Co-evolving architectures for cellular machines

Moshe Sipper

Eytan Ruppin

www.moshesipper.com

Motivation

- CAs can be evolved to perform non-trivial computational tasks.
  [Packard, 1988],
  [Mitchell *et al.*, 1993–],
  [Sipper, 1994–] (non-uniform CAs).

- The density task.

- Previous studies: one-dimensional CAs.

- We extended our studies to two-dimensional CAs.
  Results: markedly higher performance, shorter computation times.

- A two-dimensional, locally connected grid can be embedded in a one-dimensional grid with local and distant connections ($O(\sqrt{N})$).

Goal: Study the relationship between performance and connectivity on a global task, in one-dimensional CAs.

- Find more efficient CA architectures via evolution.

- Possible application to the solution of a general wiring problem for a set of distributed processors.

Outline

- Architecture considerations.
- Cellular programming.
- Fixed architecture performance.
- Co-evolving architectures.
- Co-evolving low cost architectures.
- Conclusions, Discussion, The Future.

Architecture considerations

- *Architecture*: connectivity pattern of CA cells.

- Standard 1D architecture: r local neighbors on either side.

- Our model: Non-uniform CAs with non-standard architectures.

- *Circulant graphs*: $C_N(n_1, n_2, \ldots, n_k)$. n_j- connection lengths.

- *Distance* between two cells: number of connections traversed on shortest path.

- Our architectures: $C_N(a, b)$.

Architecture considerations (cont'd)

Conjecture:

- High performance on global tasks requires rapid information propagation throughout the CA.

- The rate of information propagation across the grid inversely depends on the average cellular distance, *acd*.

Architecture considerations (cont'd)

- *acd* landscape is extremely rugged. Reason: if $\gcd(a, b) \neq 1$ the *acd* is markedly higher than when $\gcd(a, b) = 1$.

- Every $C_N(a, b)$ architecture is isomorphic to a $C_N(1, d')$ architecture, for some d', referred to as the *equivalent* d'.

- We may therefore study the performance of $C_N(1, d)$ architectures. Conclusions applicable to $C_N(a, b)$.

- General $C_N(a, b)$ architectures considered when one wishes to minimize *cost*, $a + b$.

Issues studied

1. How strongly does the *acd* determine performance on global tasks?

2. Can high performance architectures be evolved, that is can "good" d or (a, b) values be discovered through evolution?

3. Can high performance architectures be co-evolved, that exhibit low connectivity cost as well?

Fixed architecture performance

Architecture is fixed, rules evolve.

- Evolutionary runs using $C_N(1, d)$ architectures, with different values of d.

- Results: Markedly higher performance is attained for values of d corresponding to low acd values and vice versa.

- Performance is linearly correlated with acd.

- Local task: maximal performance attained using minimal d.

Performance of evolving architectures

Architecture evolves as well as rules.

- Can an efficient architecture co-evolve along with the cellular rules?
  Moreover, can heterogeneous architectures achieve high performance?
  $C_N(1, d_i)$, $C_N(a_i, b_i)$

- Modified cp algorithm: cell's genome encodes connections as well as rule table.

- Results, $C_N(1, d_i)$: Evolution succeeds in "selecting" connection lengths d_i that coincide in most cases with minima points of the *acd* graph. This is coupled with high performance.

Co-evolving low cost architectures

Architecture evolves as well as rules.
Added constraint-
low connectivity cost: d_i *or* $a_i + b_i$.

- Cp algorithm: modified fitness function.

- Results, $C_N(1, d_i)$: Low cost architectures are indeed evolved, exhibiting markedly lower connectivity cost, with only a slight degradation in performance.

- Results, $C_N(a_i, b_i)$: Better performance coupled with considerably lower connectivity cost.

Conclusions

- The performance of fixed-architecture CAs solving global tasks depends strongly and linearly on their average cellular distance.

- Compared with the standard $C_N(1, 2)$ architecture, considerably higher performance can be attained at very low connectivity values.

- High performance architectures can be co-evolved, obviating the need to specify in advance the precise connectivity scheme.

- High performance architectures can be co-evolved, that exhibit low connectivity cost as well.

Conclusions (cont'd)

- Performance landscape is qualitatively similar to *acd* landscape.

- Uniform, fixed architectures may get stuck in a low performance local minimum.

  Evolving, heterogeneous architectures, where each cell "selects" its own connectivity, end up with a melange of local minima, yielding in many cases higher performance.

- Added efficiency of $C_N(1, \sqrt{N})$ architectures suggests that the density problem has a good embedding in two dimensions.

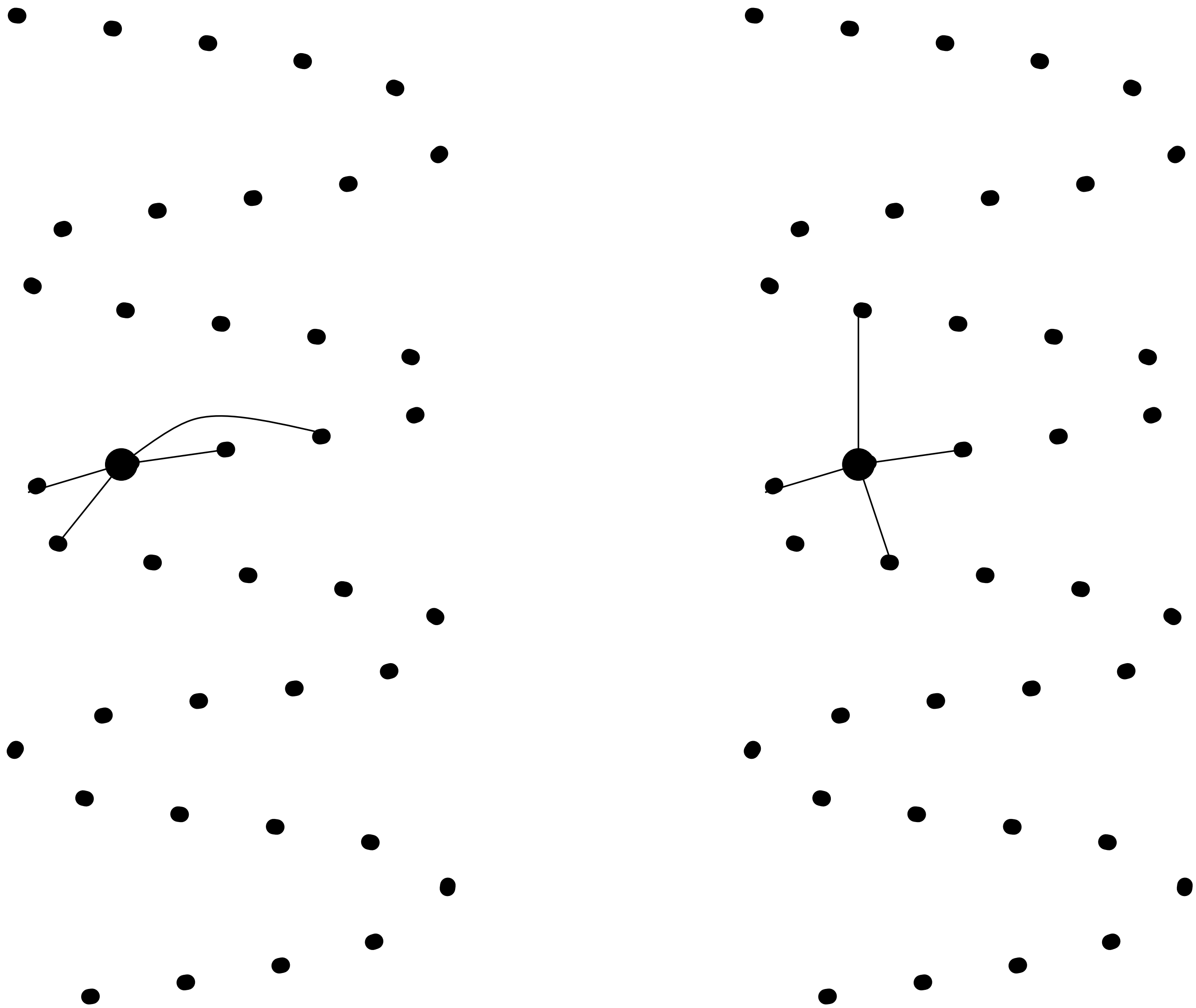
- Performance landscape may have a global maximum at $a, b = O(\sqrt{N})$ (but with $a \neq b$).

