



Physarum

Slime mould computing simulations

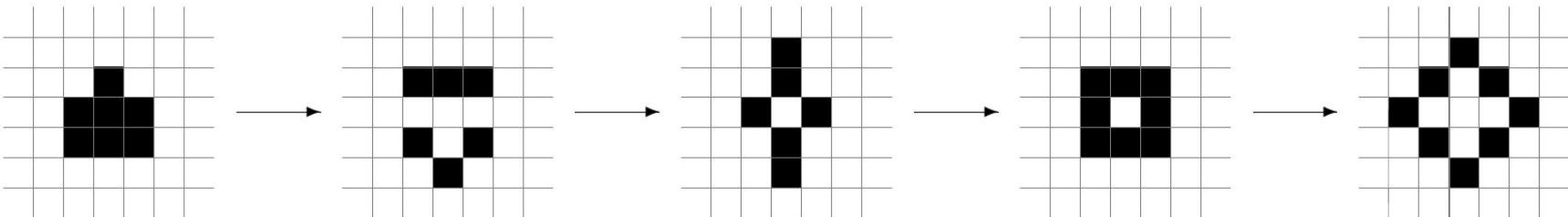
Outline del lavoro

- Automa Cellulare come modello matematico
- Physarum Polycephalum
- Esperimenti in laboratorio
- Simulazioni con automi cellulari
- Framework di implementazione
- Modello corrente nella comunità scientifica
- Modello sperimentale
- Simulazioni con entrambi i modelli
- Analisi dei risultati e validazione



Automi cellulari

- Modello matematico sviluppato da Stanislaw Ulam e John Von Neumann
- Utilizzato per descrivere l'evoluzione di sistemi complessi discreti
- Rappresentato come una griglia costituita da un numero finito di celle
- Cambiamento simultaneo delle celle con regole locali
- Noto esempio: The Game of Life



Physarum Polycephalum

- Protista melmoso unicellulare
- Ciclo di vita:
 - Fase vegetativa: massa citoplasmatica di colore giallo (plasmodio)
 - Esplorazione ambientale organizzata
 - Formazione di tubi protoplasmatici
 - Fase dormiente: tessuto essiccato resistente (sclerozio)
- Organismo intelligente in presenza di condizioni favorevoli

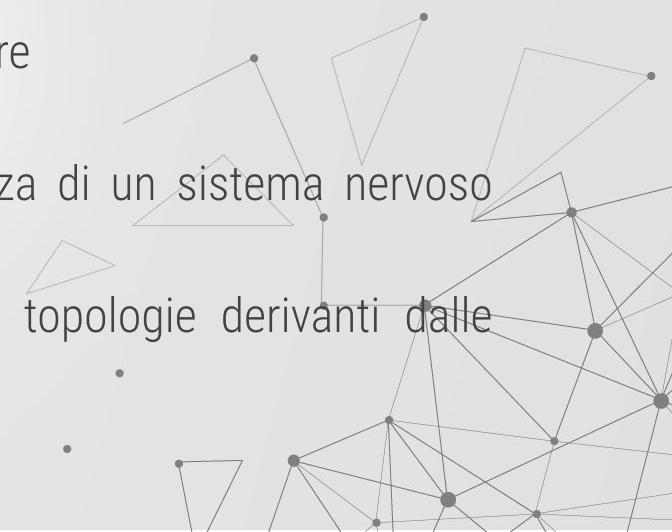


Esperimenti biologici

- Intelligenza sperimentata nell'ambito dei problemi di ottimizzazione
- Capacità di costruire reti efficienti mostrando comportamento adattivo e pattern complessi
- Lavori rilevanti:
 - Maze solving (Nakagaki et. al): la muffa riesce a trovare il percorso ottimale tra due punti in un labirinto
 - Network formation (Tero et. al): la muffa mostra risultati qualitativamente comparabili alla rete ferroviaria di Tokyo

Simulazioni in ambito informatico

- Utili per studiare fenomeni biologici:
 - Senza dover disporre di un laboratorio
 - Senza possedere le conoscenze tecniche
 - In maniera sempre riproducibile
 - Con una velocità di esecuzione molto maggiore
- Automi Cellulari per simulare la Physarum
 - Le regole locali corrispondono alla mancanza di un sistema nervoso centrale della muffa
 - Physarum mostra comportamenti globali e topologie derivanti dalle azioni locali



Implementazione

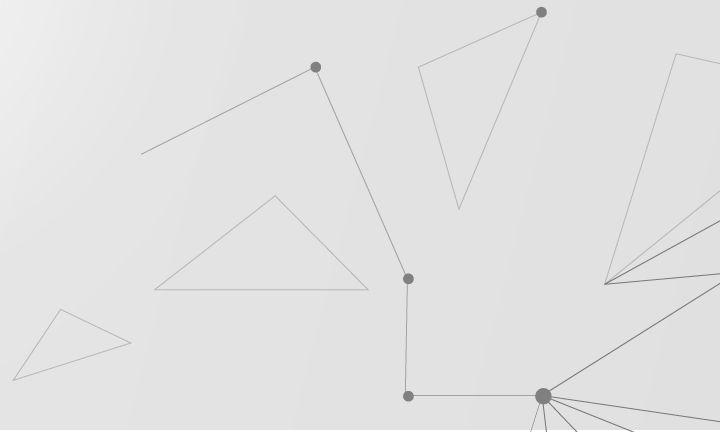
Strumenti:

- Git
- LaTeX
- Unity game engine -> WebGL/webassembly
- VueJS + Vuetify

Punti essenziali

- Encoding mappa
- Comunicazione bidirezionale VueJS <-> Unity
- TileMap
- Color Shading

○

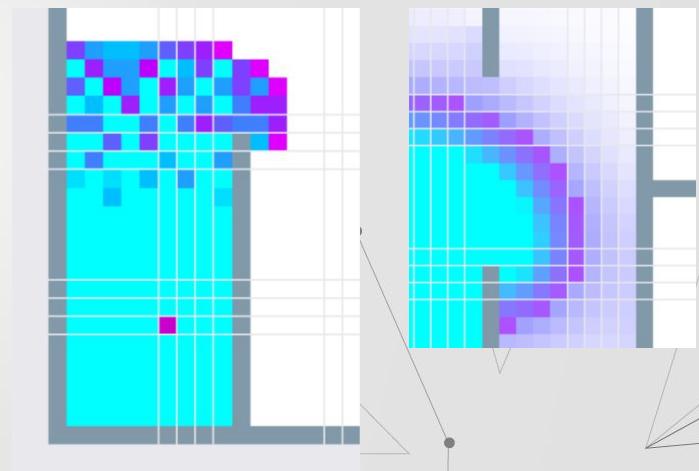


Color Shading

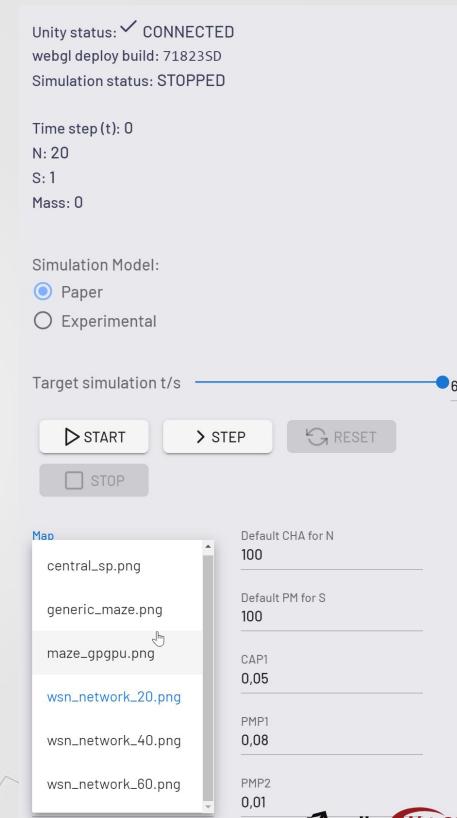
- Prescaling
- Linear Interpolation
- Additional expanding shading



