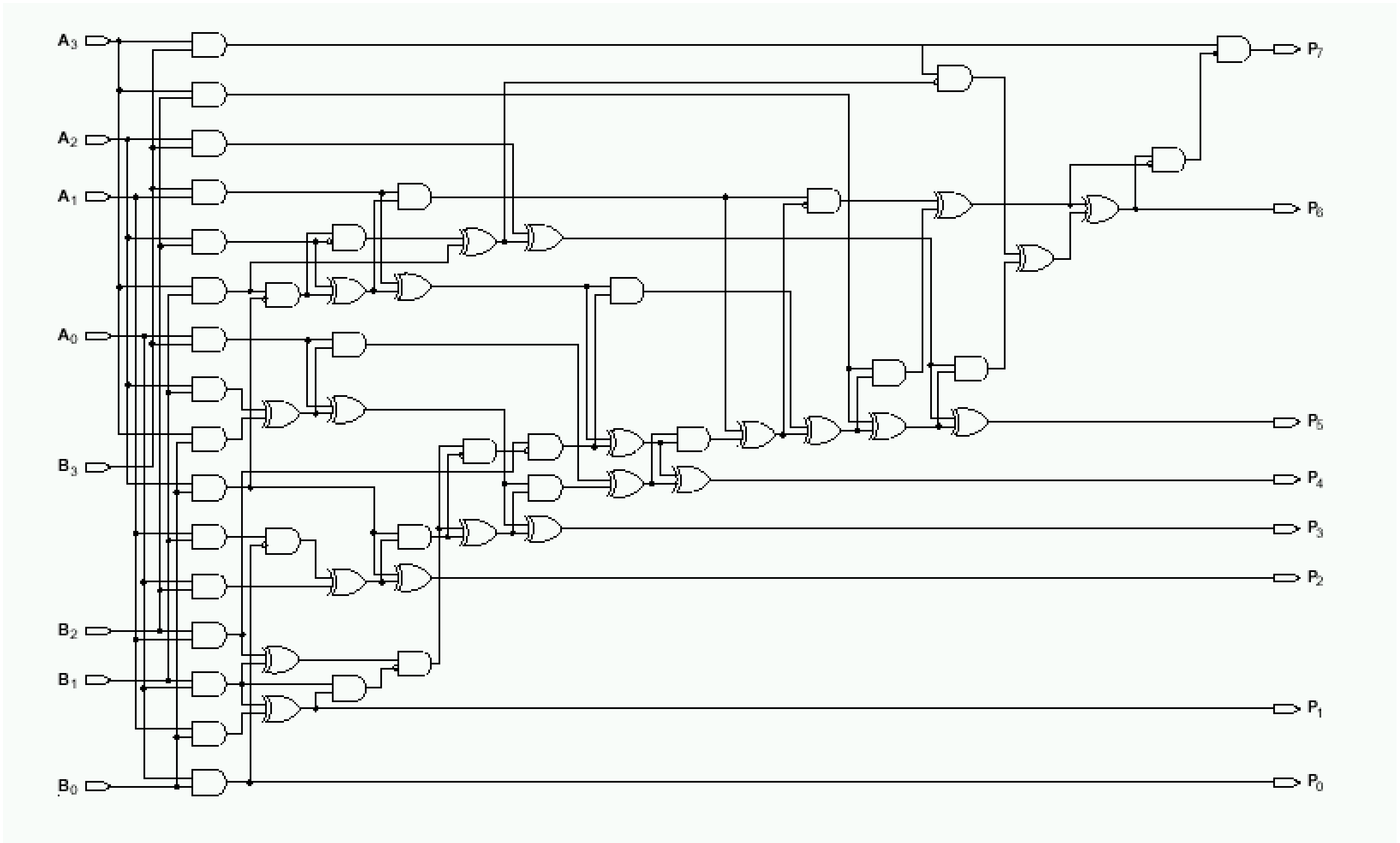


... and its optimality

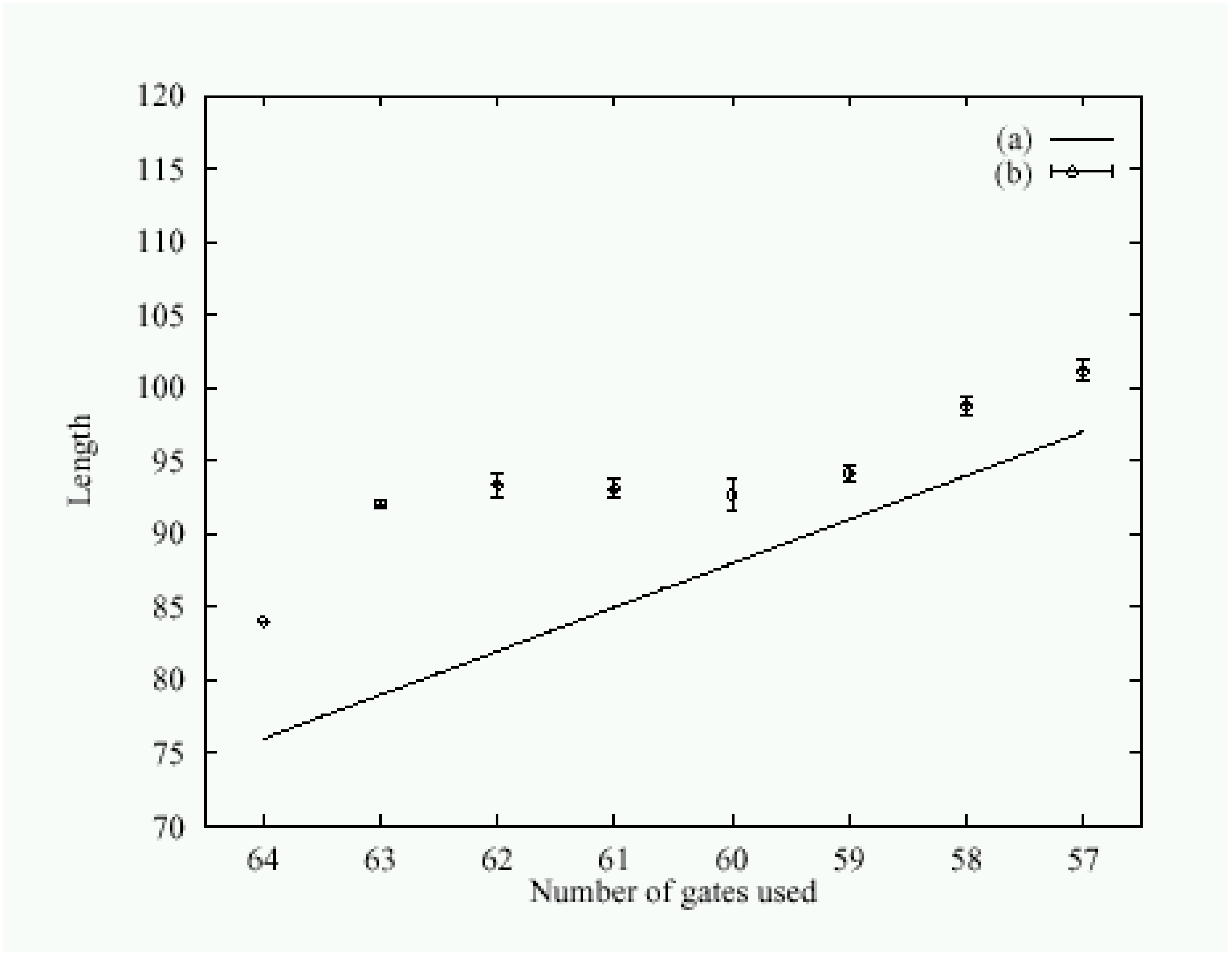