Future work

- Solve a general wiring problem for a set of distributed processors.
  - Given a set of processors, find a connectivity scheme that minimizes average processor distance.
  - Problem constraints: minimal and maximal connection lengths, pre-specified neighbors for some or all cells, etc'.

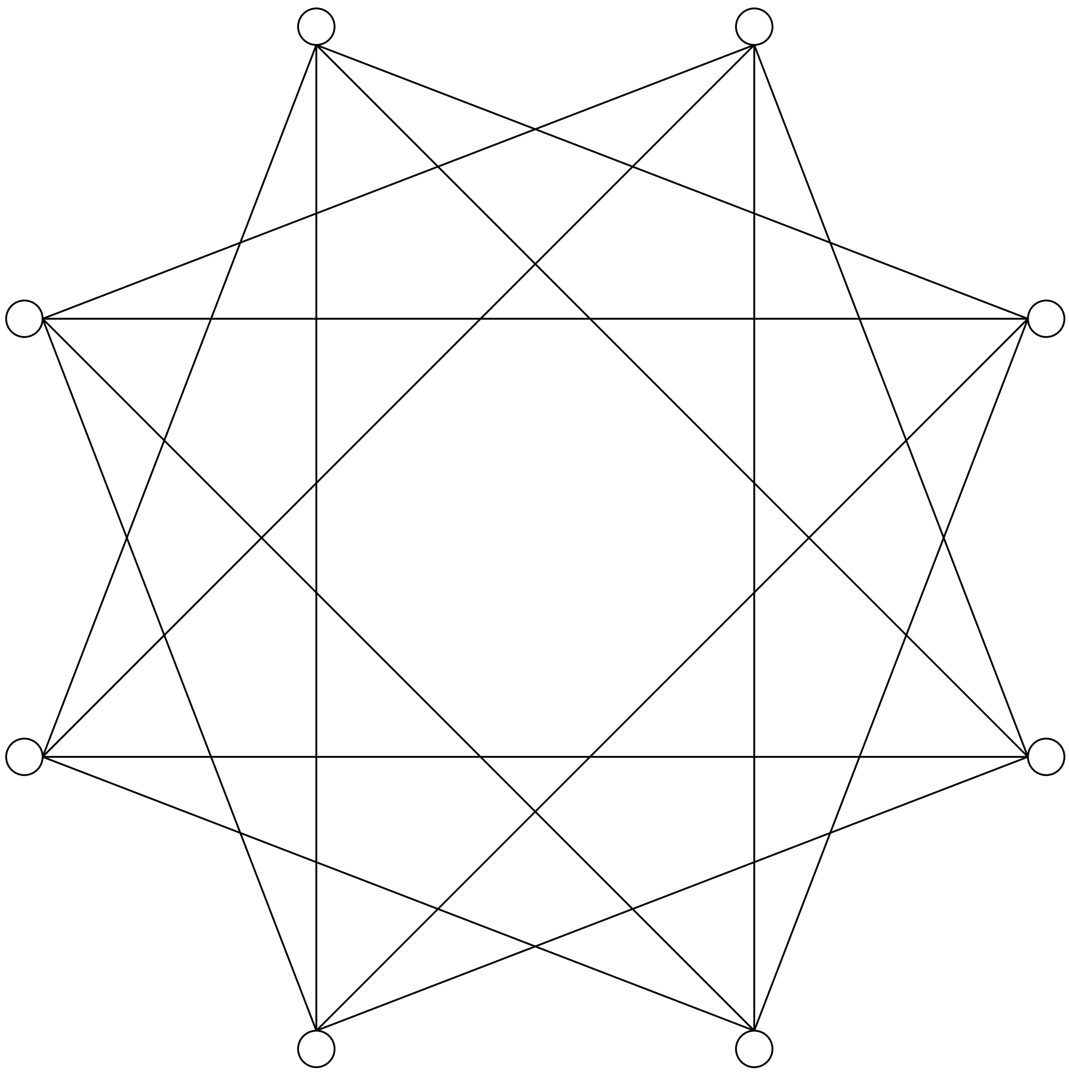
- Evolve degree of connectivity.

- Ultimately, we wish to attain a system that can adapt to the problem's inherent "landscape".

