

Pierfrancesco Benanti

**UTILIZZO DELL'ANT COLONY SYSTEM PER
L'OTTIMIZZAZIONE DI UN SERVIZIO FEEDER DEL
TRASPORTO COLLETTIVO**

Tesi di laurea in Ingegneria Civile e Ambientale

Relatore:

Prof. Ing. Giuseppe Inturri

Correlatore:

Prof. Alessandro Pluchino

Anno Accademico 2017-18

INDICE

Introduzione	3
Capitolo 1.....	4
METODOLOGIA APPLICATA	4
1. Cenni sulle linee guida per l'analisi ed il progetto di una Rete di Trasporto Collettivo ..	4
1.1. Determinazione della rete di base	5
1.2. Analisi della distribuzione spaziale degli spostamenti	5
1.3. Ubicazione dei capolinea.....	5
1.4. Programma di esercizio (Scheduling).....	6
2. Servizi di Feederaggio nelle aree a domanda debole: Il Vehicle Routing Problem	6
2.1. Clienti.....	10
2.3. Veicoli.....	10
2.4. Formulazione Matematica.....	11
3. Una soluzione al VRP: Gli algoritmi Euristici	12
Capitolo 2.....	15
CASO STUDIO	15
1. Ant Colony System: Un algoritmo innovativo.....	15
2. Applicazione progettuale	21
3. Calcolo del percorso migliore in una rete di trasporto collettivo tramite ACS:	
Descrizione programma	24
4. Simulazioni e Risultati	35
Conclusioni	57
Bibliografia.....	58

INTRODUZIONE

In generale il sistema dei trasporti, a servizio della mobilità di persone e dello spostamento di merci, è una delle fondamentali esigenze da soddisfare in un contesto socio-economico urbano.

Uno dei tipici problemi nell'ambito dell'ingegneria dei trasporti è la progettazione di una rete di trasporto collettivo (network design), che rappresenta l'attività che porta alla definizione del servizio da offrire in termini di topologia della rete di trasporto stessa e frequenze da realizzare. Solitamente le informazioni di pianificazione devono raggiungere un livello di dettaglio tale da consentire ipotesi attendibili di costi e benefici derivanti dall'attuazione del servizio pianificato.

Principalmente a un servizio di trasporto collettivo è demandato il compito di essere accessibile e l'accessibilità è uno dei motivi principali dei molteplici studi attuati. Un trasporto collettivo, se accessibile e di buona qualità, può essere considerato una valida alternativa per la domanda di mobilità che, specialmente in Italia, predilige l'auto come principale modo di trasporto, specialmente se si tiene conto della sempre crescente congestione dei flussi di traffico, dei problemi di parcheggio e dell'inquinamento atmosferico.

La recente legislazione in materia di trasporti ha posto particolare attenzione al ruolo dei sistemi di trasporto collettivo emanando una serie di disposizioni, che riguardano principalmente le aree rurali dove spesso si verifica una carenza di servizi, che sono fondamentali per una vita civile da garantire per tutti. Tali aree vengono chiamate "aree a domanda debole".

Il comma 4 dell'articolo 14 del Decreto Legislativo 19 novembre 1997 n. 422 recita che "Per l'esercizio dei servizi pubblici di trasporto locale in territori a domanda debole, al fine di garantire il soddisfacimento delle esigenze di mobilità nei territori stessi, le regioni possono individuare modalità particolari di espletamento dei servizi di linea. [...]"

La presenza di queste particolari aree territoriali rappresenta un elemento che non può essere ignorato. Da recenti studi, una possibile soluzione è rappresentata da una nuova tipologia di sistema di trasporto collettivo, chiamata DRT (Demand Responsive Transit), che rappresenta la combinazione di sistema di trasporto pubblico tradizionale e sistema di trasporto a domanda individuale, in cui il servizio viene fornito su richiesta degli utenti secondo le loro necessità ed esigenze di mobilità.

Le linee guida per la redazione dei Piani Urbani della Mobilità Sostenibile, contenute nel decreto del Ministero dei Trasporti del 4/8/2017, indicano tra le strategie suggerite

l'introduzione di sistemi di mobilità motorizzata condivisa, quali car-sharing, bike-sharing, van-sharing, car-pooling e l'utilizzo dell'ITS e di sistemi di infomobilità per favorire l'integrazione di sistemi di trasporto, per la fornitura di dati sulla rete prioritaria urbana e per lo sviluppo di servizi innovativi di mobilità.

