

Metody wzorowane na naturze:
symulowane wyżarzanie
automaty komórkowe

Maciej Mrozowski
116592 INF/IO

1. Symulowane wyżarzanie

1.1 Wstęp

Otoczająca nas natura zna wiele sposobów na rozwiązywanie różnych problemów i człowiek od zawsze starał się podparywać i asymilować pewne modele jej zachowania. Jedną z tych metod są algorytmy genetyczne (GA), wykorzystujące zasady selekcji naturalnej do generowania coraz to lepszych osobników (które w terminologii GA nazywamy rozwiązaniami), które zapewniają populacji (zbiorowi rozwiązań) przetrwanie. Ponieważ zazwyczaj w przypadku GA gwarantujemy stałą wielkość populacji – populacja nie ulega wymarciu na skutek złego przystosowania – wymagamy tego, by była coraz lepsza – zatem, jest to niejako problem optymalizacji funkcji oceny populacji. Ogólnie od algorytmu chcemy by znalazł w akceptowalnym czasie optimum globalne. Są to niestety wykluczające się warunki, więc musimy odpowiednio wyważać pomiędzy szybkością osiągania optimum a zakresem przeglądania przestrzeni rozwiązań, co wiąże się z koniecznością pozostawiania słabo przystosowanych osobników, by osłabiali dążenie do lokalnych optimum. W przypadku GA sterujemy tym na drodze eksperymentów ze współczynnikami – prawdopodobieństwem krzyżowania i mutacji. O ile krzyżowanie w założeniu ma produkować coraz to bardziej dostosowane osobniki (na przestrzeni kilku pokoleń), o tyle mutacja zapewnia nam ten losowy czynnik, który może “zepsuć” nam osobnika, co jednak może doprowadzić do zwiększenia się obszaru poszukiwań, co z kolei może (ale nie musi) doprowadzić nas do innego optimum (być może globalnego). Niestety sterowanie tymi parametrami nie jest jedynym determinantem sukcesu. Główny wpływ na rozwój populacji ma przede wszystkim sposób dokonywania selekcji naturalnej – co wiąże się bezpośrednio ze sposobem zaprezentowania pojedynczego rozwiązania jako osobnika oraz zdefiniowaniem sposobów wyżej wymienionych operacji na nim (krzyżowanie i mutacja). Mogą zatem już na samym początku pojawić się problemy. Natura podsyła nam prostsze rozwiązania.

1.2 Metoda

Wyżarzanie jest pojęciem [S2] metalurgicznym. W sytuacji gdy atomy w kawałku metalu są rozmieszczone przypadkowo, metal jest kruchy i łamliwy. W procesie wyżarzania metal podgrzewa się do wysokiej temperatury powodując gwałtowne wstrząsanie atomami. Jeżeli będzie on nagle ochłodzony to mikrostruktura zostanie uwięziona w niestabilnym stanie przypadkowym. Jeśli jednak ochładzanie będzie powolne to wraz ze spadkiem temperatury atomy będą tworzyły regularne układy, dość stabilne w danej temperaturze. Podczas tego procesu może się zdarzyć, że temperatura chwilowo podniesie się na skutek reorganizacji struktury krystalicznej. Algorytm symulowanego wyżarzania bazuje na algorytmie zaproponowanym przez Metropolis [S8] w celu znalezienia stanu równowagi energetycznej zestawu atomów przy danej temperaturze. Powiązanie pomiędzy tym algorytmem a matematyczną minimalizacją zostało poraz pierwszy odnotowane przez Pincusa [S9], natomiast Kirkpatrick [S4] położył podwaliny pod budowę nowej techniki optymalizacyjnej do rozwiązywania kombinatorycznych problemów.