Results for the Four * Four-Bit Multiplier

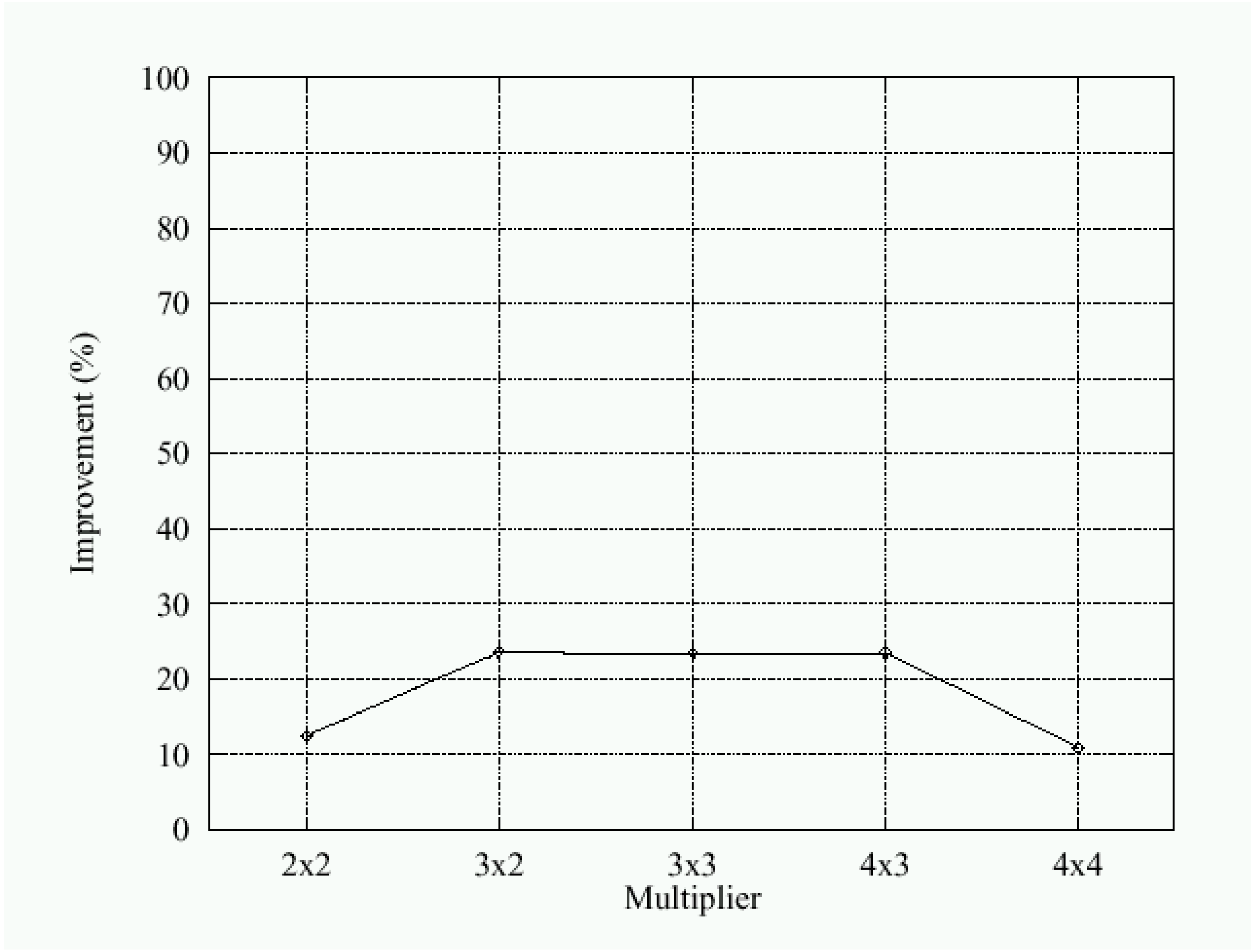
- The evolved design ...