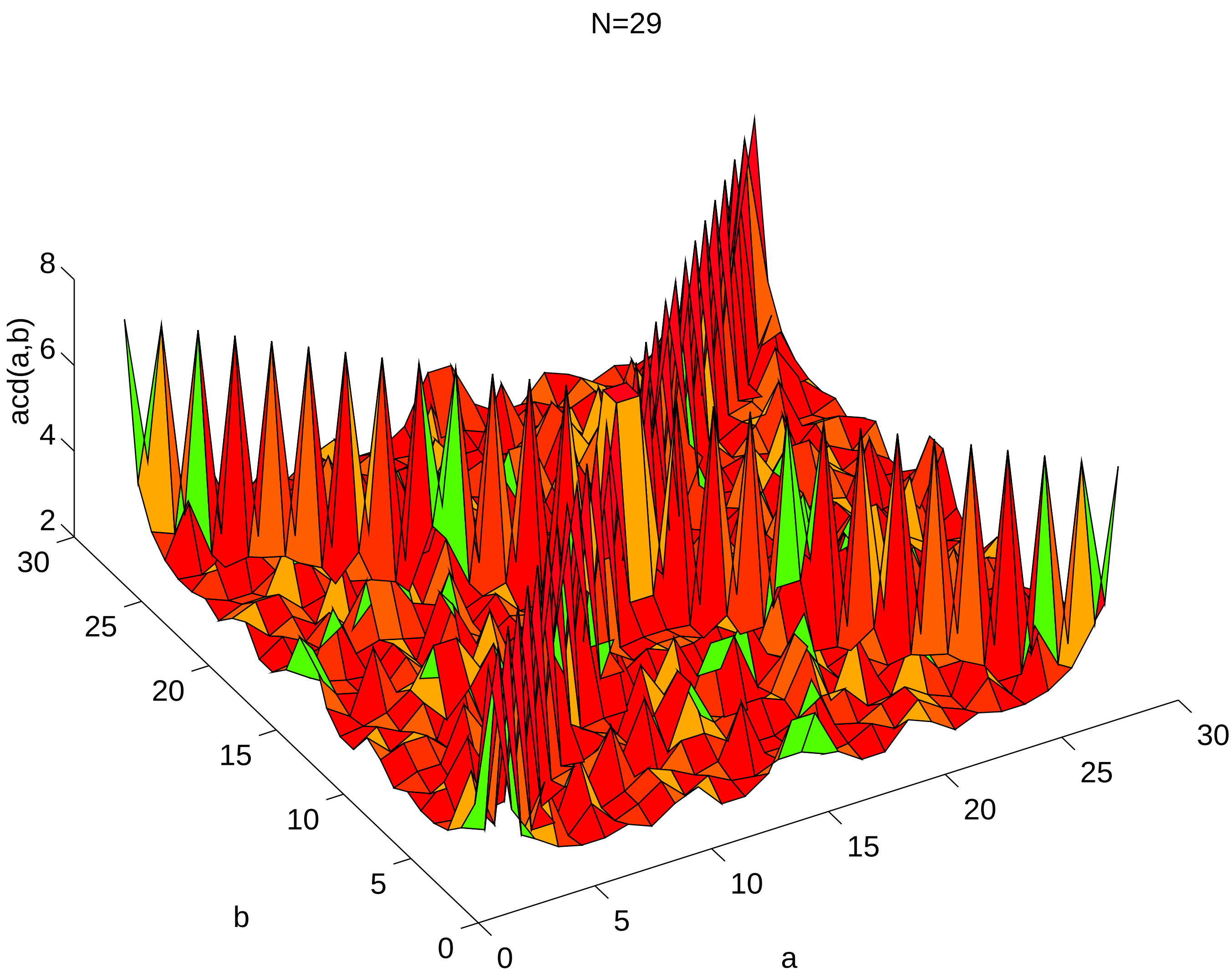


Folded 1D
(r=2)