Le aree a domanda debole in particolare possono beneficiare dei cosiddetti servizi di feederaggio, ovvero una tipologia di trasporto secondario effettuato mediante mezzi di trasporto di dimensioni non rilevanti tra centri di grandi dimensioni ed altri di dimensioni minori, al fine di ridurre i costi del trasporto principale effettuato solo tra i grandi centri con mezzi di grandi dimensioni.

Il problema della progettazione dei servizi di feederaggio rappresenta un motivo di ricerca per l'aumento dell'efficienza di una rete di trasporto collettivo.

Scopo di questa trattazione sarà quello di verificare la capacità di modelli basati sul principio di ottimizzazione della colonia di formiche (o Ant Colony System) come validi strumenti per la progettazione e definizione delle linee di feederaggio del trasporto pubblico.

Capitolo 1

METODOLOGIA APPLICATA

1. Cenni sulle linee guida per l'analisi ed il progetto di una Rete di Trasporto Collettivo

Si definisce Trasporto Collettivo un sistema di trasporto in grado di trasportare più persone, garantendo un servizio accessibile alle esigenze dei vari utenti.

Il Trasporto Collettivo si divide in:

- Pubblico: disponibile per tutti (es. Autobus di Linea);
- Privato: riservato ad alcuni (es. Autobus Noleggiato);
- Individuale: svolto con veicoli in grado di trasportare un numero limitato di persone (es. Taxi, Automobile Personale);

In particolare, una Rete di Trasporto Collettivo è definita dall'insieme di infrastruttura e di servizio fornito su un tracciato fisso da veicoli singoli o convogli in esercizio secondo un programma di orari prefissato.

Esistono diversi requisiti che ci permettono di definire una corretta progettazione e pianificazione di una rete di trasporto collettivo. In questo paragrafo descriveremo i punti principali di cui un progettista deve tenere conto.

1.1. Determinazione della rete di base

In tale fase vengono individuati i rami della rete stradale utilizzabili dalle linee di trasporto collettivo. A partire dal grafo della rete stradale completa si eliminano quei rami che non sono utilizzabili dagli autobus sia per le caratteristiche geometriche (ad esempio per larghezza inferiore ai 3-4 metri, per pendenza eccessiva, per svolte con raggi di curvatura bassi, ecc), sia per vincoli normativi (ad esempio archi appartenenti a zone pedonali).

1.2. Analisi della distribuzione spaziale degli spostamenti

Gli spostamenti tra le zone di traffico dell'area di studio vengono generalmente rappresentati mediante matrici (*matrici o-d*) il cui generico elemento, D_{od} , rappresenta il numero di spostamenti che partono dalla zona o e sono diretti nella zona d .

È opportuno suddividere l'insieme degli spostamenti che interessano l'area di studio in 4 sottoinsiemi costituiti da:

- *spostamenti interni*: con entrambi gli estremi interni all'area di studio;
- *spostamenti di scambio*: dall'interno verso l'esterno dell'area di studio, ovvero col solo estremo iniziale interno all'area di studio;
- *spostamenti di scambio*: dall'esterno verso l'interno dell'area di studio, ovvero con il solo estremo finale nell'area di studio;
- *spostamenti di attraversamento*, ovvero spostamenti con entrambi gli estremi esterni all'area di studio.

1.3. Ubicazione dei capolinea

I capolinea devono essere posizionati in relazione a caratteristiche peculiari della domanda o a dislocazioni ottimali sul territorio. In particolare sono candidati a divenire capolinea tutti i terminal delle autolinee extraurbane (ove esistenti), i terminal ferroviari, marittimi ed aeroportuali (ove esistenti), i parcheggi di interscambio, e le piazze principali dei quartieri periferici. Si deve inoltre, verificare (soprattutto per gli eventuali capolinea dislocati nel centro) che ci sia lo spazio sufficiente per la sosta e la manovra degli autobus.

1.4. Programma di esercizio (Scheduling)

È il risultato finale della procedura di calcolo della frequenza di servizio di ogni linea, del numero di veicoli richiesti, degli orari di partenza e transito di ogni corsa.

Il programma di esercizio di ogni linea deve soddisfare due requisiti fondamentali:

- Fornire un'adeguata capacità di trasporto per il volume di traffico di passeggeri previsto;
- Offrire una frequenza di servizio minimo (ovvero un accettabile tempo di attesa) dal punto di vista del livello di servizio.