Ogólnie symulowane wyżarzanie (SA) jest algorytmem z rodziny Generuj i Testuj [S1] (jest zatem heurystyczny, tzn nie gwarantuje osiągnięcia celu) i jest niejako modyfikacją algorytmu Hill Climbing, która dopuszcza możliwość wycofania się w kierunku gorszego rozwiązania, co z kolei stwarza możliwość przeszukania większej przestrzeni rozwiązań – co może doprowadzić do opuszczenia lokalnego i skierowanie się w kierunku globalnego minimum, do którego dotarcie jest najbardziej pożądane. Przy symulowaniu tego zjawiska wprowadzamy pojęcie temperatury jako pewnej funkcji czasu. Jest to model ingerencji w proces wyżarzania poprzez schładzanie (przyspieszanie) lub ogrzewanie. Jak wspomnieliśmy wcześniej, dopuszczamy chwilowy wzrost stanu energetycznego. Wprowadzamy zatem pojęcie zmiany stanu energetycznego. Prawdopodobieństwo przejścia od jednego stanu do drugiego wyrażamy wzorem:

$$p = e^{\frac{-\Delta E}{kT}} \quad (1.2.1)$$

ΔE	– różnica energetyczna przy przejściu ze stanu do stanu
T	– aktualna temperatura
CS (<i>CurrentState</i>)	– aktualny stan energetyczny
NS (<i>NextState</i>)	– nowy stan
f : Temperatura x Czas \rightarrow Temperatura	– funkcja zmniejszająca temperaturę w czasie
t	– czas
Tk	– temperatura końcowa

Współczynnik k oznacza stałą Boltzmann, którą tutaj możemy pominąć (przyjąć jako 1).
Oto algorytm:

```

1.  $T \leftarrow T_p$  // ustal temperaturę początkową
2.  $CS \leftarrow \{ \text{generuj losowo stan lub wyznacz inną metodą} \}$ 
3. FOR  $t \leftarrow 1$  TO  $\infty$ 
4.      $T \leftarrow f(T, t)$  // modyfikuj temperaturę zgodnie z upływem czasu
5.     IF  $T \leq Tk$  THEN
6.         RETURN  $CS$  // osiągnięto stan stabilny, więc zakończ
7.     ENDIF
8.      $NS \leftarrow \{ \text{generuj losowo nowy stan} \}$ 
9.      $\Delta E \leftarrow E(NS) - E(CS)$  // oblicz różnicę
10.    IF  $\Delta E \leq 0$  THEN // jeżeli wylosowaliśmy lepszy stan niż obecny
11.         $CS \leftarrow NS$  // będzie on naszym nowym stanem odniesienia
12.    ELSE
13.         $CS \leftarrow NS$  WITH PROBABILITY  $p = \exp(-\Delta E / T)$ 
14.    ENDIF
15. ENDFOR

```

Ważne jest aby wybrać (najlepiej empirycznie) odpowiednio wysoką początkową, odpowiednią końcową temperaturę oraz funkcję f modelującą ochładzanie. Zbyt szybkie ochładzanie z dużym prawdopodobieństwem doprowadzi do osiągnięcia minimum lokalnego – czego chcemy unikać. W porównaniu do innych nieliniowych algorytmów optymalizacyjnych, SA jest względnie łatwe w implementacji, co w połączeniu ze statystyczną gwarancją odnalezienia optimum – czyni tę metodę atrakcyjną. Spowodowało to wiele prób implementacji, które to z kolei ujawniły pewne wady tej metody oraz wprowadziły jej sprawienia. Zaobserwowano standardowe wyżarzanie Boltzmann (1.2.4) i stwierdzono, że może być to czasochłonne. Ponadto problemem może być [S10] dostosowanie parametrów do specyfiki rozwiązywanego problemu.

Autor [S3] proponuje mnożenie temperatury przez stałą obliczoną ze wzoru:

$$c = \exp(\ln(Tk/Tp)/(nT-1)) \quad (1.2.2)$$

gdzie nT oznacza liczbę temperatur. Można oczywiście mnożyć T przez dowolną stałą mniejszą od 1 lub wykorzystać parametr czasu do obliczania na bieżąco. Przy szybkim wyżarzaniu Cauchy'ego stosuje się [S11]:

$$T \leftarrow T/t \quad (1.2.3)$$

Pojęcie “standardowe” Boltzmana zakłada logarytmiczny krok:

$$T \leftarrow T/\ln(t) \quad (1.2.4)$$

Lester Ingber w trakcie badań nad symulowanym wyżarzaniem opracował kilka innych podejść przyspieszających działanie algorytmu – techniki adaptacyjne [S12] a także wykładnicze zmniejszanie temperatury:

$$T \leftarrow T \exp(-c t^{1/D}) \quad (1.2.5)$$

gdzie D jest rozmiarem przestrzeni parametrów [S11]. To ostatnie podejście empirycznie okazało się nie gorsze od wyżej wymienionych dwóch technik.