... and its optimality

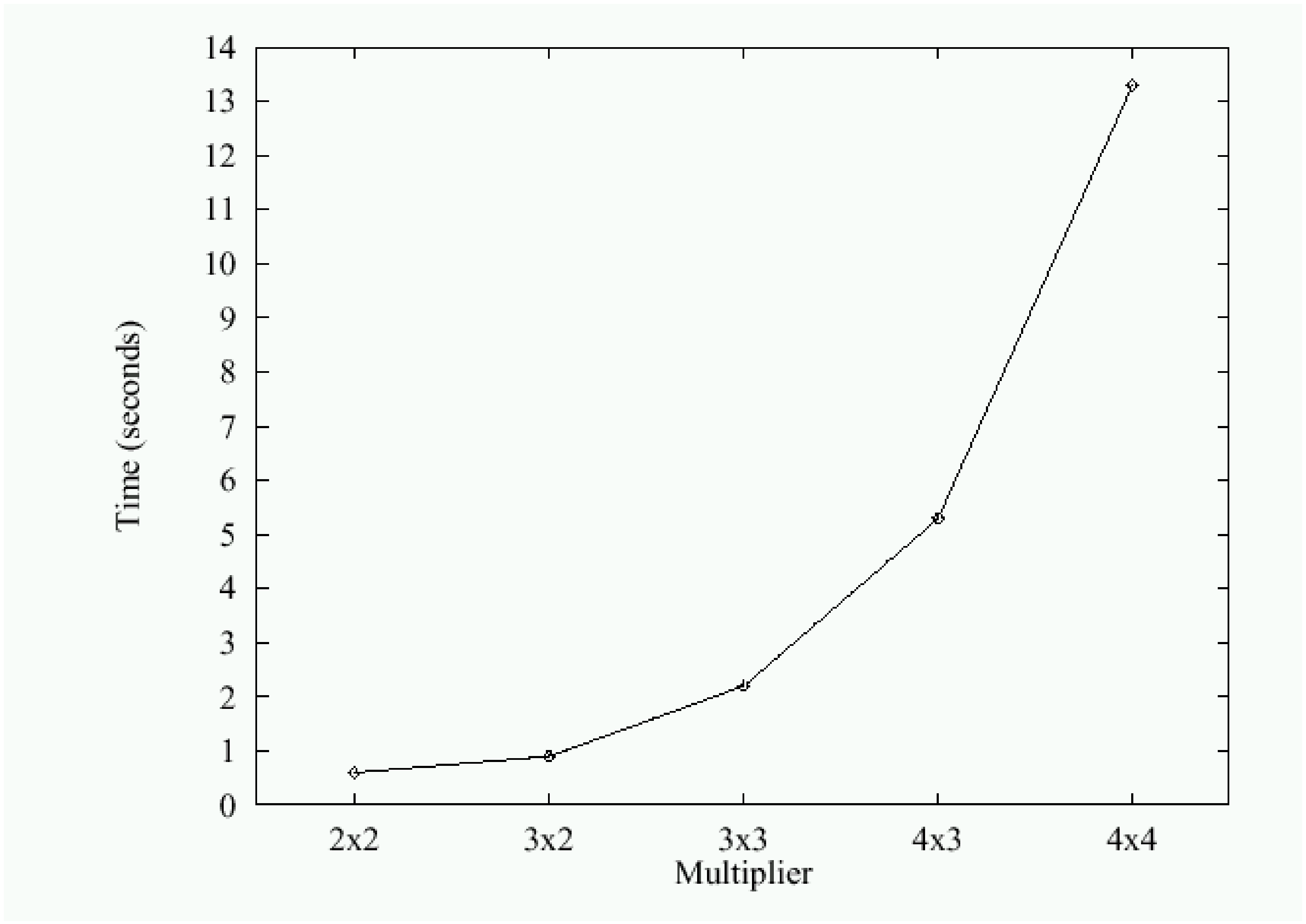


... Perhaps multiplication can be carried out in a more efficient way



- The percentage of improvements in terms of number of two-input gates attained for the binary multiplier circuits.

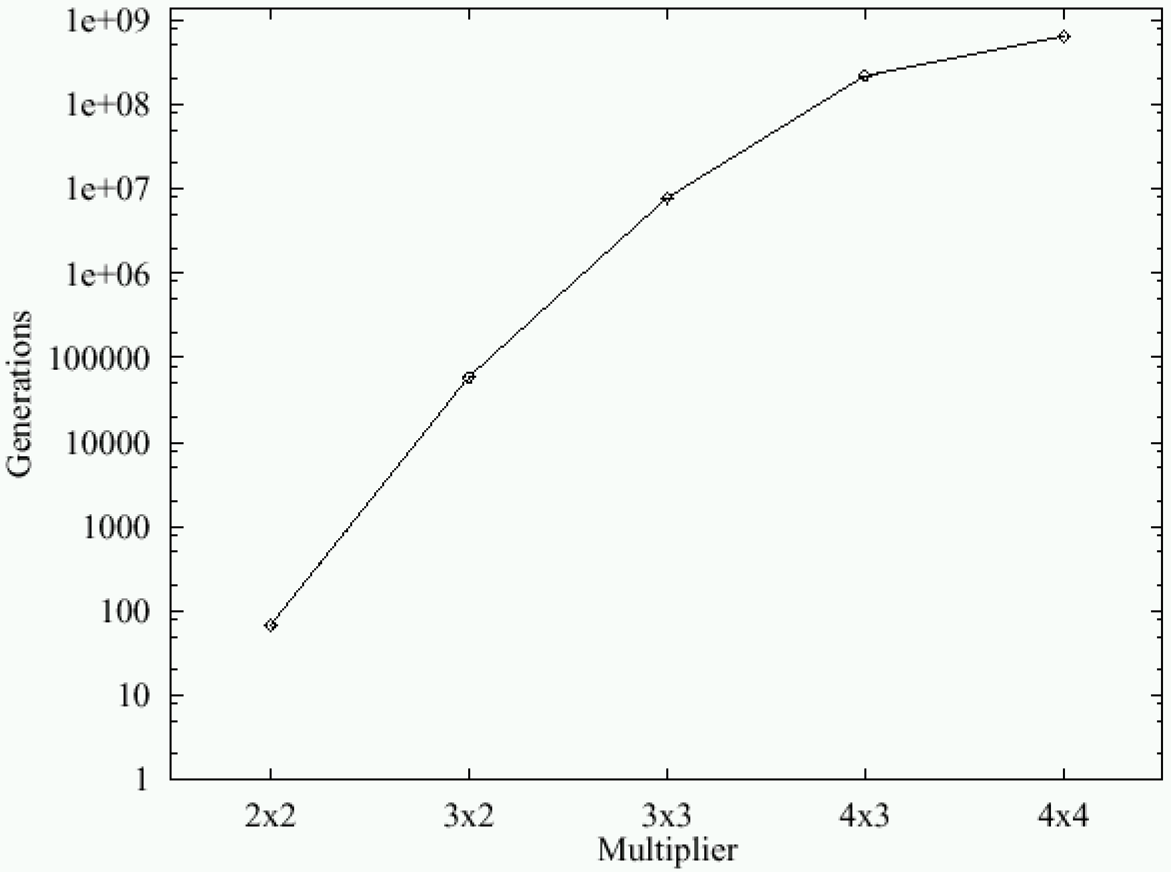
... although fast, the evolutionary design of large circuits is time consuming even on the”neutral bridge"



- The time consumed by Pentium 200MHz computer to perform 10,000 generations with a population of 5 elements for various multiplier circuits.

... the evolution of large circuits requires millions of generations

- The number of generations required to evolve the reported circuits.



Conclusions and Concerns

- 1) Embedding landscape neutrality, one could build a “bridge" in the space of all solutions between the conventional and other more efficient designs.
- 2) The technique allows the automatic design of “better" circuits, in which a functionally correct solution is guaranteed!
    - ... however, how to ensure the existence of the “neutral bridge".
- ... furthermore, how to tackle the problem of scale.
- 3) Examples of very efficient multiplier circuits have been shown.
- Does it mean that the techniques used in the conventional design are not sufficient.
- Why the design of more efficient circuits was possible.