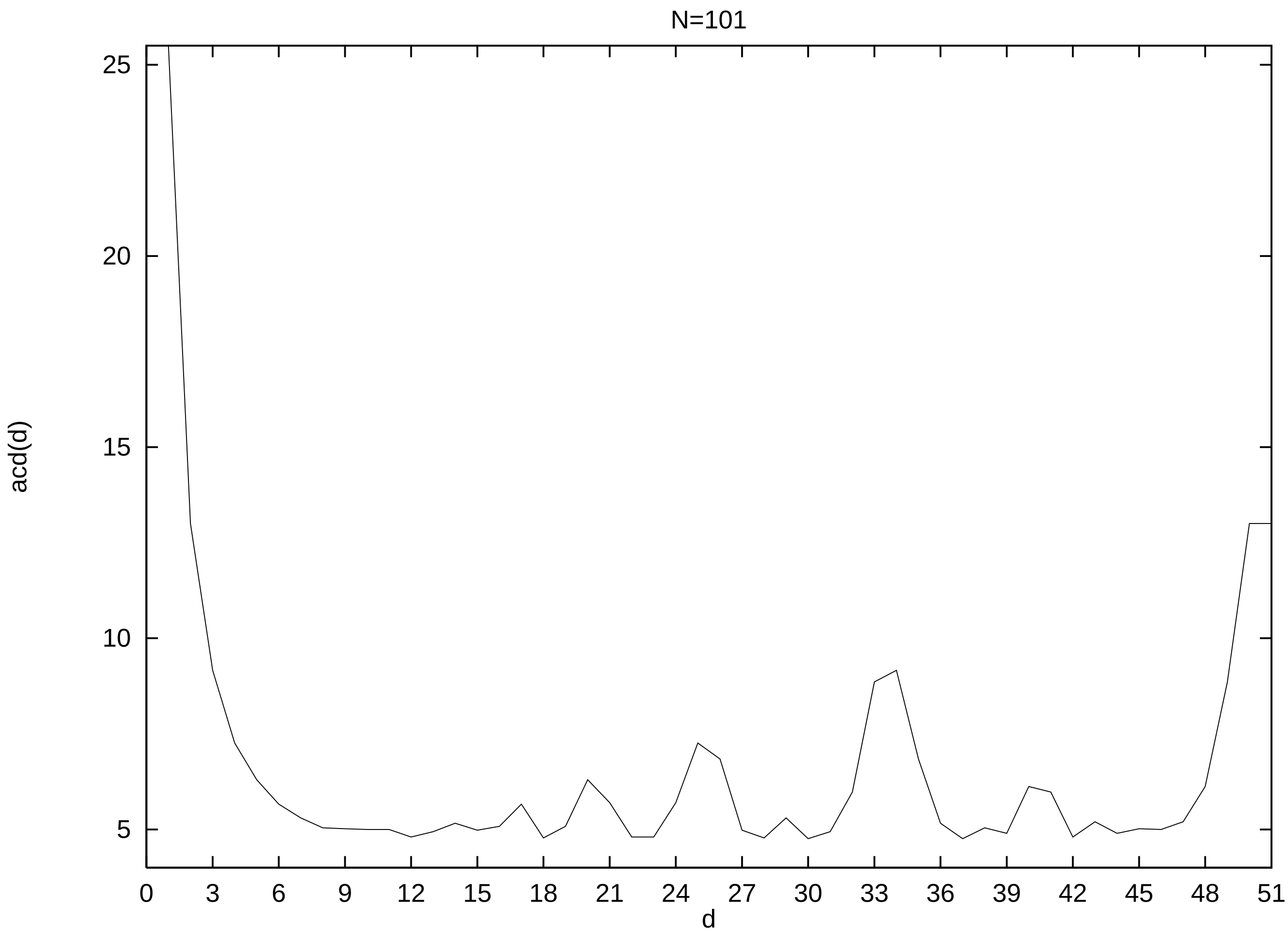
2D



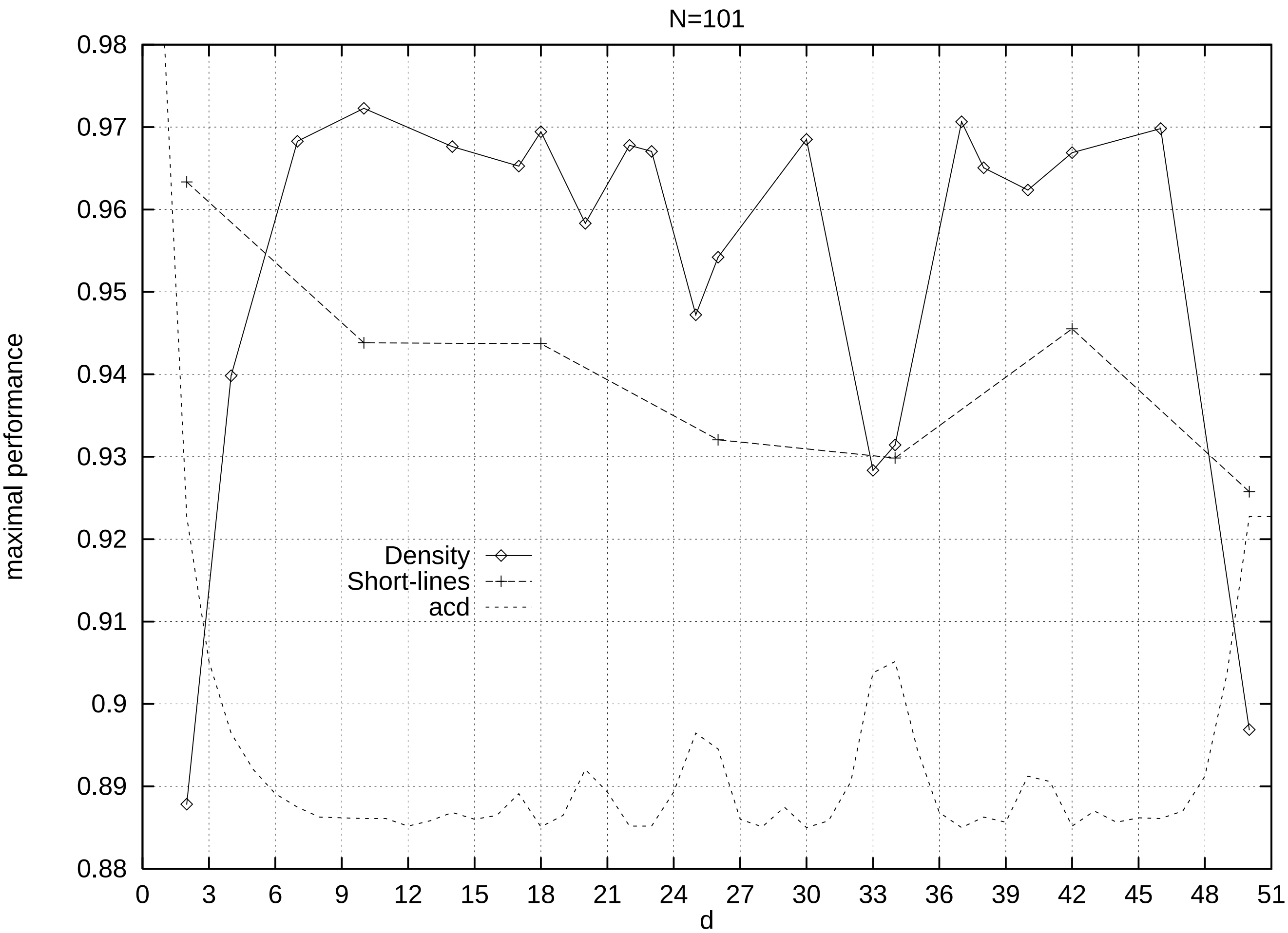
A $C_8(2,3)$ circulant graph.
Each node is connected to four neighbors, with connection lengths of 2 and 3.