W zastosowaniu modelu wyżarzania wygodniej jest operować na pojęciach, które stosujemy w przypadku algorytmów genetycznych. Interesujące nas rozwiązanie przedstawiamy w postaci osobnika, który tutaj odpowiada stanowi energetycznemu, w którym z kolei znajduje się metal. Z osobnikiem związana jest funkcja przystosowania, której minimalizacja nas interesuje. Warto nadmienić, że wybór nowego stanu (osobnika, rozwiązania) może być mniej lub bardziej losowy. Dopuszcza się kilka pojęć:

- losowanie stanu spośród całej przestrzeni rozwiązań – może prowadzić do dużych fluktuacji wyników – nietrzymanie się minimów lokalnych, może jednak powodować wiele pustych przebiegów w pętli z uwagi na szybko malejące prawdopodobieństwo trafienia - niespełnienia warunku (wiersz nr 10 pseudokodu) – a więc wpływa na wydłużenie czasu przeszukiwania.
- jeżeli rozwiązanie zakodujemy w pewnej postaci (np. zbiór cech), to możemy wygenerować następnego osobnika poprzez zmianę losowo wybranej cechy w obecnym osobniku (bezwartunkowy odpowiednik mutacji w GA). Może to być np. zmiana losowego bitu [S6] w reprezentacji osobnika – nie daje to z pewnością takich możliwości w przeszukiwaniu przestrzeni rozwiązań jak powyższe podejście – co wcale nie oznacza że jest gorsze. Jak zwykle w dziedzinie algorytmów wzorowanych na naturze, wszystko uzależnione jest od rodzaju problemu oraz doboru parametrów algorytmu.

2. Automaty komórkowe

2.1 Historia

Pojęcie automatu komórkowego (CA) poraz pierwszy pojawiło się w latach 40-tych XX wieku za sprawą Stanisława Ulama. Był on uczniem wybitnego polskiego matematyka Stefana Banacha. Przed wybuchem II Wojny Światowej wyjechał on jednak [C1] do Stanów Zjednoczonych, do Los Alamos (we Lwowie nie mógł znaleźć pracy ani zdobyć tytułu profesora ponieważ był Żydem, Stanach Zjednoczonych był m.in współtwórcą bomby wodorowej). Stało się to za namową Jonasa von Neumanna, który kilkakrotnie przybywał do Lwowa i Banachowi jak i jemu proponował pracę za oceanem. Ulam zainspirował von Neumanna swoim pomysłem zamodelowania dyskretnych kwadratowych komórek w przestrzeni dwuwymiarowej, które mogą znajdować się w dwóch stanach: włączony lub wyłączony, a