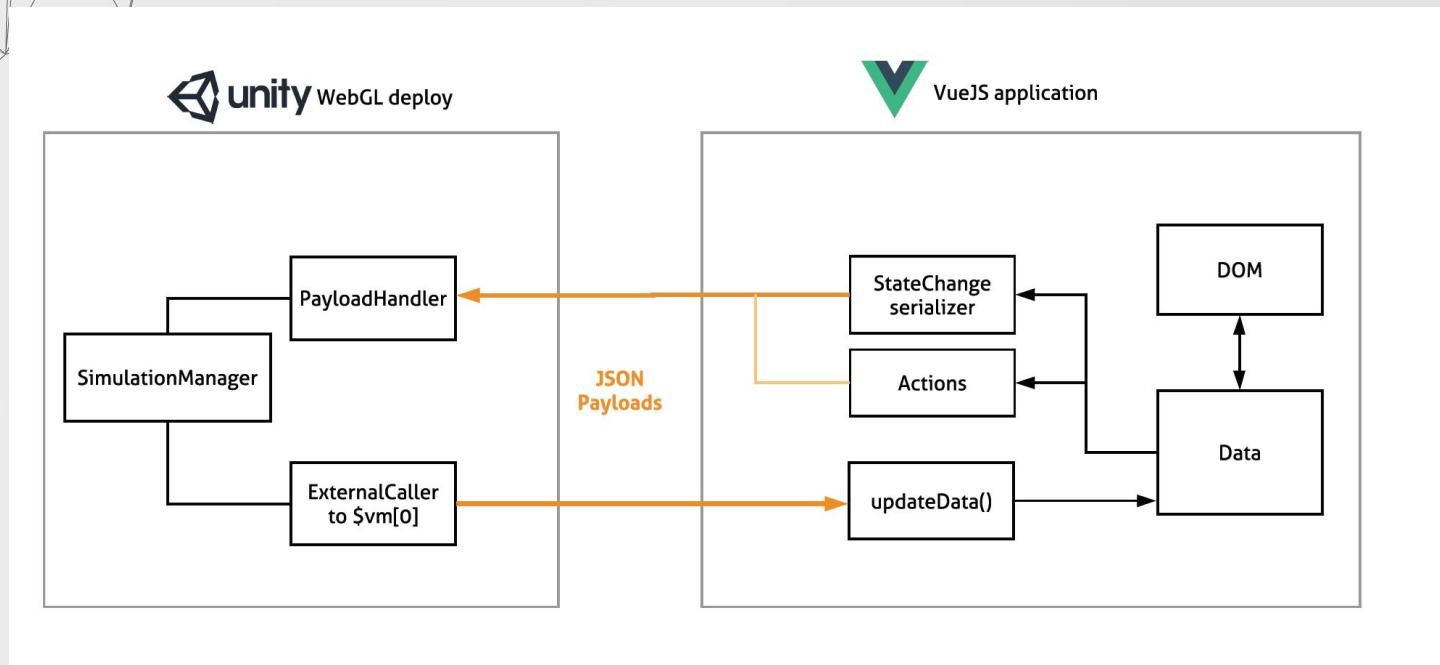
- Cell mass is scaled with a factor of 50 (40 on low-mass cells). $PM = PM/50$
 - On low-mass cells ($PM < 15$): $C = (1 - PM, 1 - PM, 1)$
 - $C = (1 - PM, PM, 3)$
- Experimental Model:
- Empty cells are white.
 - Cell mass is scaled with a factor of 10. $PM = PM/10$
 - $C = ((0.8 - PM) * 1.25, (PM - 0.20) * 1.25, 1)$