Nel prossimo paragrafo verrà trattato un sottoproblema della progettazione di una rete di trasporto collettivo, che coinvolge la definizione di una linea che svolga in modo efficiente il servizio di raccolta e distribuzione dei passeggeri da aree dove la domanda di mobilità risulta bassa ad aree ad alta densità. Lo studio della connessione tra queste aree avviene mediante la formulazione del Vehicle Routing Problem.

2. Servizi di Feederaggio nelle aree a domanda debole: Il Vehicle Routing Problem

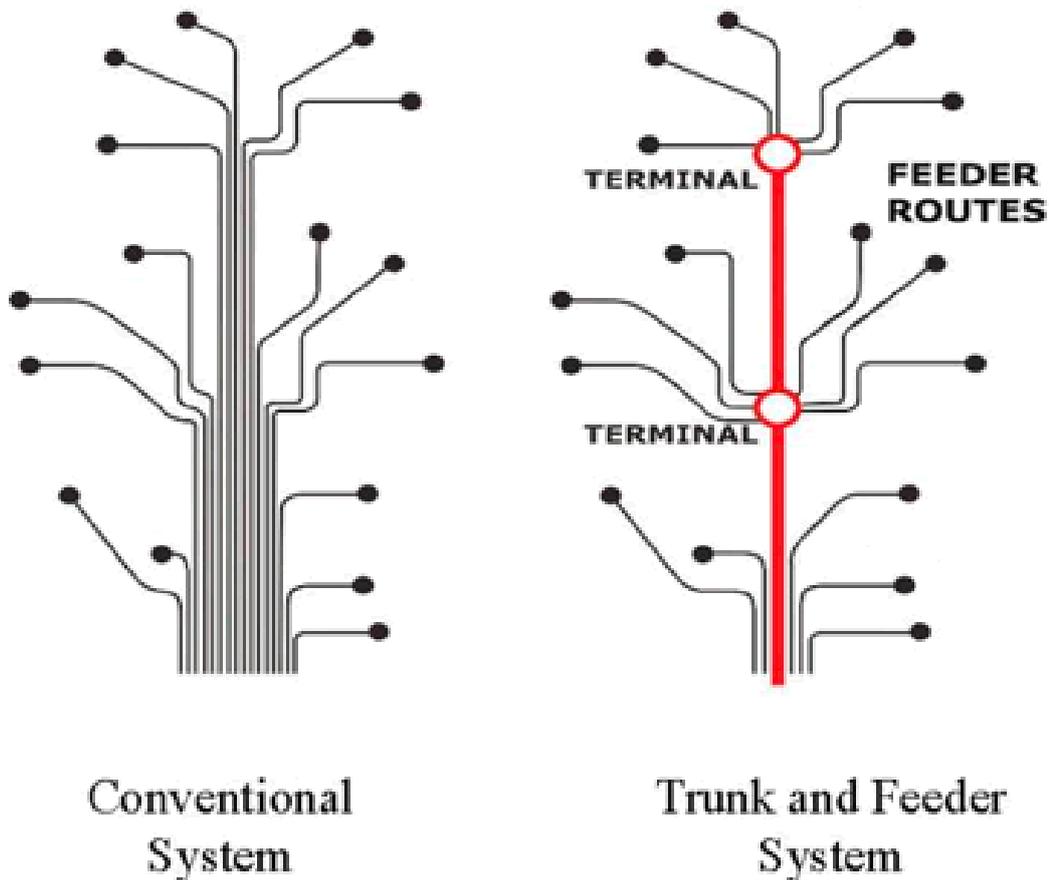
Si definiscono aree a domanda debole quelle porzioni di territorio, urbane o extraurbane, entro le quali esiste domanda di trasporto la cui ridotta entità, unitamente alla dispersione nello spazio e nel tempo e ad eventuali difficoltà dell'esercizio legate alla conformazione territoriale, è tale da rendere poco efficace e molto costosa l'organizzazione di servizi tradizionali di trasporto pubblico di linea.

L'ambito a domanda debole costituisce una vera e propria realtà territoriale come servizi in particolari fasce orarie (per esempio quelle notturne), o periodi dell'anno (per esempio i giorni festivi); servizi rivolti a particolari categorie di utenti. Rientrano in questa tipologia di domanda aree o relazioni caratterizzate da domanda scarsa quale ad esempio quella rilevabile nei comuni con una piccola porzione di popolazione sparsa sul territorio, che richiede al servizio pubblico di linea numerose fermate per raggiungere le aree centrali.