Scalability Problems of Digital Circuit Evolution: Evolvability and Efficient Designs

Vesselin K. Vassilev
South Bank University
London
and
Julian F. Miller
The University of Birmingham
Birmingham

Digital Circuit Evolution: Concept and Final Goal

- ... by evolving novel designs ... to discern generalizable principles of design that may allow automatically to produce large and efficient electronic circuits.

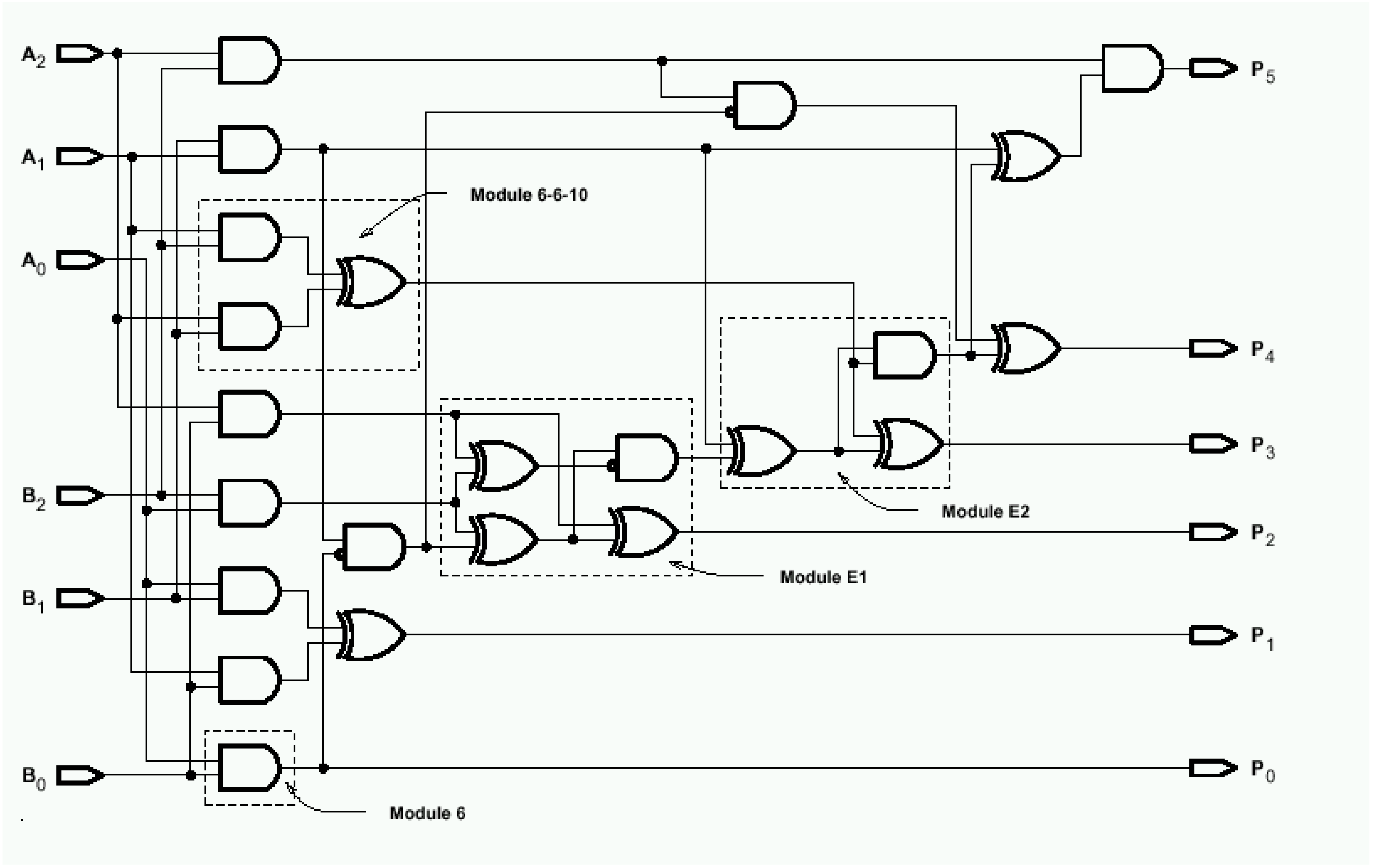
Why not “evolving" large and efficient circuits ...

- To evolve large and efficient circuits is a difficult thing to do.
- The major problem is ... the very fast growth in the number of gates used in the target circuit as the number of inputs of the evolved logic function is increased.
- This is referred to as the problem of scale.

A possible way to go ...

- ... to identify suitable building blocks that are higher functions rather than two-input gates.
- Perhaps, such blocks can be identified by looking at other evolved designs of smaller functions.