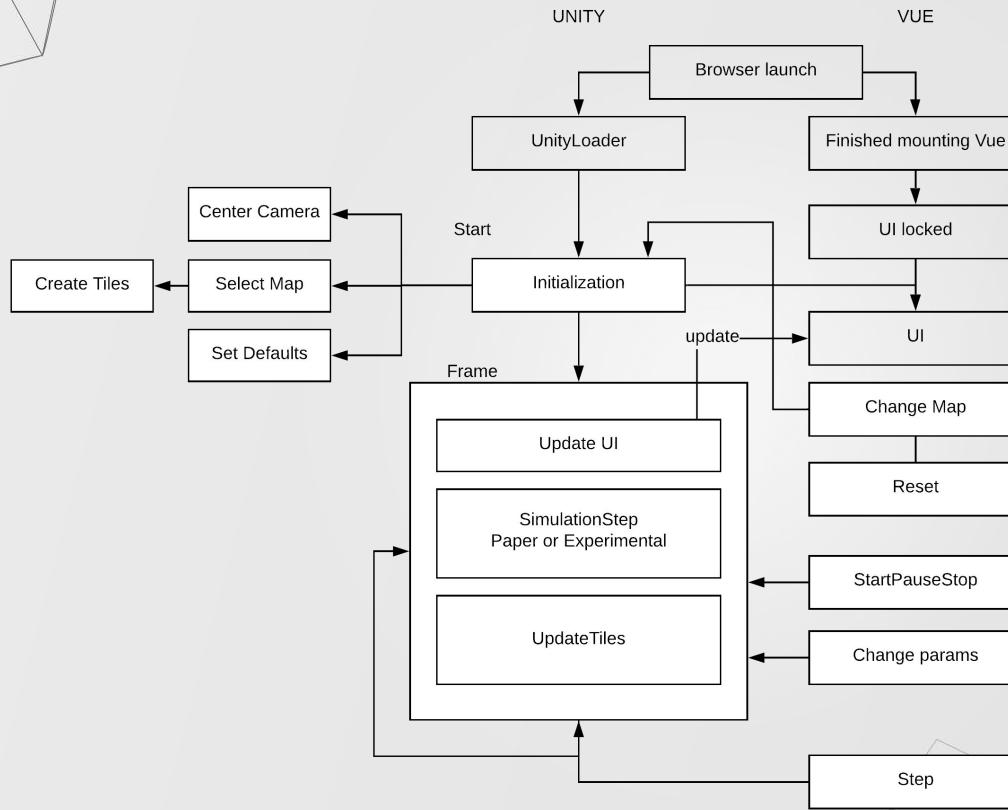
Implementazione

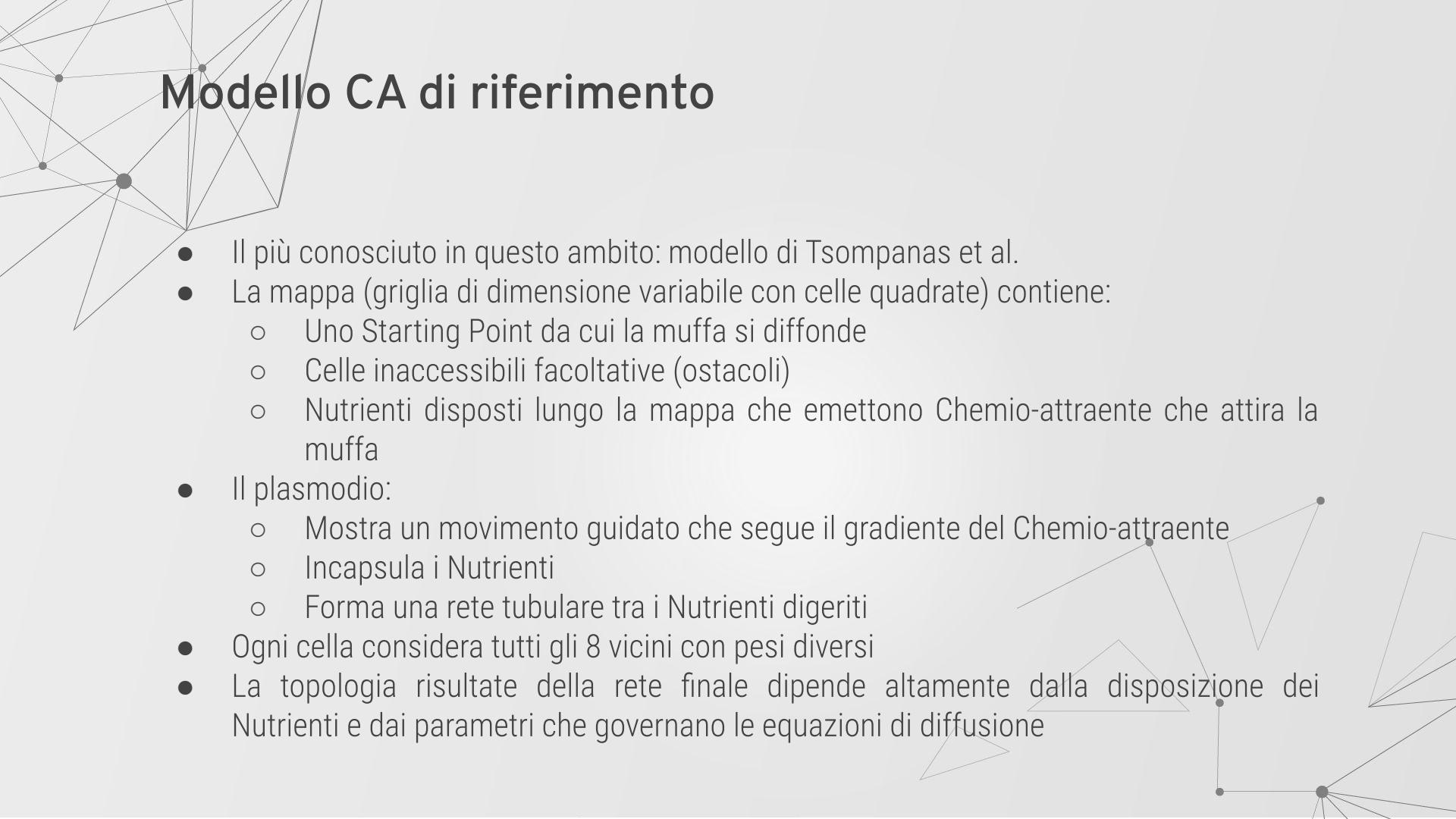


VueJS \leftrightarrow Unity using PayloadHandler



Simulation lifecycle events





Modello CA di riferimento

- Il più conosciuto in questo ambito: modello di Tsompanas et al.
- La mappa (griglia di dimensione variabile con celle quadrate) contiene:
 - Uno Starting Point da cui la muffa si diffonde
 - Celle inaccessibili facoltative (ostacoli)
 - Nutrimenti disposti lungo la mappa che emettono Chemio-attraente che attira la muffa
- Il plasmodio:
 - Mostra un movimento guidato che segue il gradiente del Chemio-attraente
 - Incapsula i Nutrimenti
 - Forma una rete tubolare tra i Nutrimenti digeriti
- Ogni cella considera tutti gli 8 vicini con pesi diversi
- La topologia risultante della rete finale dipende altamente dalla disposizione dei Nutrimenti e dai parametri che governano le equazioni di diffusione

Stato locale

Stato della cella (i,j) al tempo t:

$$S T_{(i,j)}^t = [AA_{(i,j)}, PM_{(i,j)}^t, CHA_{(i,j)}^t, TE_{(i,j)}^t]$$

Dove:

- AA: indica se la cella è libera o è un ostacolo (available)
- PM: valore della massa della muffa nella cella corrente
- CHA: valore del Chemio-attraente nella cella
- TE: indica se la cella fa parte di una formazione tubulare



Parametri iniziali ed equazioni di diffusione

Ad inizio simulazione:

$$PM_{(i,j)}^t = \begin{cases} 100, & \forall i, j : c_{(i,j)} \in S \\ 0, & \text{else} \end{cases}$$

$$CHA_{(i,j)}^t = \begin{cases} 100, & \forall i, j : c_{(i,j)} \in N \\ 0, & \text{else} \end{cases}$$

Equazione di diffusione PM:

$$PM_{(i,j)}^{t+1} = PM_{(i,j)}^t + PMP1 \times [PMvNN_{(i,j)}^t + PMP2 \times PMeMN_{(i,j)}^t]$$

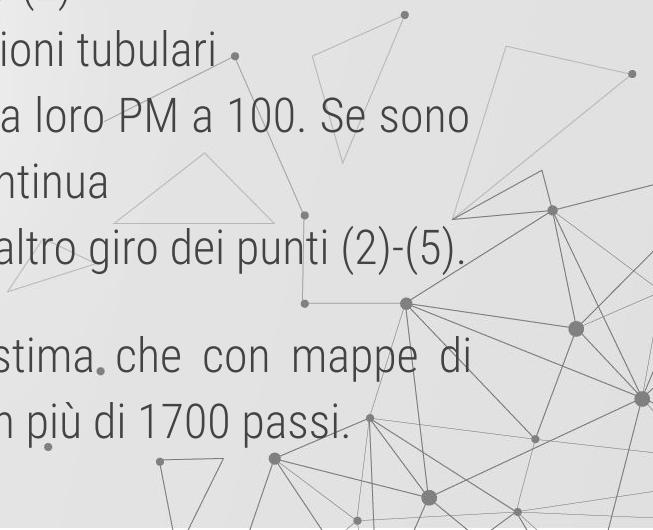
Equazione di diffusione CHA:

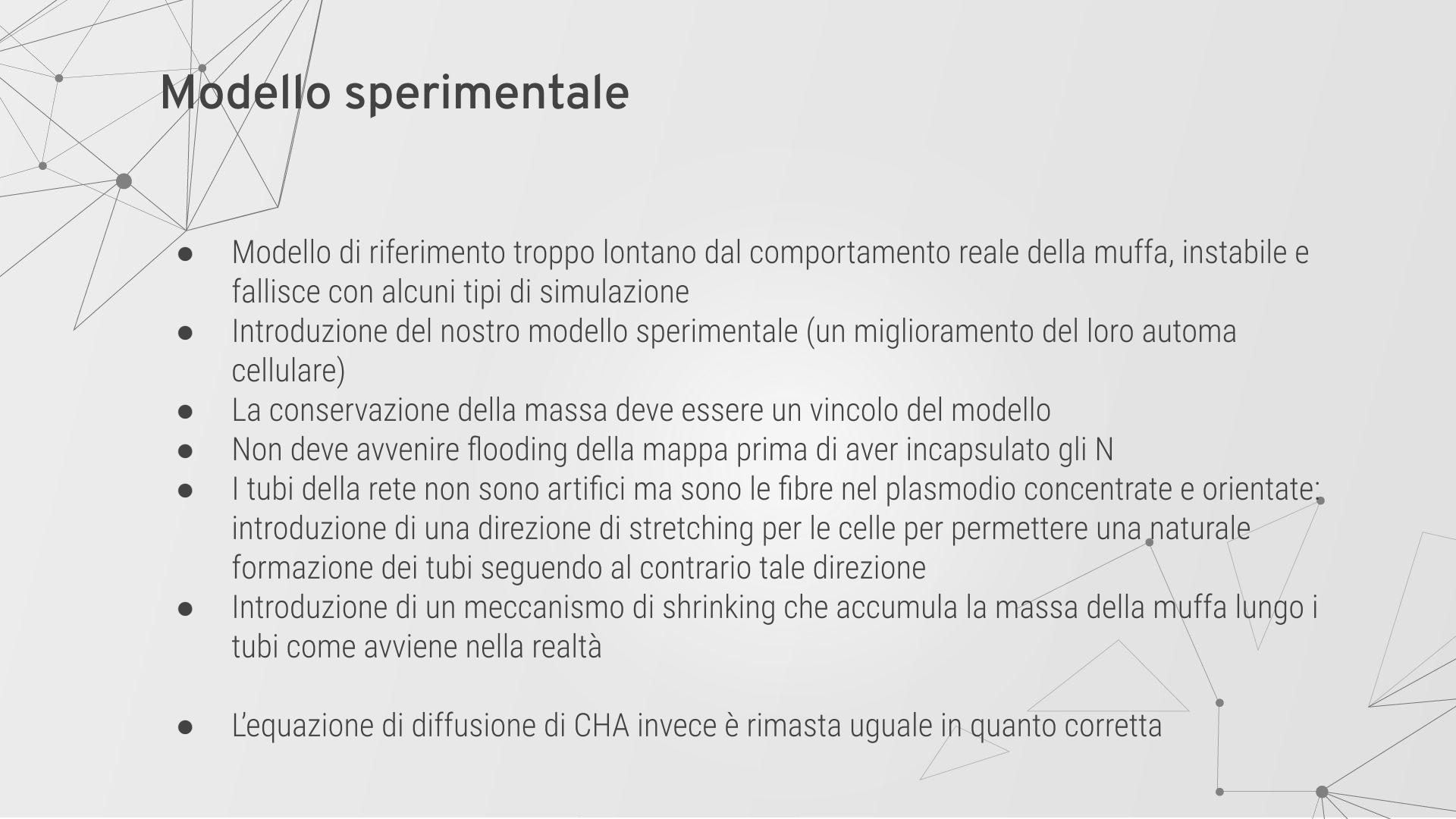
$$CHA_{(i,j)}^{t+1} = CON \times CHA_{(i,j)}^t + CAP1 \times (CHAvNN_{(i,j)}^t + CAP2 \times CHAeMN_{(i,j)}^t)$$

Algoritmo a 6 step

1. Inizializza il modello
2. Applica le equazioni di diffusione per 50 ticks
3. Controlla se ci sono N encapsulati con almeno una massa uguale a ThPM. Se ce n'è almeno uno continua, altrimenti torna al punto (2)
4. Tutti gli N encapsulati vengono connessi con formazioni tubulari.
5. Tutti gli N di (4) vengono trasformati in S settando la loro PM a 100. Se sono passati meno di 5000 ticks torna a (2), altrimenti continua
6. Trasforma tutti gli S in N tranne il penultimo. Fai un altro giro dei punti (2)-(5).

Il punto 6 di fatto non viene mai eseguito perché si stima che con mappe di dimensione 75 x 75 raramente si raggiungono tutti gli N in più di 1700 passi.





Modello sperimentale

- Modello di riferimento troppo lontano dal comportamento reale della muffa, instabile e fallisce con alcuni tipi di simulazione
- Introduzione del nostro modello sperimentale (un miglioramento del loro automa cellulare)
- La conservazione della massa deve essere un vincolo del modello
- Non deve avvenire flooding della mappa prima di aver incapsulato gli N
- I tubi della rete non sono artifici ma sono le fibre nel plasmodio concentrate e orientate: introduzione di una direzione di stretching per le celle per permettere una naturale formazione dei tubi seguendo al contrario tale direzione
- Introduzione di un meccanismo di shrinking che accumula la massa della muffa lungo i tubi come avviene nella realtà
- L'equazione di diffusione di CHA invece è rimasta uguale in quanto corretta

Stato locale

Stato della cella (i,j) al tempo t :

$$ST_{(i,j)}^t = [AA_{(i,j)}, PM_{(i,j)}^t, CHA_{(i,j)}^t, TE_{(i,j)}^t, Dir_{(i,j)}^t, Age_{(i,j)}^t]$$

Dove le variabili sono le stesse del primo modello eccetto:

- Dir: direzione dello stretching delle fibre contenute nella cella
- Age: età (numero di tick) che sono passati dalla prima volta che la massa ha raggiunto la cella



Parametri iniziali ed equazioni di diffusione

Ad inizio simulazione S ottiene PM a nostra discrezione, gli N ottengono un valore di CHA come nel modello di Tsompanas et al.

Direzione di stretching:

$$DirectionOfStretch_{(i,j)}^{t+1} = \begin{cases} None, & PM_{(i,j)}^t = 0 \\ MaxPMNeighbourRelativePosition, & \text{else} \end{cases}$$

where $MaxPMNeighbourRelativePosition$ = North if north's neighbour has the maximum PM and so on.