Analogamente possono costituire domanda debole anche situazioni in aree metropolitane, in cui la debolezza della domanda emerge dall'elevato grado di dispersione delle origini e destinazioni degli spostamenti e soprattutto da una rilevante disomogeneità di utilizzo del mezzo pubblico tra le ore di punta affollate e le ore di morbida sottoutilizzate.

Uno dei tipici problemi legati alle aree a domanda debole consiste nella progettazione di una corretta rete di feederaggio come connessione ai nodi principali del trasporto collettivo. Lo studio della suddetta progettazione avviene tramite la formulazione del Vehicle Routing Problem.

Figura 1: Comparazione tra trasporto convenzionale e le reti di feederaggio (Comprehensive Mobility Plans-2008)



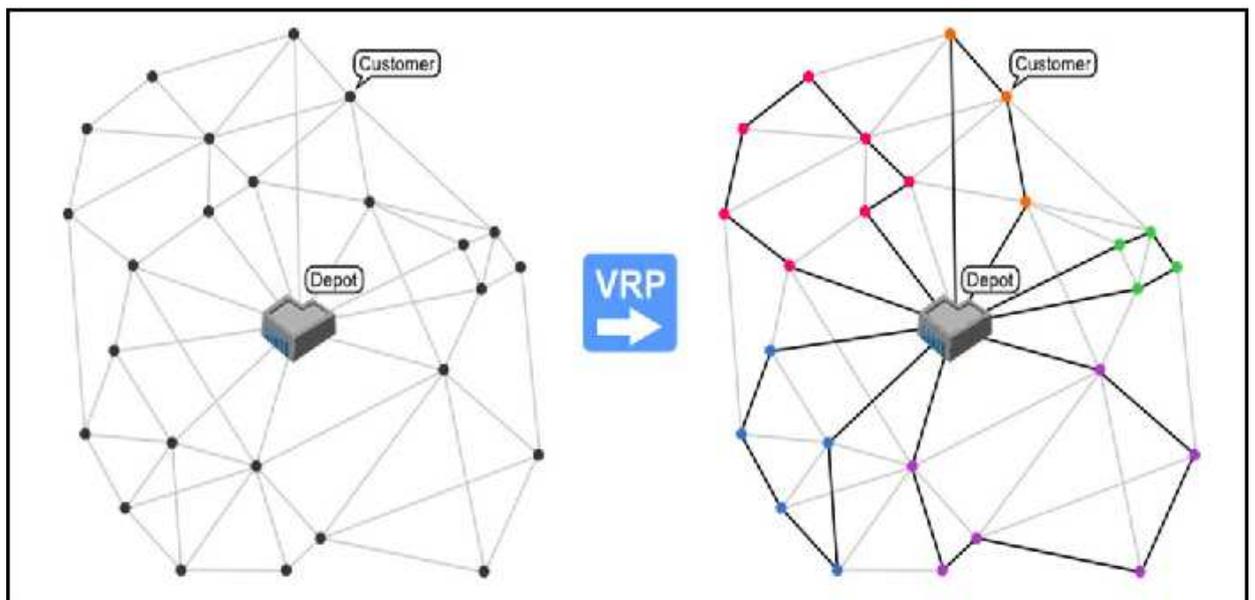
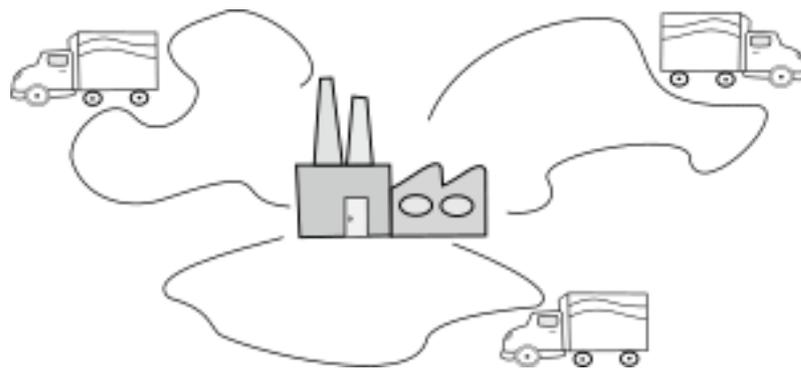
Il Vehicle Routing Problem (VRP) è un tipico problema operativo nelle reti di distribuzione, e consiste nello stabilire i percorsi di una serie di veicoli per servire un insieme di clienti. I veicoli, con determinate caratteristiche, devono visitare un insieme di clienti distribuiti all'interno di una rete di trasporto a partire da uno o più nodi di partenza.