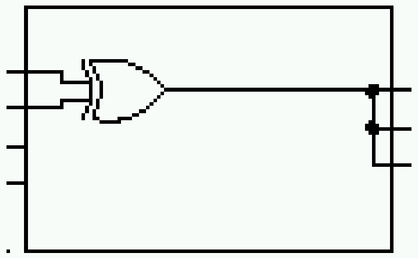
Evolving circuits and modules



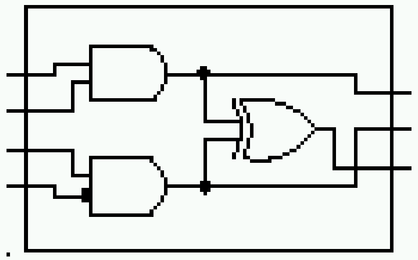
- An evolved three-bit multiplier circuits and the possible building blocks (modules).

Embedding the new building blocks within the genotype

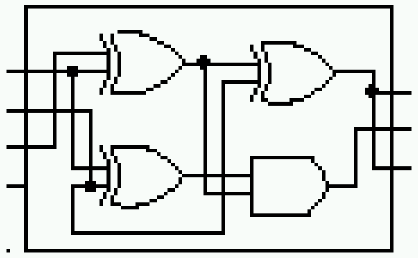
Module 10



Module 6-7-10



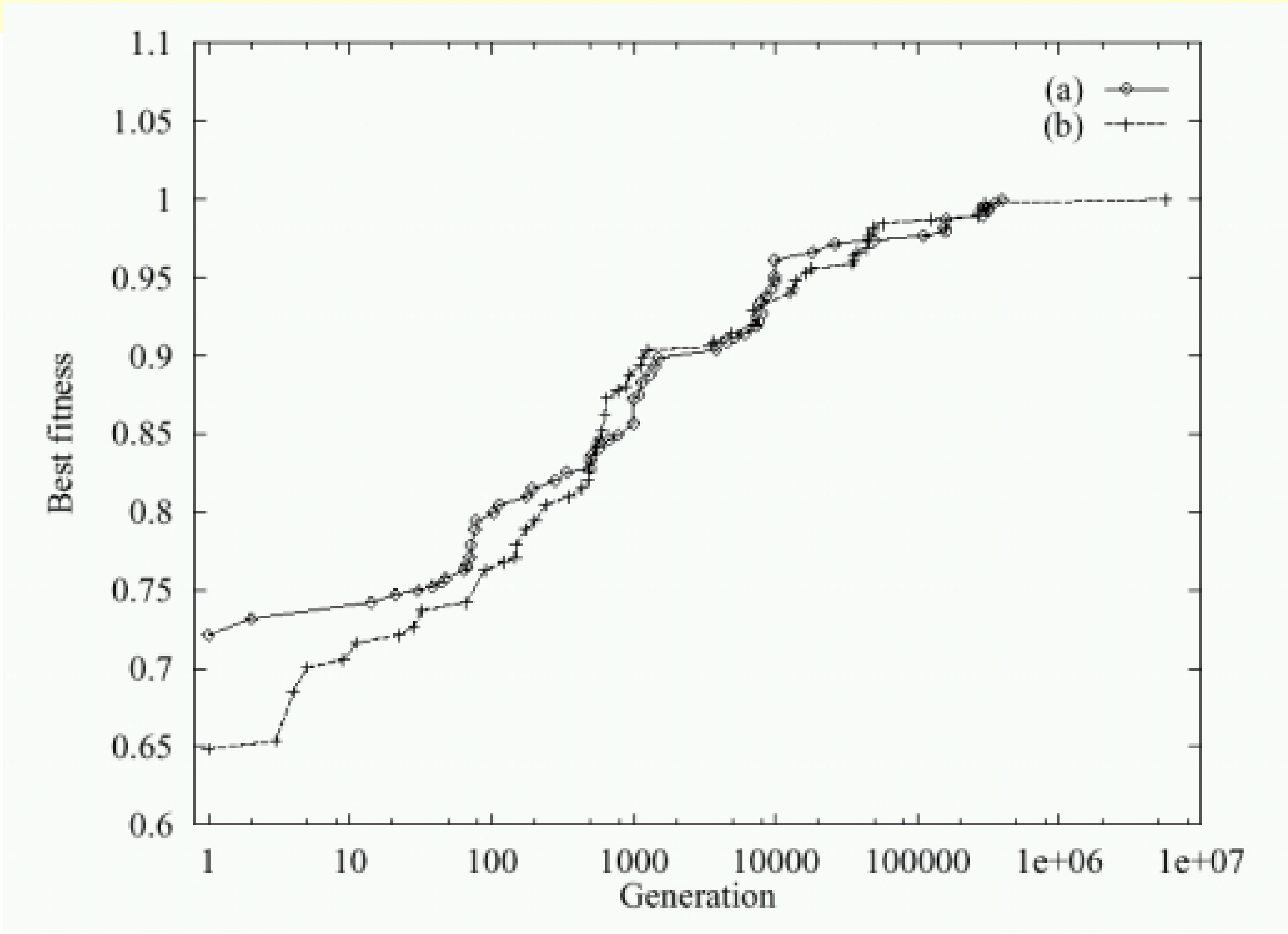
Module E_1



The Evolutionary Algorithm used

- 1) Randomly initialize a population of genotypes.
- 2) Evaluate fitness of genotypes.
- 3) Copy the fittest genotype into a new population (offspring).
- ... alternatively, if there are several equally fit genotypes, then select one of these at random and copy it into the new population.
- 4) Fill remaining places in the new population by mutated versions of promoted best genotype.
- 5) Go to 2 until stop criterion is reached.