stan determinowany jest przez reguły sąsiedztwa. [C3] Ulam za ich pomocą generował złożone figury geometryczne o periodycznej strukturze. W tym czasie von Neumann, polegając na pracy Turinga, pracował nad modelem automatu samo-replikującego się - "pierwotnej zupy" - cieczy z której miało powstać życie. Ulam zasugerował, by użyć do tego przestrzeni komórkowych, a więc zdyskretyzować przestrzeń oraz jak się okazało również i czas. Pozwalało to uniknąć błędów zaokrągleń, a przy okazji otwierało nowe możliwości. W 1970 roku John H. Conway opublikował najsłynniejszy chyba na świecie automat komórkowy – Life. Można patrzeć na niego jak na żywe środowisko. Stosuję się w nim terminologię żywych i martwych komórek. Swego czasu istniały kluby fanów tego automatu. Pozornie wygląda to na zabawkę, jednak nie zapominajmy że jest to zabawka o podstawach matematycznych, zdolna do każdej operacji logicznej. Na początku lat 80-tych XX wieku za sprawą Stephena Wolframa, automaty komórkowe zaczęły być dostrzegane przez fizyków. Wolfram próbował użyć ich do sklasyfikowania procesów dynamicznych. Ponieważ za CA kryje się formalizm matematyczny, zatem można teoretycznie za ich pomocą zamodelować różne procesy dynamiczne. Zatem jeżeli udałoby się sklasyfikować automaty – pojawiłaby się możliwość dokonania klasyfikacji procesów w terminologii automatów (nieśmiało nasuwa się analogia do algorytmu i teorii maszyny Turinga, (*przyp. autorski*)). Niestety dyskretny charakter automatu komórkowego wyklucza to zastosowanie, przede wszystkim ze względu na chęć zamodelowania ciągłych procesów przez fizyków (niemożliwość przetłumaczenia ich na równania różniczkowe). Natomiast idealny automat do testowania twierdzeń propabilistycznych musiałby mieć nieskończoną sieć komórek, co z kolei jest nie do przyjęcia przez informatyków. Mimo tych niepowodzeń i pewnego rozczarowania, automaty komórkowe znalazły swoje zastosowanie w dziedzinie szeroko rozumianej symulacji, co pozwoliło część zjawisk fizycznych przenieść za pośrednictwem CA do innych dziedzin.

2.2 Definicja

Automat komórkowy jest pojęciem matematycznym, zatem związany jest z nim pewien formalizm. CA jest opisany przez :

- sieć komórek $\{ i \}$ przestrzeni D -wymiarowej
- zbiór stanów $\{ s_i \}$ pojedynczej komórki, zazwyczaj taki sam dla każdej komórki, zawiera k elementów
- reguła F określająca w chwili $t + 1$ stan komórki zależny od stanu tej komórki oraz jej otoczenia w chwili t . Funkcję tę można określić ogólnie $s_i(t + 1) = F(\{ s_j(t) \})$, gdzie j należy do otoczenia i -tej komórki.

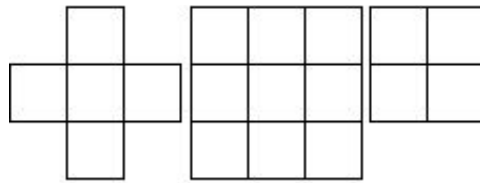
Jest to definicja automatu *deterministycznego*, jeżeli funkcja F zależy dodatkowo od pewnej zmiennej losowej, automat komórkowy jest *propabilistyczny*. Z automatem komórkowym związać możemy jeszcze jeden parametr – r – promień sąsiedztwa.

Często stosowaną notacją służącą określeniu całej rodziny automatów jest podanie pary liczb (k, r) , ilości stanów oraz promienia otoczenia. Wadą tej metody jest brak określenia wymiaru przestrzeni, mimo tego stosuje się tą metodę – głównie do automatów jedno- i dwuwymiarowych. CA mimo swojej prostoty, nie są wcale łatwe do ogarnięcia, ze względu na ogrom swoich możliwości (np. trudno je sklasyfikować), dlatego też zazwyczaj nie wychodzi się poza trzeci wymiar – wchodzimy wtedy w wymiar abstrakcyjny – trudny do wyobrażenia sobie w kategoriach symulacji rzeczywistości (aczkolwiek pozostaje nam czas, jednak jest on już parametrem reguły automatu, mimo to wykorzystujemy czas jako dodatkowy wymiar, zwłaszcza przy graficznej ilustracji działania jednowymiarowych automatów).

Do tej pory przemilczeliśmy kwestię sąsiedztwa. W przypadku przestrzeni dwuwymiarowej wyróżnić można kilka jego rodzajów, a oto wybrane z nich [C3]:

- sąsiedztwo von Neumanna – rozważamy tylko sąsiadów, z którymi mamy wspólne boki
- sąsiedztwo Moore'a – z którymi dzielimy wierzchołki.
- sąsiedztwo Margolusa – rozważamy grupę komórek 2×2 , użyte do symulacji zachowania gazów

Ilustrują je poniższe rysunki (odpowienio: von Neumanna, Moore'a, Margolusa):



Rys. 2.2.1. Sąsiedztwo von Neumanna, Moore'a, Margolusa.