Un problema di instradamento dei veicoli può essere definito andando a descrivere nel dettaglio le caratteristiche dei veicoli, dei clienti e della rete di trasporto che ne definiscono il contesto operativo. Si possono distinguere due grandi classi di problemi, a seconda di come siano configurati i clienti:

- Arc Routing: i clienti sono uniformemente distribuiti lungo le connessioni della rete di trasporto (es. lungo la strada);
- Node Routing: i clienti sono distribuiti come entità distinte nei nodi della rete di trasporto (di un grafo).

Nella maggior parte delle applicazioni vengono studiati problemi di node routing. Tra queste tipologie di problemi si possono definire vari casi a seconda delle differenti modalità operative del contesto applicativo preso in esame. Una prima distinzione si può fare andando a considerare, ad esempio, la capacità di carico di un veicolo. Se questa risulta superiore alla somma delle domande di tutti i clienti allora il problema del VRP si semplifica e si configura come un problema di Travelling Salesman Problem (TSP). Altrimenti, nel caso in cui la somma delle richieste di tutti i clienti risulti superiore alla capacità di trasporto di un singolo veicolo si parla allora propriamente di problemi di Vehicle Routing.

Figura 2- Esempio di Vehicle Routing Problem e della sua soluzione (Networking and Emerging Optimization-2013)



Possibili applicazioni del VRP sono frequenti in problemi logistici e distributivi di dettaglio. Per esempio, si può pensare a contesti applicativi in cui si debba distribuire al dettaglio (oppure provvedere alla raccolta) un certo bene. Oltre questi campi applicativi, si sono sviluppate nel tempo nuove forme di VRP, definite per le linee feeder.

Di seguito vengono descritte in dettaglio le variabili considerate nei generici problemi di VRP:

2.1. Clienti

Nella versione generale del VRP si considerano clienti caratterizzati solamente da un quantitativo di merce che deve essere ritirata o consegnata. Per esempio, in alcune applicazioni il cliente deve essere servito in un dato intervallo temporale. Si parla in questi casi di problemi di vehicle routing con time windows. Una particolarità di questa classe di problemi è che se il veicolo raggiunge il cliente al di fuori della finestra temporale consentita allora il veicolo deve aspettare (nel caso in cui arrivi prima dell'inizio della finestra temporale) o deve andar via senza portare a termine l'operazione (nel caso in cui sia in ritardo). Un'altra famiglia di problemi che vengono presi in considerazione sono i pick-up and delivery problem, in cui alcuni clienti richiedono il ritiro di merce, mentre altri richiedono la consegna di merce, e la merce ritirata da un cliente può sia essere consegnata ad un deposito oppure essere consegnata ad un altro cliente in zona. Una complicazione di cui si deve tener conto in questa famiglia di problemi è dovuta alla capacità del veicolo, infatti, in fase di soluzione del problema, si dovrà tener conto della massima capacità del veicolo e dell'ordine in cui i clienti vengono visitati.

2.2. Veicoli

Un altro parametro rilevante per la classificazione dei problemi di VRP è dato dalle caratteristiche dei veicoli. Si parla di:

- Flotte omogenee: i veicoli che compongono il problema sono tutti identici tra loro;
- Flotte eterogenee: i veicoli differiscono tra loro per diverse caratteristiche (es. capacità di carico, velocità, autonomia ecc..)

Altra possibile dimensione da considerare per la classificazione dei problemi di VRP è data dalla funzione obiettivo. In alcune versioni si cerca di minimizzare la distanza percorsa dai veicoli, in altre versioni i costi (che possono includere costi fissi e costi variabili), o il numero di veicoli necessari.

2.3. Formulazione Matematica

Il CVRP (Capacitated Vehicle Routing Problem) è la versione più comune di questa famiglia di problemi. Ciò che caratterizza questa tipologia di problemi è il fatto che il servizio è di semplice consegna senza raccolta. Inoltre le richieste dei clienti sono note a priori e devono essere soddisfatte da un solo veicolo; tutti i veicoli sono identici e basati su di un singolo deposito centrale. Gli unici vincoli imposti riguardano le capacità dei veicoli. L'obiettivo è minimizzare il costo totale di servizio.