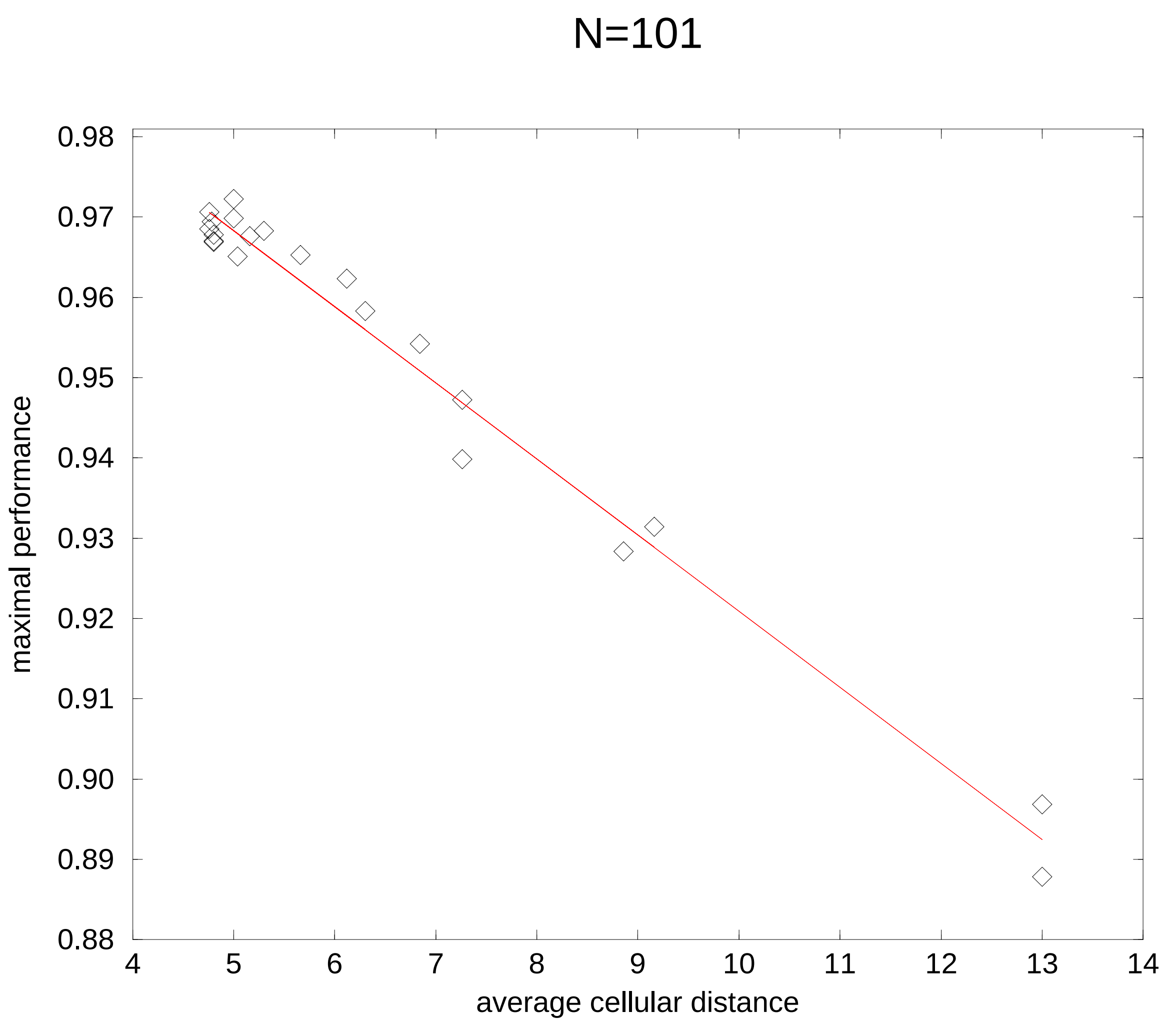
The ruggedness of the *acd* landscape is illustrated by plotting it as a function of connection lengths (a, b), for $C_{29}(a, b)$.



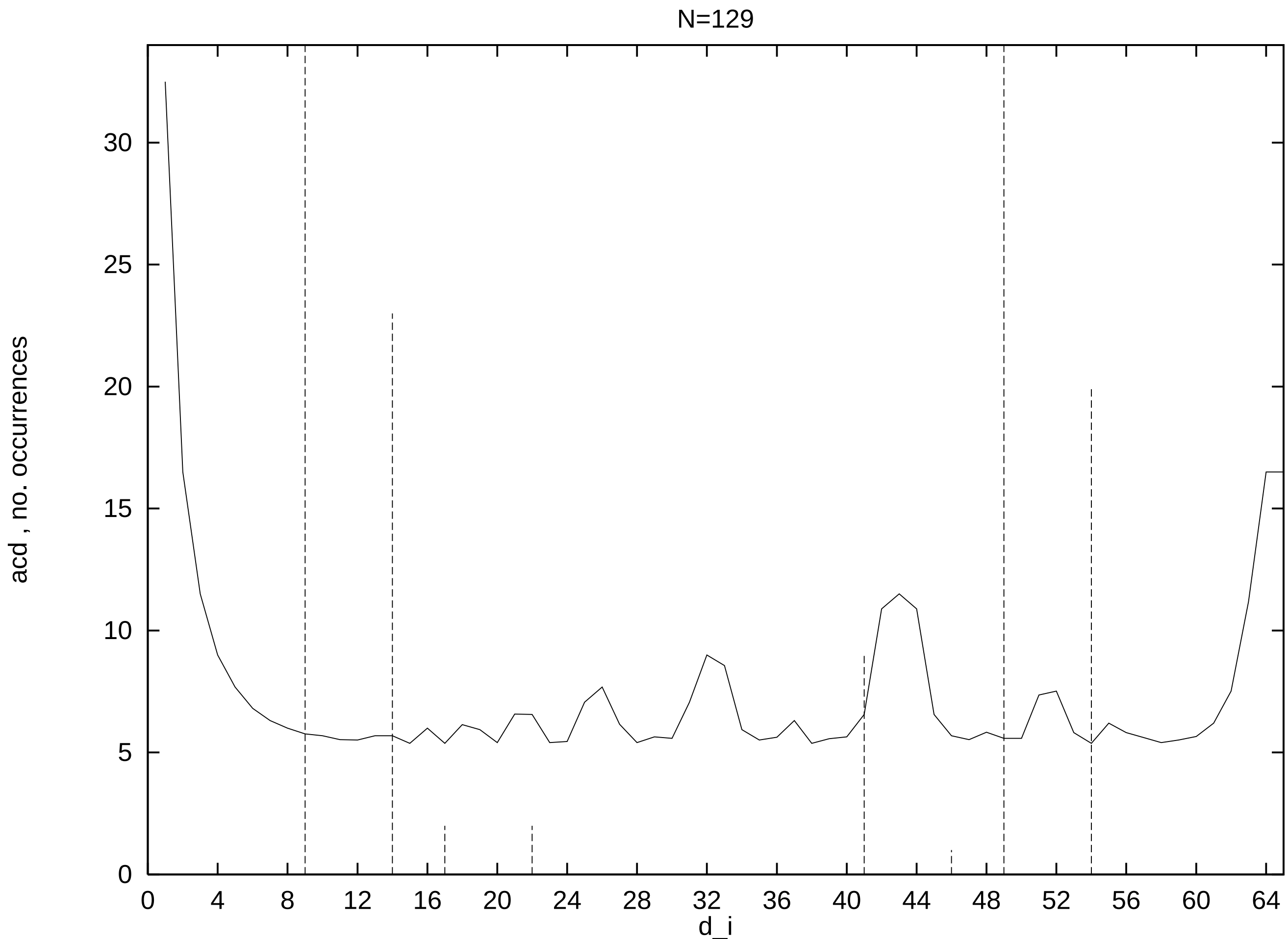
$C_{101}(1, d)$: acd as a function of d. acd is plotted for $d \leq N/2$, as it is symmetric about $d = N/2$.