Automaty komórkowe funkcjonują w oparciu o trzy fundamentalne zasady [C3]:

- równoległość - procesy podczas ewolucji systemu zachodzą jednocześnie. Wszystkie komórki są uaktualniane niezależnie od siebie
- lokalność – na stan komórki wpływ ma jedynie stan jej samej oraz stan jej najbliższego otoczenia
- homogeniczność – wszystkie reguły są uniwersalne, obowiązują w całej przestrzeni komórek.

2.3 Życie jest najprostszym przykładem

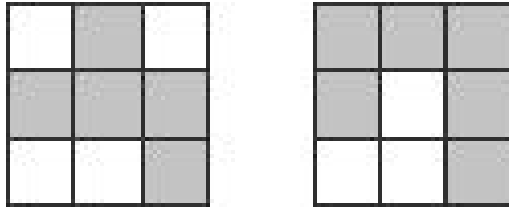
...i chyba najlepszym. Automat komórkowy *Life* opracowany przez Conway'a jest dwuwymiarowym automatem z rodziny $(2, 1)$. On właśnie wprowadza terminologię żyjącej (znajdującej się w stanie 1) oraz martwej komórki (stan 0) – zatem jak widać komórka może znajdować się w $k = 2$ stanach oraz ogranicza się do najbliższego sąsiedztwa ($r = 1$), które tutaj jest sąsiedztwem Moore'a.

Reguły określające stan pojedynczej komórki w czasie $t + 1$ są następujące:

- martwa komórka otoczona przez trzy żywe, staje się żywa (“rodzi się”)
- żywa komórka w sąsiedztwie dwóch lub trzech żywych komórek nie zmienia swojego stanu, zatem pozostaje żywa
- w każdym innym przypadku komórka pozostaje martwa lub umiera

Widzimy tutaj pewną analogię do rzeczywistości, zwłaszcza gdy wyobrazimy sobie sytuację gdy mamy do czynienia ze skończonymi zasobami pewnego dobra – bardzo ogólnie – osobniki łączą się w większe grupy by zapewnić sobie większą szansę dostępu do zasobu, tworzenie opłaca się do momentu, gdy grupa jest zbyt duża, powiedzmy że powstają konflikty, podziały i krąg się zamyka (jednakże zakładanie ograniczoności zasobu prowadzi do różnych dziwnych wniosków. Znany jest paradoks: w pewnej gminie mieszka 10 gospodarzy, każdy z nich posiada jedną krowę, która codziennie daje 10 litrów mleka. Przy założeniu, że zasoby trawy są skończone, ilość mleka pobierana od wszystkich krow wynosi $10 \times 10 = 100$ litrów. Jeżeli jeden z gospodarzy postanowi mieć więcej mleka i kupi drugą krowę, wtedy łączna ilość mleka dostarczana przez wszystkie krowy będzie wynosić $11 \times 9 = 99$, przy czym gospodarz posiadający dwie krowy będzie mieć $2 \times 9 = 18$ litrów mleka, pozostali będą mieli po 9 litrów. Zmniejszyła się zatem ilość mleka przypadająca na gospodarzy posiadających tylko jedną krowę. Jeżeli zaobserwowanie coraz mniejszych ilości pobieranego mleka spowoduje w tych gospodarzach chęć zakupu drugiej i kolejnej krowy – spowoduje to ostatecznie śmierć wszystkich krow z powodu braku trawy. (*przyp. autorski*)).

Wracając do automatów komórkowych, następny rysunek ilustruje przykładową sytuację automatu *Life*, przy przestrzeni ograniczonej do rozmiaru 3×3 komórki. Kolor szary oznacza, że komórka jest aktywna.