Nella formulazione viene solitamente considerato un grafo $G = (N, A)$ non orientato e pesato sugli archi. Ogni arco è collegato tra due nodi i e j con il costo di percorrenza c_{ij} ed il nodo deposito è il nodo $d \in N$. Si ipotizza che tutti i nodi abbiano domanda unitaria e che la flotta di veicoli abbia una massima autonomia Q . Ogni veicolo quindi è in grado di servire al più Q clienti prima di essere costretto a tornare al deposito.

La formulazione matematica proposta nel problema generale risulta inservibile per problemi di grande dimensione o per i quali sia necessario avere una soluzione in tempi di calcolo molto brevi.

Inoltre la grande complessità dei problemi di VRP rende molto difficile, o al limite impossibile, il calcolo di una soluzione ottima.

Attraverso diversi studi, si è scelto di procedere nell'analisi del VRP attraverso l'utilizzo di euristiche, ovvero un approccio alla soluzione del problema tramite particolari algoritmi, trattati per sommi capi nel paragrafo seguente.

3. Una soluzione al VRP: Gli algoritmi Euristici

Lo sviluppo di algoritmi è mirato a fornire una soluzione di buona qualità ad un problema difficile con un limitato tempo di calcolo.

In questo senso ci vengono in aiuto gli algoritmi euristici, che impiegano tempi di calcolo relativamente ristretti per fornire soluzioni di buona qualità. In particolare gli algoritmi euristici si dividono in:

- Euristici Classici: ottengono buone soluzioni con limitati tempi di calcolo;
- Metaeuristici: approfondiscono la ricerca della soluzione ottima nelle zone più promettenti dello spazio delle soluzioni, implementando sofisticate regole di ricerca e di ricombinazione dei risultati parziali ottenuti.

Quest'ultima tipologia di algoritmi viene utilizzata molto spesso per la risoluzione di problemi di VRP più complessi poiché i risultati ottenuti sono solitamente di qualità superiore rispetto agli euristici classici, per quanto richiedano un tempo maggiore per giungere alla soluzione. Inoltre, l'esecuzione di questi algoritmi è subordinata alla corretta valutazione e impostazione di un predeterminato numero di parametri, propri dell'algoritmo stesso, al fine di adattare il metodo di risoluzione al problema e ottenere la soluzione migliore possibile.

La famiglia dei metaeuristici proposti per il problema del vehicle routing si può suddividere in sei categorie di seguito presentate:

- Simulated Annealing (SA);
- Deterministic Annealing (DA);
- Tabu Search (TS);
- Algoritmi genetici (GA);
- Reti Neurali (NN);
- Ant Colony System (ACS);

- **Simulated Annealing**

L'approccio proposto con il metodo SA prevede che, ad ogni iterazione t , venga scelta casualmente una soluzione x appartenente a una struttura di dati chiamata neighborhood $N(x_t)$; se è verificato che $f(x) \leq f(x_t)$, allora X_{t+1} è posta uguale a x , altrimenti:

- a) $X_{t+1}=x$ con probabilità P_t
- b) $X_{t+1}=x_t$ con probabilità $1-P_t$

$$\text{Con } P_t = e^{-\frac{f(x) - f(x_t)}{\theta t}}$$

- **Deterministic Annealing**

La principale differenza che distingue la tecnica Deterministic Annealing dal precedente Simulating Annealing consiste nel fatto che le mosse che si intendono effettuare sono accettate o rigettate secondo regole deterministiche.

- **Metaeuristici Tabu Search**

Questa tecnica esplora lo spazio delle soluzioni attraverso la determinazione, ad ogni iterazione t , della soluzione migliore contenuta in un sottoinsieme del neighborhood $N(t)$. Al fine di evitare cicli all'interno della ricerca e lo stazionamento della stessa in un intorno di un minimo locale, le soluzioni migliori vengono marcate da un attributo che le rende non selezionabili nelle iterazioni successive: sono definite quindi tabu.

- **Algoritmi Genetici**

Si tratta di una tecnica che cerca di risolvere problemi imitando i processi tipici dell'evoluzione naturale. In via del tutto generale, l'idea di fondo consiste nel mantenere memoria di stringhe di bit, chiamate cromosomi. L'evoluzione della popolazione `e ottenuta applicando degli operatori che simulino i più importanti fenomeni naturali: riproduzione e mutazione.