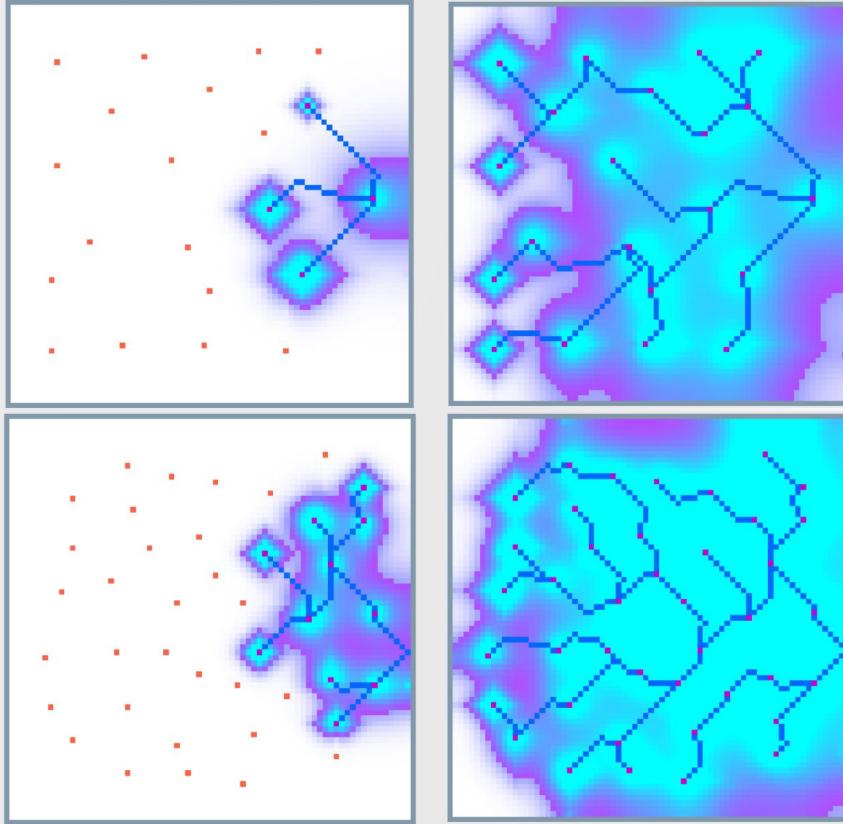
Equazione di diffusione PM:

$$PM_{(i,j)}^{t+1} = \begin{cases} PM_{(i,j)}^t + PMvnuct_{(i,j)}^t + PMmuct_{(i,j)}^t, & CHA_{(x,y)}^t == 0 \text{ and } CHA_{(i,j)}^t == 0 \\ PM_{(i,j)}^t + PMvngct_{(i,j)}^t + PMmgct_{(i,j)}^t, & \text{else} \end{cases}$$

Equazione di diffusione CHA:

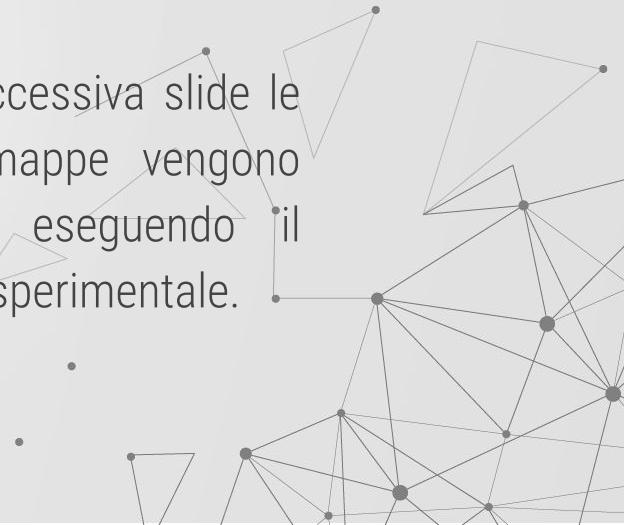
$$CHA_{(i,j)}^{t+1} = CON \times CHA_{(i,j)}^t + CAP1 \times (CHAvNN_{(i,j)}^t + CAP \times CHAeMN_{(i,j)}^t)$$

Simulazione paper - Network Formation

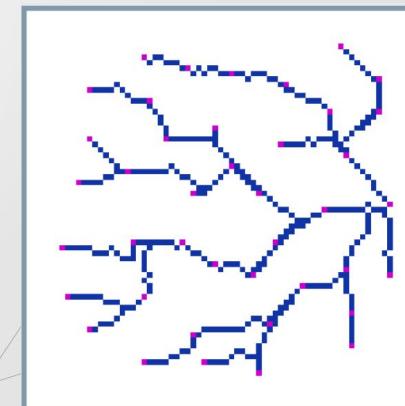
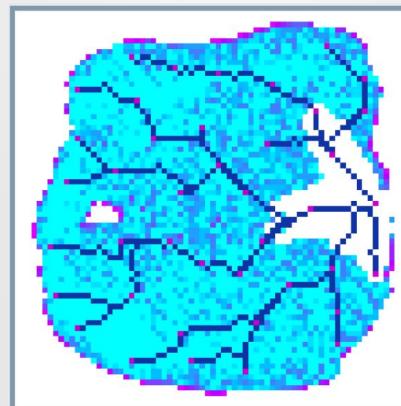
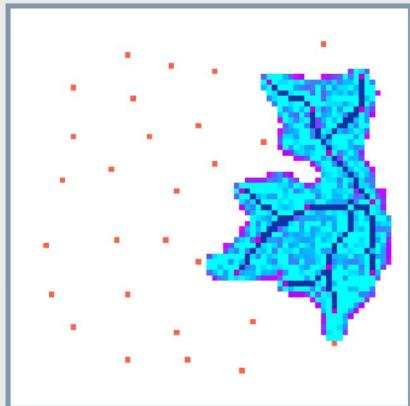
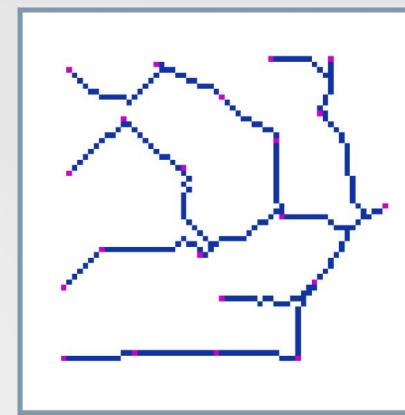
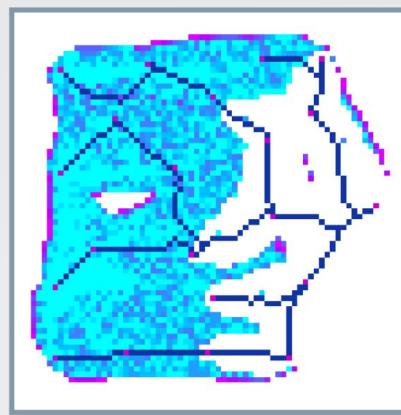


Simulazione del modello del paper per mappe raffiguranti una rete e composte da 20 e 40 nodi.