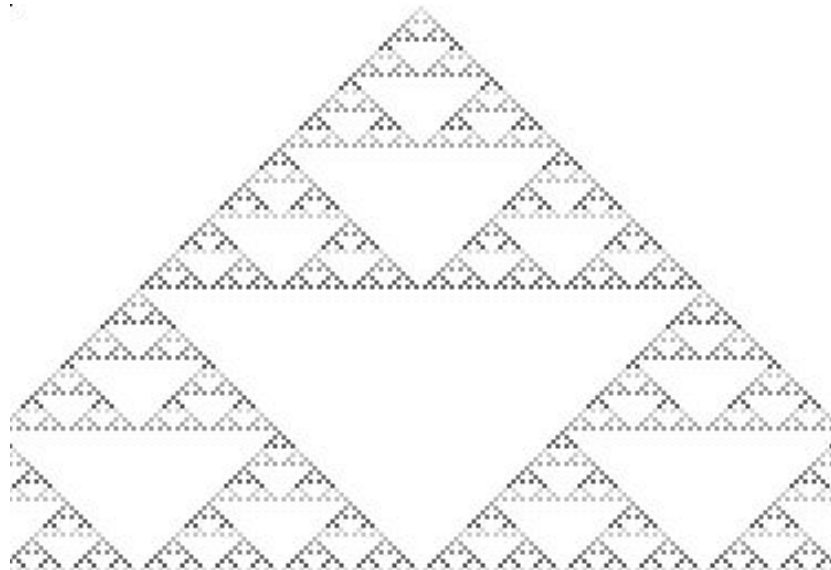
Rys. 2.3.1. Przykładowa sytuacja w czasie t , oraz w czasie $t + 1$ w automacie komórkowym *Life*.

Widać jakie możliwości jakie daje nam ten stosunkowo prosty, dwumiarowy, dwustanowy automat i możemy sobie tylko wyobrazić co jeszcze można osiągnąć poprzez ustalenie innych reguł jego funkcjonowania, wprowadzenie większej ilości stanów, zmianę sposobu postrzegania sąsiadów, np. poprzez zmianę promienia sąsiedztwa r , a przede wszystkim przez zwiększenie liczby wymiarów.

Najlepiej poznaną rodziną jest rodzina automatów jednowymiarowych $(2, 1)$. Możemy policzyć, ile automatów zawiera ta rodzina – jak się okaże – będzie to wskazówką do ich definiowania. Ponieważ funkcja F określająca stan s_i komórki i w chwili $t + 1$ przyjmuje trzy argumenty (stan komórki s_i oraz stany komórek sąsiadujących s_{i-1}, s_{i+1}), przy czym każdy stan może przyjmować $k = 2$ wartości – mamy $2^3 = 8$ różnych konfiguracji. Dla każdej konfiguracji funkcja F może określić stan komórki na $k = 2$ sposoby, z czego mam $2^8 = 256$ różnych zestawów reguł a więc 256 różnych automatów komórkowych. Może być to pewną wskazówką, by zakodować reguły w postaci binarnej (ilość automatów jest wszakże potęgą liczby 2). Przykład najlepiej zilustruje przyjętą konwencję:

111 110 101 100 011 010 001 000
 0 1 0 1 0 1 1 0 = 90

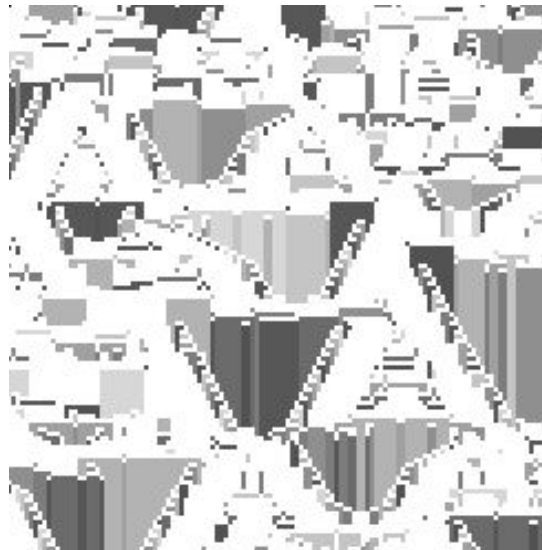
Przykładowo zapis 100 jest opisem sytuacji, jaka może wystąpić w pewnej chwili t , oznacza że lewy sąsiad rozważanej komórki jest aktywny, podczas gdy ona sama jak i jej prawy sąsiad są nieaktywne. Dla tej sytuacji reguła określa wartość 1, a więc orzeka, że ta komórka w chwili $t + 1$ stanie się aktywna. Jest to jednowymiarowy odpowiednik przytoczonej wcześniej gry w życie – automatu *Life*. Można taki automat zapisać skrótowo jako 90. Wizualizacja życia takiego automatu nie jest jednak ciekawa – jego komórki żyją w końcu w przestrzeni jednowymiarowej. Jeżeli pokusimy się o dodanie wymiaru – czasu, efekt będzie zaskakujący:



Rys. 2.3.2. Dywan Sierpińskiego – fraktalna forma uzyskana przez automat 90, a także 12.