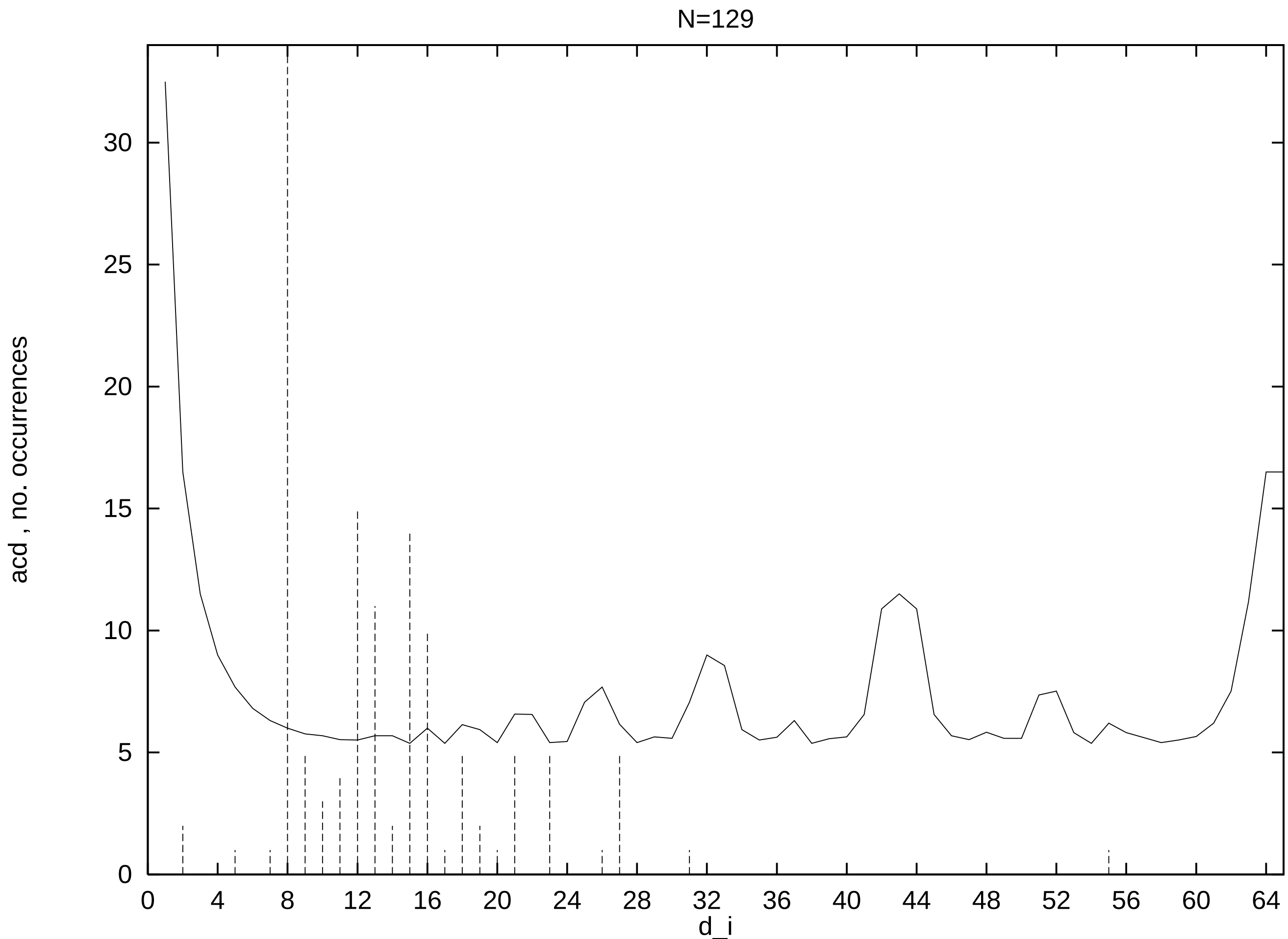
$C_{101}(1, d)$: Maximal evolved performance of density and short-lines tasks as a function of d.



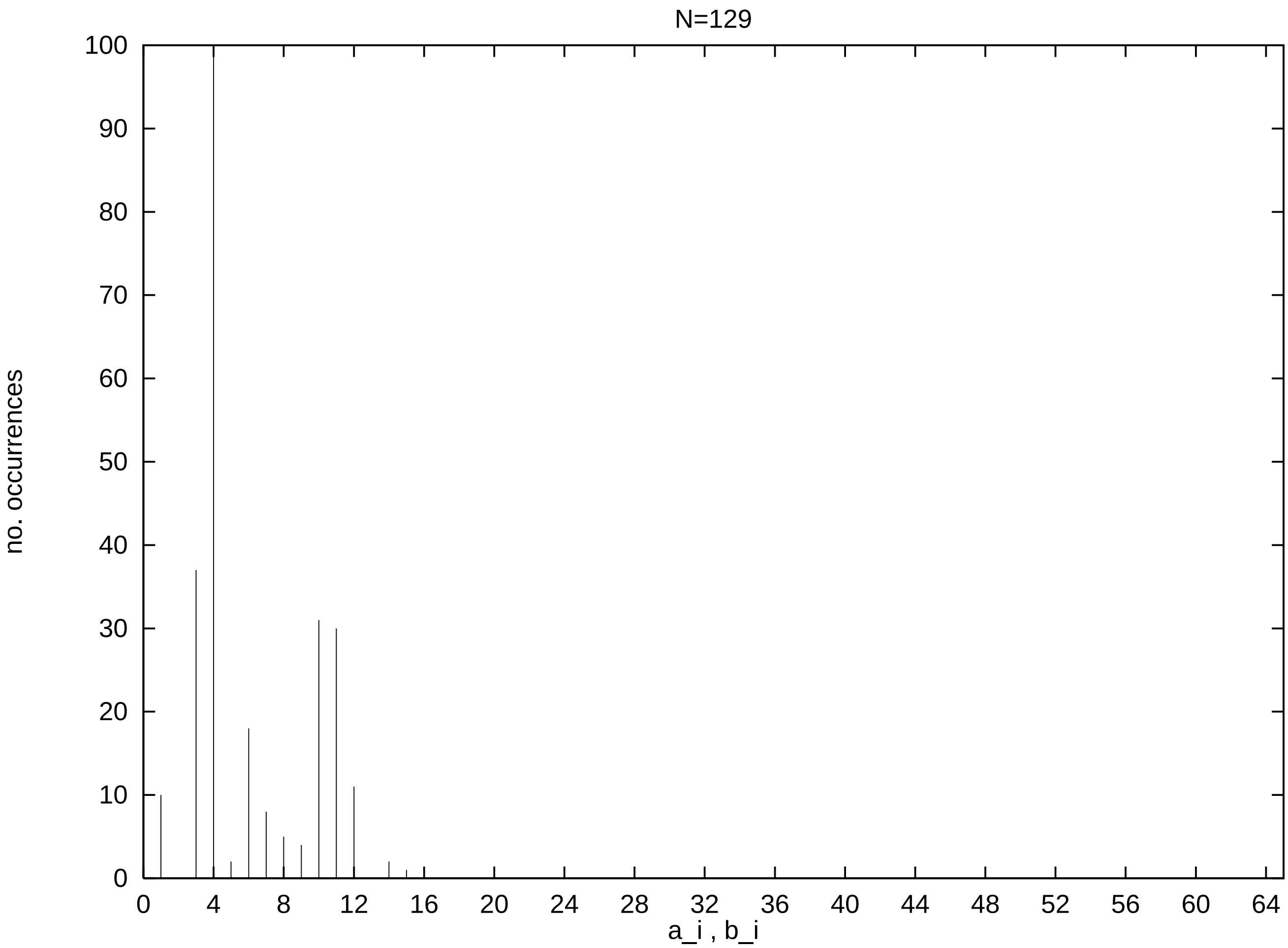
$C_{101}(1, d)$: **Maximal performance as a function of** acd**.**