Evolvability in the scaled evolutionary design



- Typical evolutionary runs for the scaled and the nonscaled scenarios of evolving the three-bit multiplier circuit.

Why is easy to evolve ...

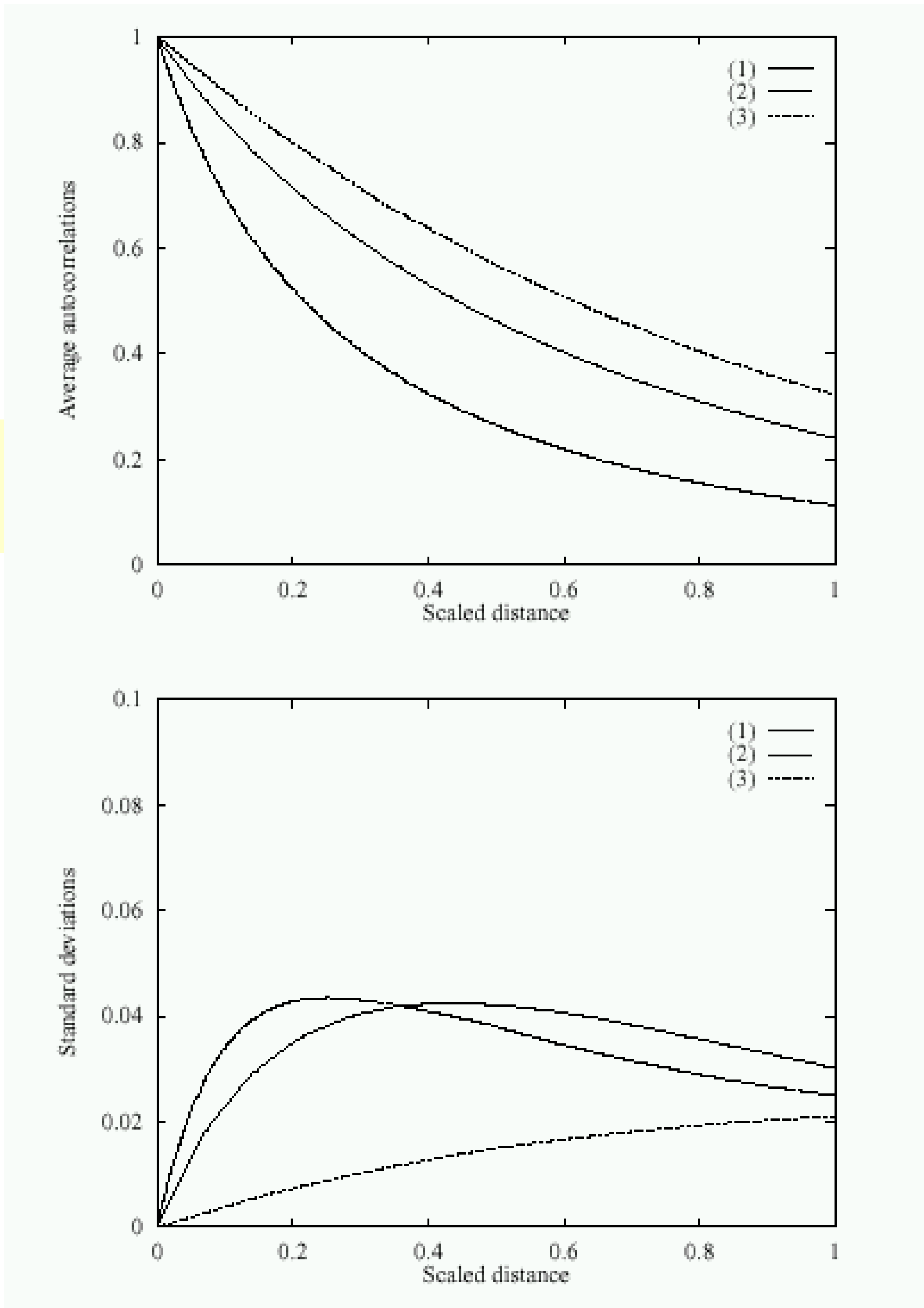
- Two possible reasons:
  - 1) The underlying fitness landscapes are easier for evolutionary search.
  - 2) The building blocks are bigger.