Na powyższym rysunku oś czasu skierowana jest w dół. Bardzo łatwo jest prześledzić rozwój komórek przy pomocy rysunku. Spójrzmy na pierwszy rząd – zawiera on tylko jedna żywą komórkę. W następnym

kroku ta komórka umrze, gdyż jej sąsiedzi są martwi – zatem zachodzi sytuacja 010. Natomiast z punktu widzenia lewego sąsiada (pamiętajmy o homogeniczności) zachodzi sytuacja 001, prawego – 100. Te komórki “ożyją” w następnym kroku. Oto inny przykład jednowymiarowego automatu – wielu widzi tu łąkę pełną tulipanów.

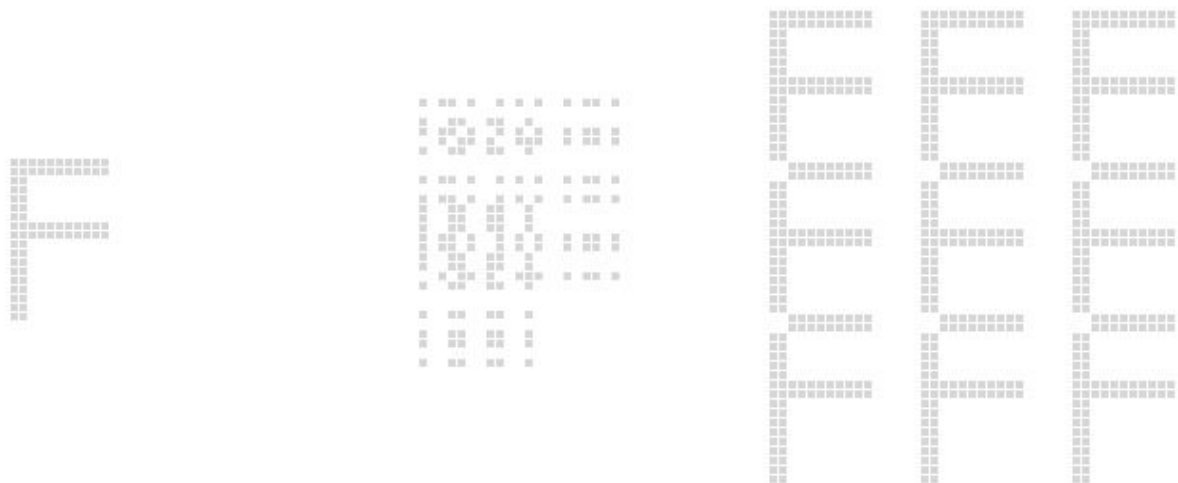


Rys. 2.3.3. Łan tulipanów wygenerowany przez głoszący automat jednowymiarowy.

Automat komórkowy 90 (*Life 1D*) polega na regule nieparzystości sąsiedztwa. Istnieją oczywiście automaty, w których regułę ustala suma stanów sąsiedztwa, a nie same ich konfiguracje. Automaty takie nazywamy [C4] głoszącymi lub liczącymi.

Innym, bardziej spektakularnym automatem głoszącym (w tym przypadku dwuwymiarowym) jest ten opracowany przez Edwarda Fredkina [C5]. Wcześniej wspomnieliśmy o zdolności automatów do samo-replikacji.

Automat, w którym wykorzystano regułę Fredkina, replikuje dowolny prosty zbiór komórek:



Rys. 2.3.4. Samo-replikujący się automat z regułą Fredkina: na początku, po czterech oraz po ośmiu krokach.

Reguła ta mówi [C3], że replikacja następuje tylko wtedy, gdy suma stanów sąsiadów jest nieparzysta.