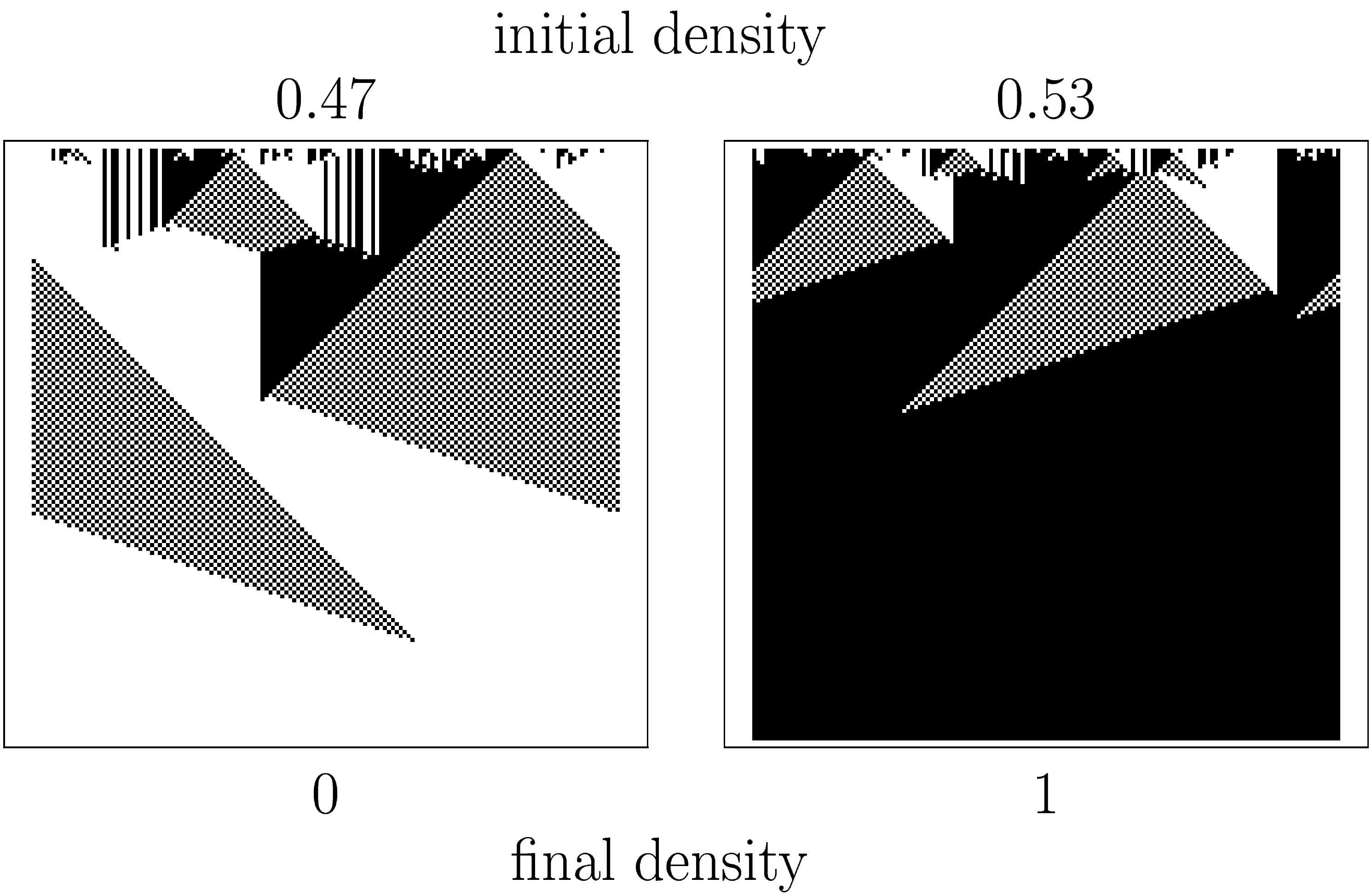
Evolving architectures.
$C_{129}(1, d_i)$.