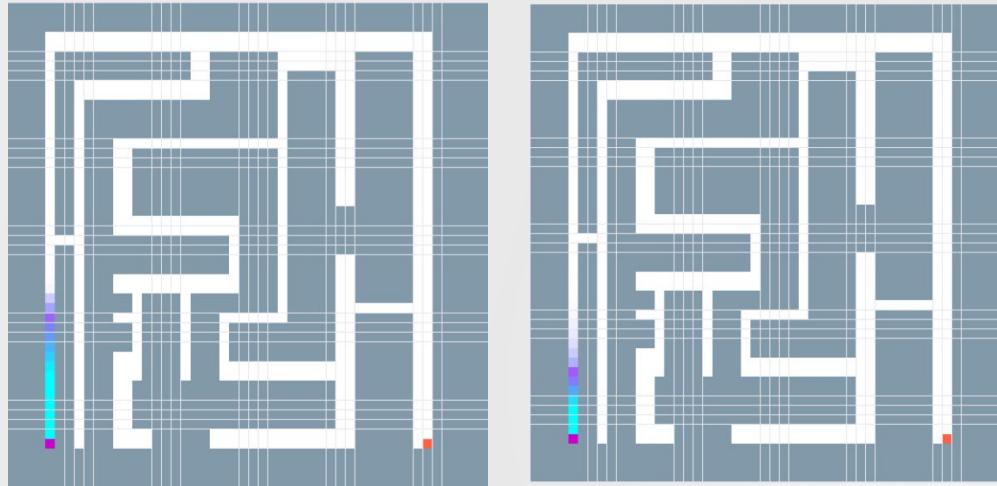
Nella successiva slide le stesse mappe vengono simulate eseguendo il modello sperimentale.



Simulazione sperimentale - Network Formation



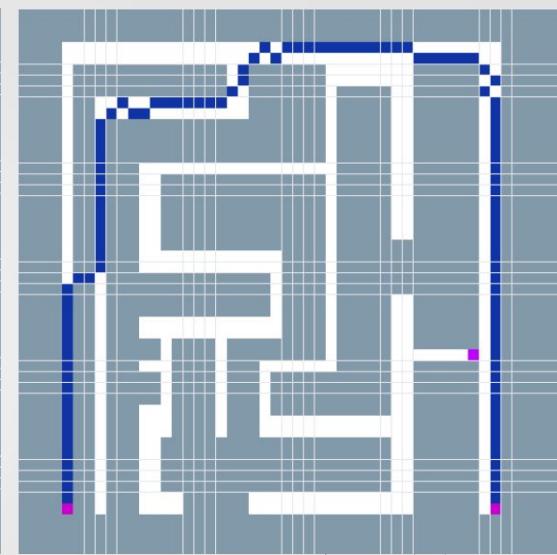
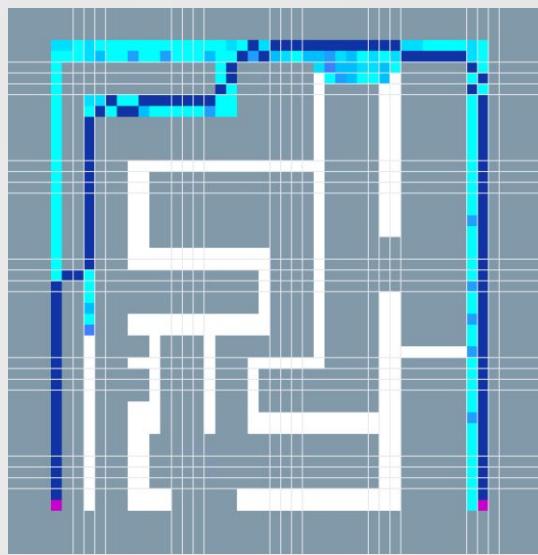
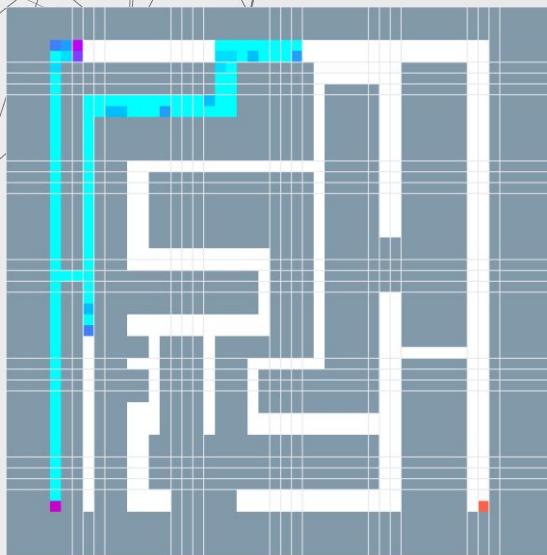
Simulazione paper - Maze solving



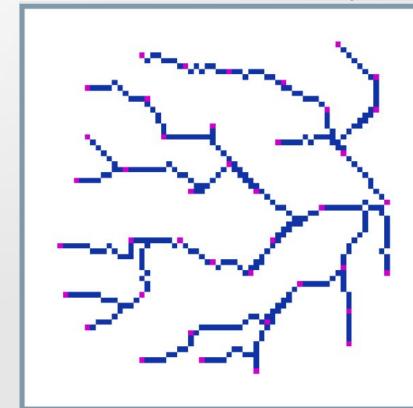
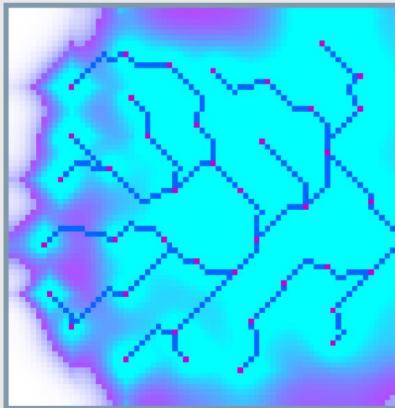
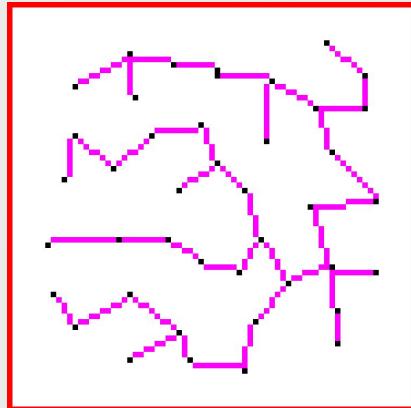
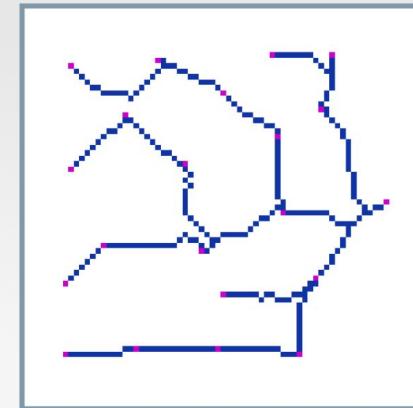
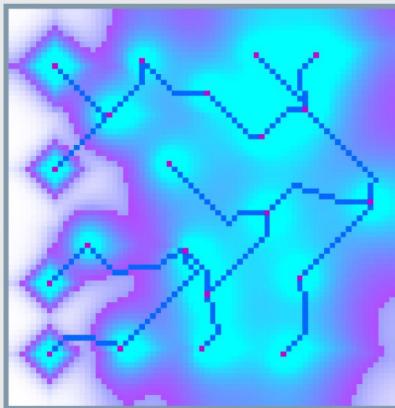
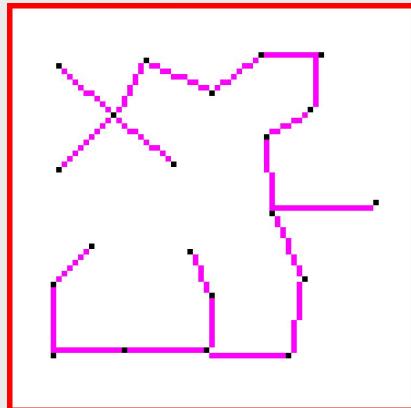
Simulazione del modello del paper per la mappa raffigurante un labirinto con S e N esattamente ai lati opposti.