The structure of circuit evolution landscapes

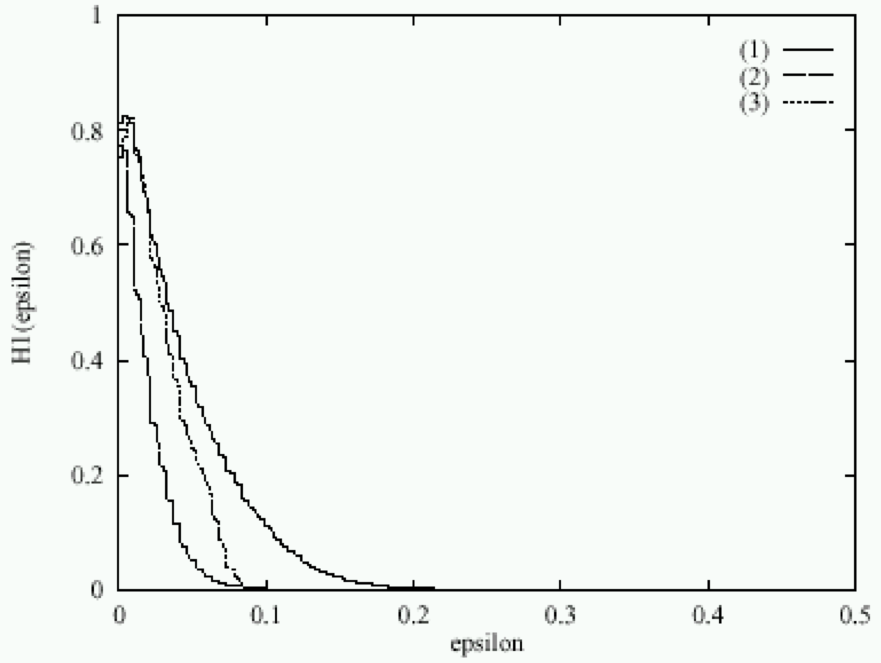
- 1) The both, scaled and non-scaled scenarios, induce landscapes with similar characteristics.
- ... therefore, the principles of evolving digital circuits discovered in the original scenario are valid when evolving with bigger building blocks.
- 2) The subspaces associated with the functionality of the array in the scaled scenario are slightly smoother than those induced by the evolution of two-input gates.
- ... therefore, by using building blocks inferred from evolved designs one may improve the evolutionary search.

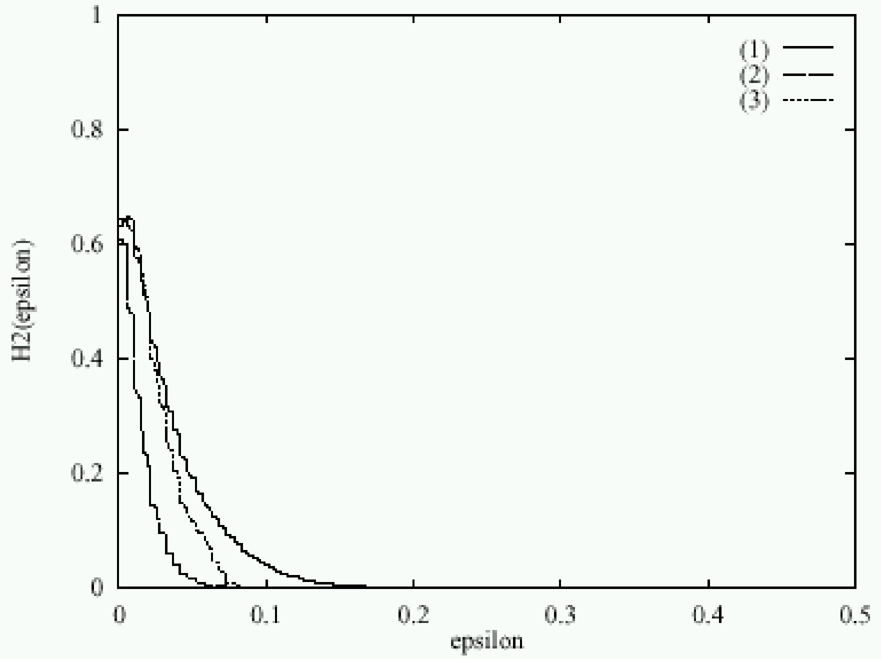
The structure of circuit evolution landscapes:

- Correlation landscape



The structure of circuit evolution landscapes: information structure





Why is difficult to evolve very efficient designs ...

- Evolution does not care about blocks.
- Evolution builds the three-bit multiplier by assembling
- the building blocks again disregarding the fact that these
- are already available.
- ... perhaps, the building blocks used are not
- the right ones?!?

- example of overlapping and strange reuse

Conclusions and Some Thoughts

- 1) The principles of evolving digital circuits are scalable.
- 2) To evolve circuits using bigger building blocks is easier.
- 3) ... However, the evolution of digital circuits with bigger building blocks does not produce very efficient designs.
- 4) To define suitable modules for the evolutionary design of some circuits such as the binary multiplier might be a very difficult task.
- ... perhaps, the most efficient multiplier is a basic computational unit ???