2.4 Zastosowanie

Automaty komórkowe ze względu na prostotę implementacji zyskały sobie popularność, a dzięki możliwościom – stały się bardzo dobrym narzędziem do symulacji zjawisk fizycznych. Zastosowania CA są liczne i różnorodne. W istocie, CA formułują [C3] zupełnie nowe wszechświaty. Nasz Wszechświat podlega prawom fizyki, które to są tylko częściowo znane i wydają się być bardzo złożone. W automatach komórkowych prawa są proste i całkowicie poznane. Można zatem testować i analizować globalne zachowanie uproszczonej przestrzeni – np. przeprowadzając symulację zachowania gazu, który przecież składa się z molekuł, których zachowanie zależy właśnie od najbliższego sąsiedztwa. CA wykorzystuje się m.in. w:

- badaniu ferromagnetyzmu według modelu Isinga [C6], który to reprezentuje materiał jako sieć, w której każdy węzeł jest w określonym stanie magnetycznym. Stan ten – w tym przypadku jedna z dwóch orientacji spinów pewnych elektronów – zależy od stanu węzłów sąsiadujących
- symulacji procesu perkolacji [C7]
- symulacji rozprzestrzeniania się pożaru [C8]
- jako alternatywę do równań różnicowych [C9]
- symulacji procesów krystalizacyjnych [C7]
- symulacji i badania rozwoju
- symulacji formowania się opinii publicznej [C7]

i w wielu innych dziedzinach życia.

Bibliografia

- [S1] T. J. Kruk, J. Wrzesień *Korelacja w wykrywaniu anomalii* 2003
- [S2] A. Podleśny <http://www.sssg.ae.krakow.pl/sssgdaw/a99/teksty/podlesny.htm>
- [S3] (autor nieznan) *Przeszukiwanie przestrzeni stanów - algorytmy heurystyczne*
- [S4] S. Kirkpatrick, C. D. Gelatt, Jr. M. P. Vecchi *Optimization by Simulated Annealing* 1983
- [S5] K. Habowski D. Kamiński *Szeregowanie zadań w linii produkcyjnej, metodą symulowanego wyżarzania* 2001
- [S6] H. Kwaśnicka *Optymalizacja, idea algorytmu genetycznego, GA, HillClimbing i symulowane wyżarzanie przykład* 2004
- [S7] B. T. Luke <http://members.aol.com/btluke/metro01.htm>
- [S8] N. Metropolis, A. W. Rosenbluth, M. Rosenbluth M. A. H. Teller, E. Teller *Equations of State Calculations by Fast Computing Machines* 1953
- [S9] M. Pincus *A Monte Carlo Method for the Approximate Solution of Certain Types of Constrained Optimization Problems* 1970
- [S10] L. Ingber *Simulated annealing: Practice versus theory* 1993
- [S11] L. Ingber *Very Fast Simulated Re-Annealing* 1989
- [S12] L. Ingber *Adaptive simulated annealing (ASA): Lessons learned* 1995
- [S13] J. Karczmarczuk *Ruletka dla fizyka (wprowadzenie do technik Monte-Carlo)* 2003
- [C1] <http://www.republika.pl/stako/banach/banach.html>
- [C2] <http://www.zyvex.com/nanotech/selfRepJBIS.html#vonNeumannArchitecture>
- [C3] Jean-Philippe Rennard *Introduction to Cellular automata* 2000
- [C4] K. Kułakowski *Automaty komórkowe* 1998
- [C5] <http://www.bottomlayer.com/bottom/finite-all.html#CA>
- [C6] <http://www.plmsc.psu.edu/~www/mat597c-1997/phases/Lecture3/node3.html>
- [C7] K. Malarz *Automaty komórkowe* 2003
- [C8] A. Muzy, G. Wainer E. Innocenti, A. Aiello, J.-F. Santucci *COMPARING SIMULATION METHODS FOR FIRE SPREADING ACROSS A FUEL BED* 2002
- [C9] T. Toffoli *Cellular automata as an alternative to Differential equations, in Modelling Physics* 1984
- [C10] M. Alberti P. Waddell *An Integrated Urban Development and Ecological Simulation Model* 2000