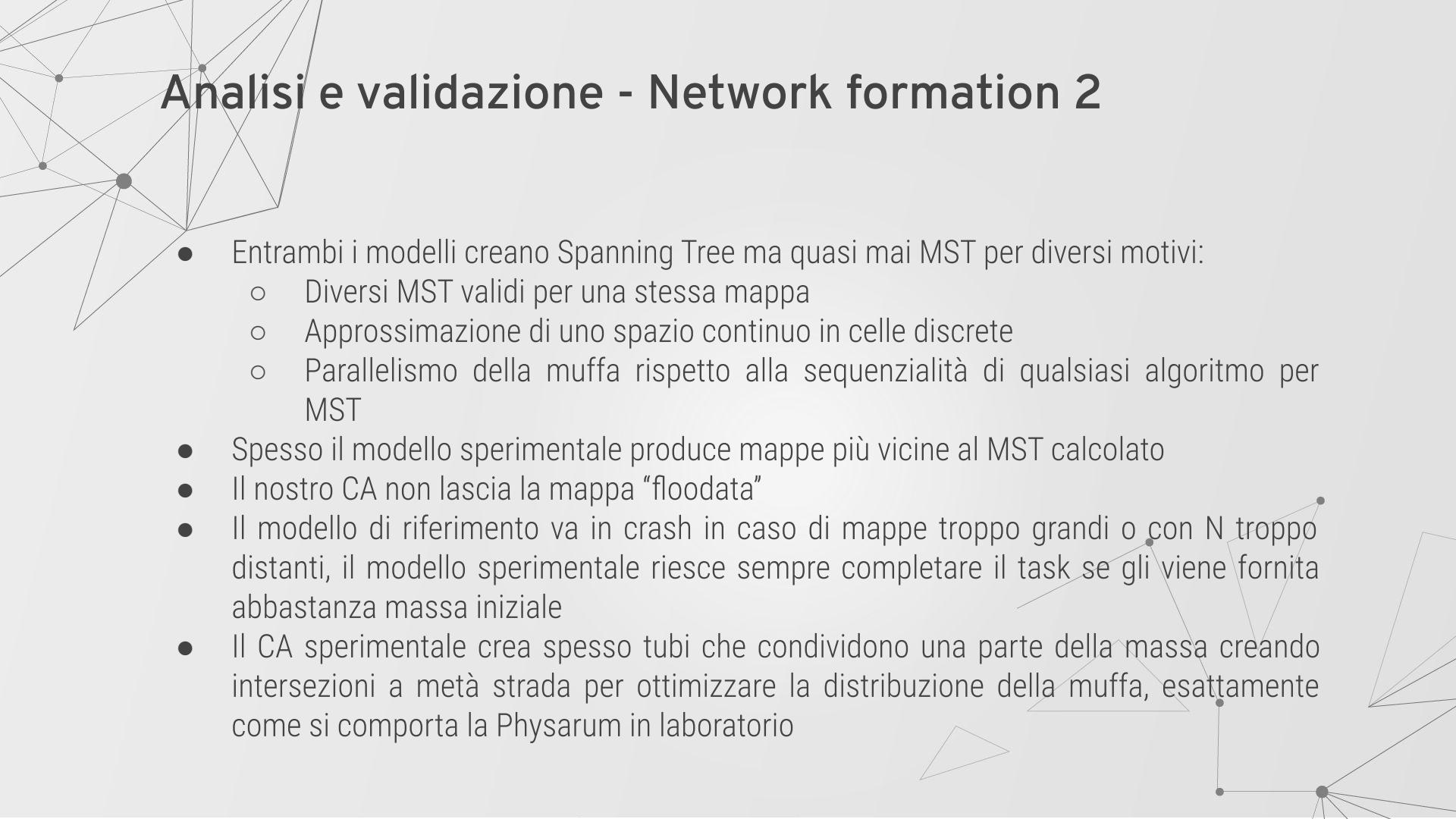
Nella successiva slide lo stesso labirinto è utilizzato per la simulazione del modello sperimentale.

Simulazione sperimentale - Maze Solving



Analisi e validazione - Network formation 1



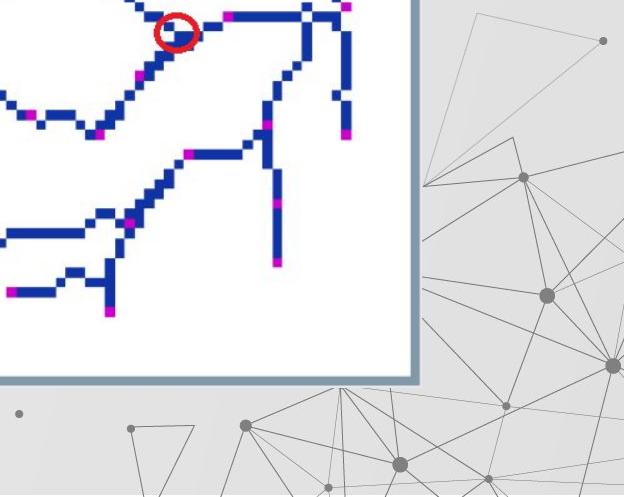
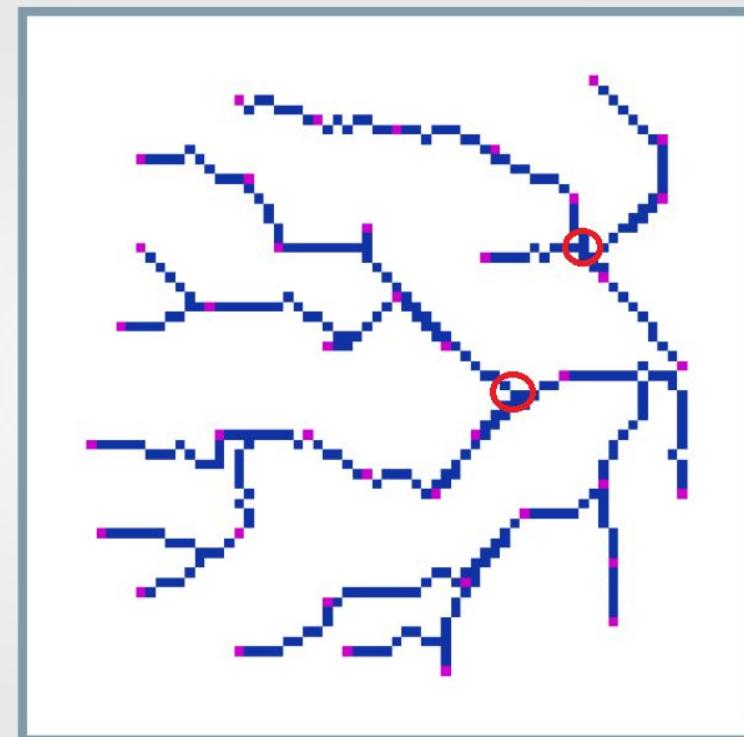
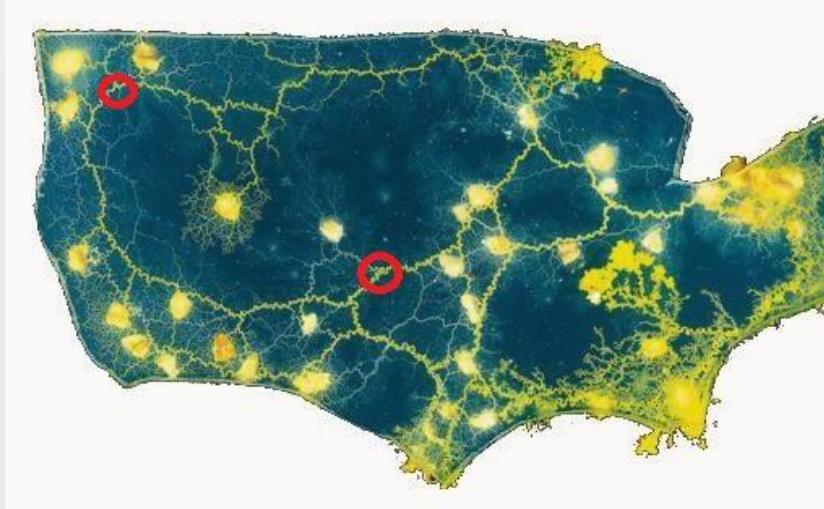