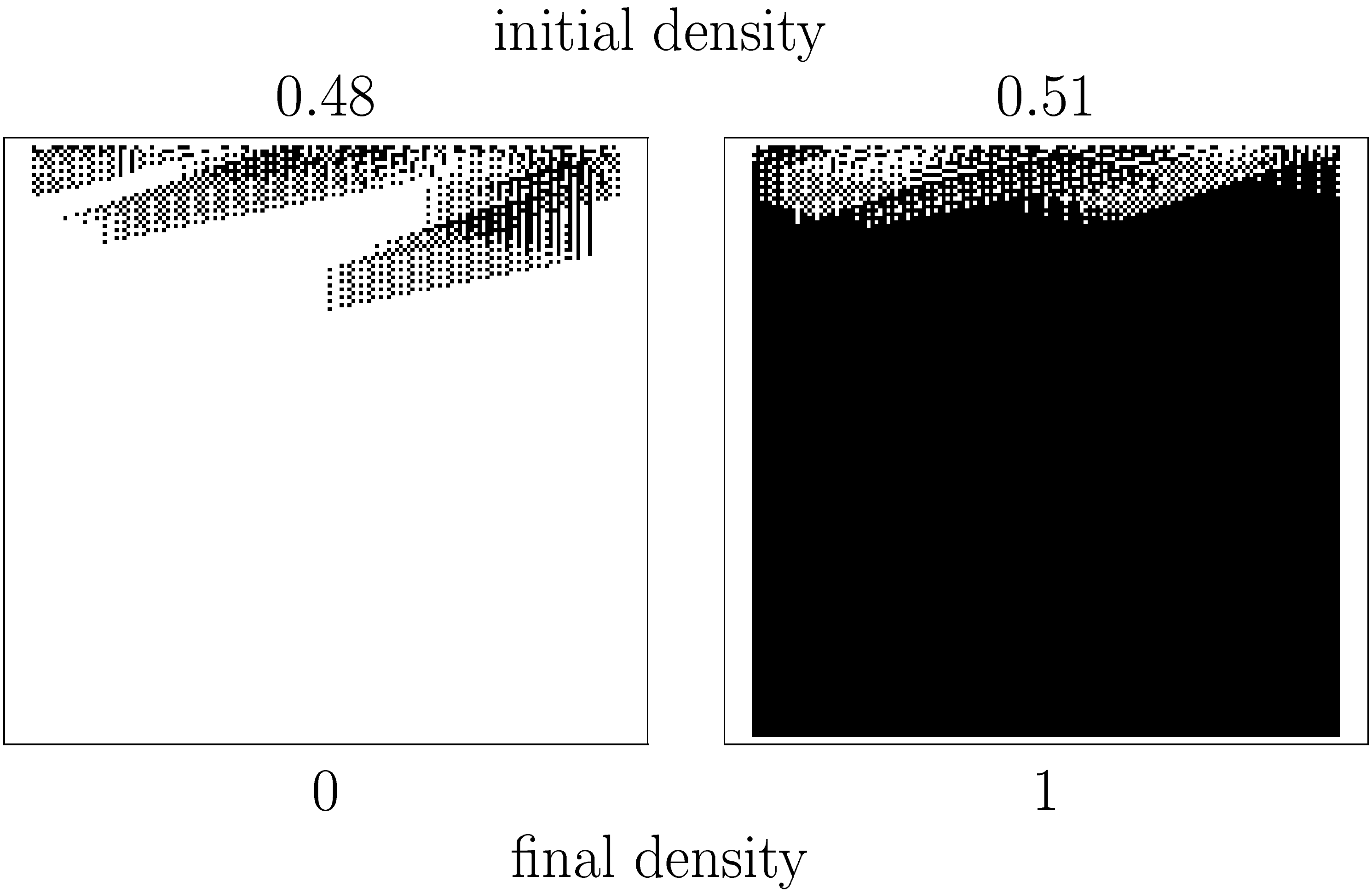
Evolving low cost architectures. $C_{129}(1, d_i)$.



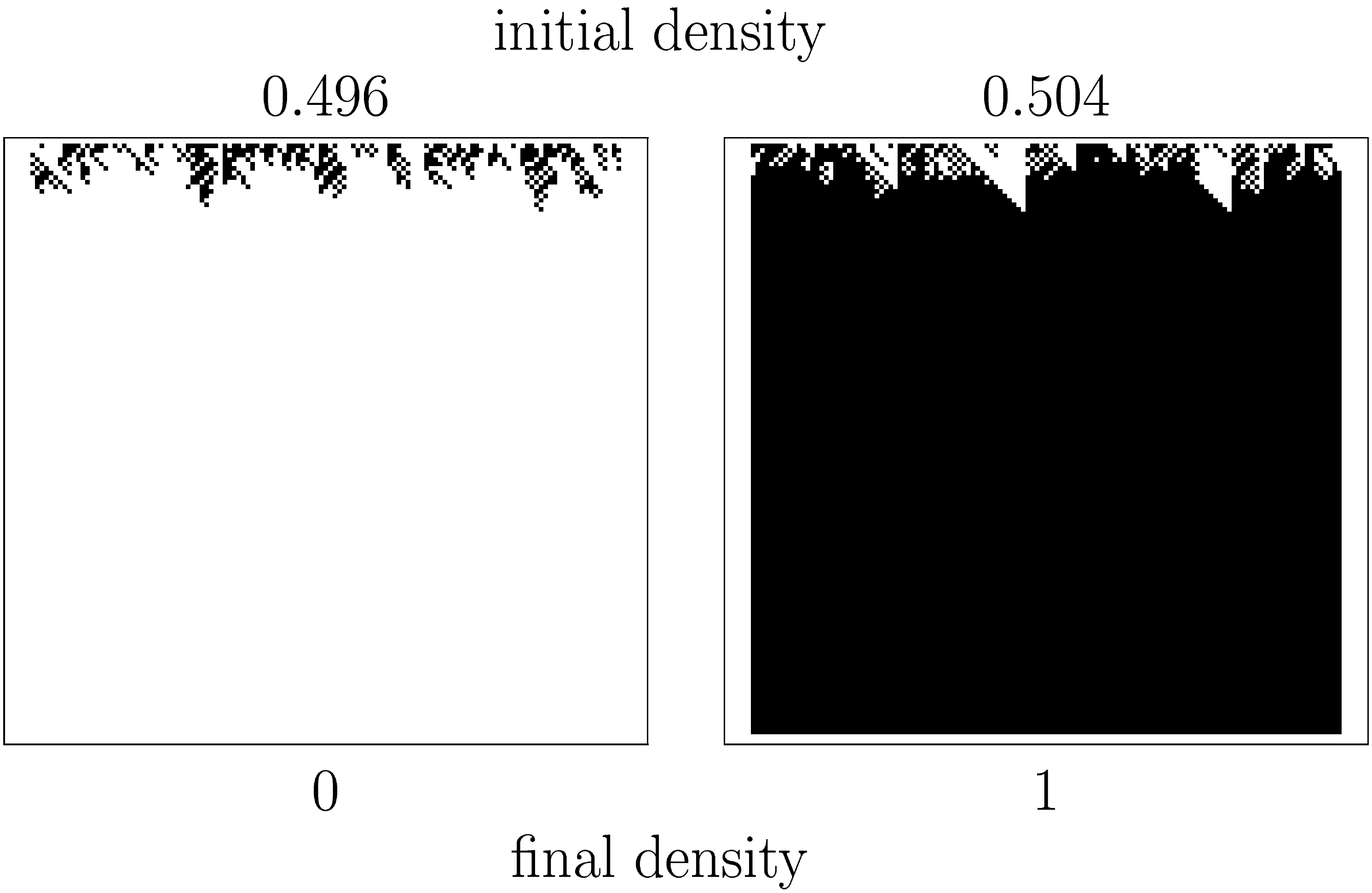
Evolving low cost architectures. $C_{129}(a_i, b_i)$.



Density task: GKL rule.



Density task: A co-evolved, non-uniform, $C_{149}(3,5)$ CA.



Density task: A co-evolved, non-uniform, $C_{129}(1, d_i)$ CA.