Analisi e validazione - Network formation 2

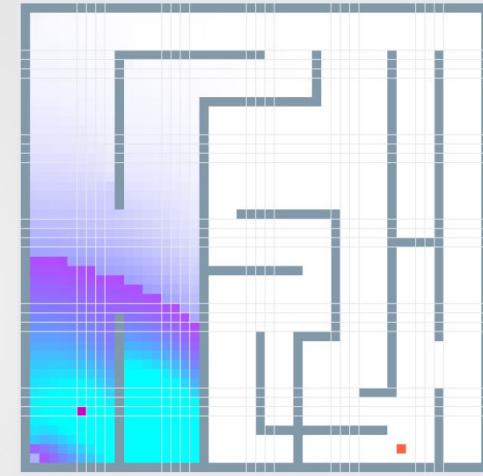
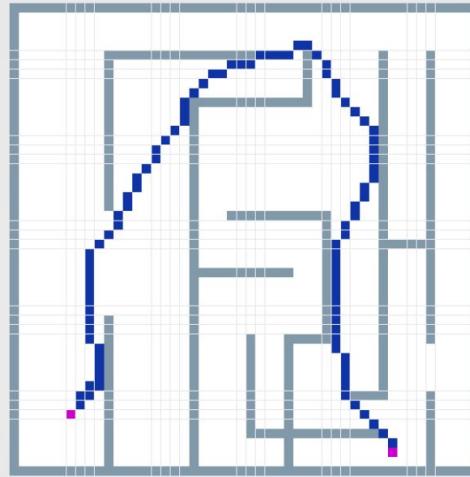
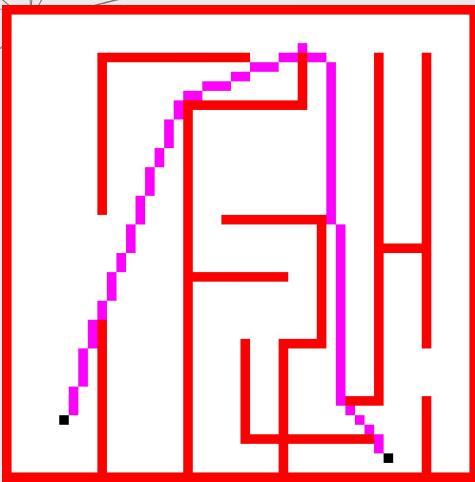
- Entrambi i modelli creano Spanning Tree ma quasi mai MST per diversi motivi:
 - Diversi MST validi per una stessa mappa
 - Approssimazione di uno spazio continuo in celle discrete
 - Parallelismo della muffa rispetto alla sequenzialità di qualsiasi algoritmo per MST
- Spesso il modello sperimentale produce mappe più vicine al MST calcolato
- Il nostro CA non lascia la mappa "floodata"
- Il modello di riferimento va in crash in caso di mappe troppo grandi o con N troppo distanti, il modello sperimentale riesce sempre completare il task se gli viene fornita abbastanza massa iniziale
- Il CA sperimentale crea spesso tubi che condividono una parte della massa creando intersezioni a metà strada per ottimizzare la distribuzione della muffa, esattamente come si comporta la Physarum in laboratorio

Analisi e validazione - Network formation 3

Intersezioni a metà strada tra due o più formazioni tubulari
nella vera Physarum e nel modello sperimentale

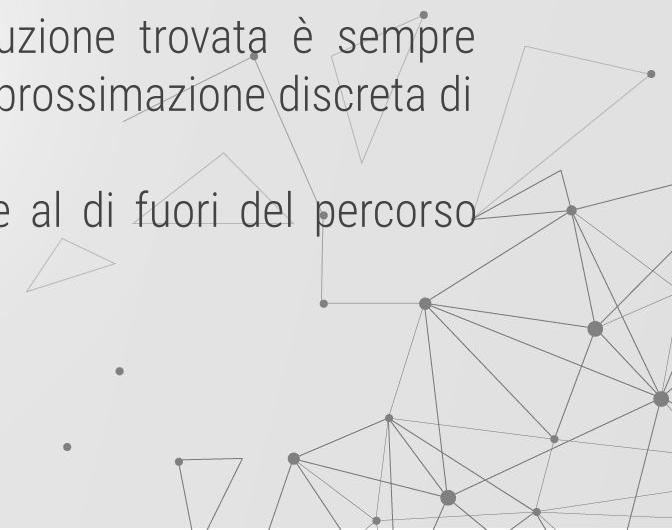


Analisi e validazione - Maze Solving 1



Analisi e validazione - Maze Solving 2

- Il modello di riferimento del paper di Tsompanas et al. non riesce a risolvere alcun labirinto proposto dalla letteratura scientifica
- Il modello sperimentale, quando assegnata abbastanza massa iniziale, risolve sempre i labirinti trovando il cammino più breve tra tutti quelli possibili
- Da un punto di vista delle singole celle la soluzione trovata è sempre subottima con un errore trascurabile dovuto all'approssimazione discreta di uno spazio continuo
- Il modello sperimentale non lascia massa inutile al di fuori del percorso minimo



Conclusioni

- Criticità ed eccessiva approssimazione del modello di Tsompanas et al.
- Validazione robusta del nostro modello sperimentale, ulteriori indagini con esempi di laboratorio sono opportune
- Framework sviluppato per veloce deployment e tuning dei parametri

Sviluppi futuri

- Funzione di shrinking da invertire
- Aggiunta di funzionalità quando più S vengono inseriti nella stessa simulazione