- **Reti Neurali**

Le reti neurali artificiali, o semplicemente reti neurali, sono modelli computazionali la cui struttura prevede l'interconnessione di diversi elementi di elaborazione elementari. Queste celle di elaborazione si ispirano ai neuroni del sistema nervoso umano, mentre i loro collegamenti rappresentano le sinapsi. Nelle reti artificiali ogni collegamento è associato ad un proprio peso numerico, il cui valore varia dinamicamente nel tempo in funzione dell'esperienza acquisita nella computazione. Grazie ai pesi associati alle connessioni, le reti neurali godono dell'importante caratteristica che è la capacità di induzione.

Si rimanda alla descrizione dell'Ant Colony System e delle sue applicazioni nel capitolo successivo.

Capitolo 2

CASO STUDIO

1. Ant Colony System: Un algoritmo innovativo

L'algoritmo di ottimizzazione della colonia di formiche (o ACS in inglese), è stato ideato inizialmente da Marco Dorigo nei primi anni '90 per la ricerca dei percorsi ottimali in un grafo e appartiene alla categoria degli **algoritmi naturali** ossia alla categoria degli algoritmi che studiano i comportamenti naturali degli esseri viventi costruiti in milioni di anni di evoluzione. La prima forma di questo particolare algoritmo si basa sul comportamento delle formiche che cercano un percorso tra la loro colonia e una fonte di cibo.

L'idea nasce dall'osservazione dallo sfruttamento delle risorse alimentari da parte delle formiche. Infatti, quest'ultime, anche se limitatamente alle capacità cognitive di una singola formica, sono in grado, collettivamente, di trovare il percorso più breve tra una fonte di cibo ed il formica.

Figura 3- Comportamento delle formiche (Mehmet Karatay-2007)



Le formiche usano l'ambiente circostante come mezzo di comunicazione: si scambiano informazioni indirettamente depositando feromone una volta trovato il percorso ottimale, in modo da descrivere lo stato del loro "lavoro". Le informazioni vengono scambiate in ambito locale, infatti una formica ha accesso al feromone solo se si trova nel luogo dove questo è stato depositato. Questo sistema è chiamato "stigmergia".

Il meccanismo che permette di risolvere un problema troppo complesso per essere affrontato da singole formiche e rappresenta un esempio di sistema auto-organizzato. Questo sistema si basa su retroazioni positive (la deposizione di feromone attira altre formiche che rafforzeranno il percorso) e negative (la dissipazione del percorso a causa dell'evaporazione impedisce al sistema di bloccarsi). È logico pensare che se la quantità di feromone rimane la stessa nel corso del tempo su tutti i percorsi, non ne verrà scelto alcuno. Tuttavia, a causa della retroazione, una piccola variazione su un percorso sarà amplificata e quindi consente la scelta di un percorso. L'algoritmo consente di passare da uno stato instabile dove nessun percorso è migliore di un altro, ad uno stato stabile in cui il percorso viene realizzato da tratti "migliori".

Figura 4- Schema algoritmo (Nojhan-2006)

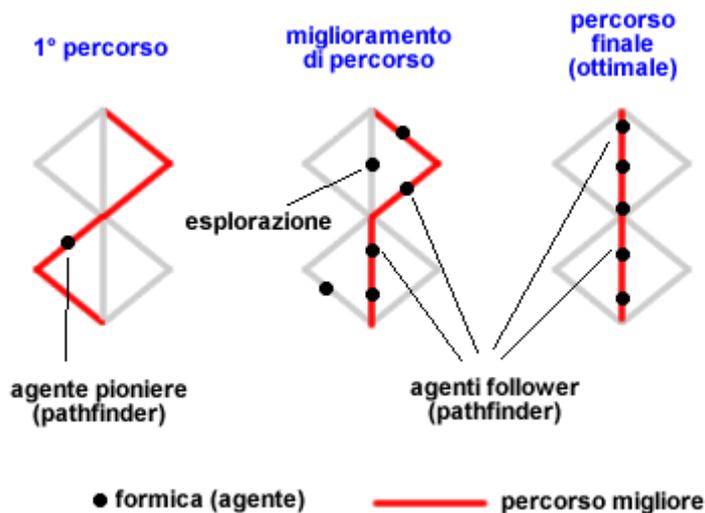
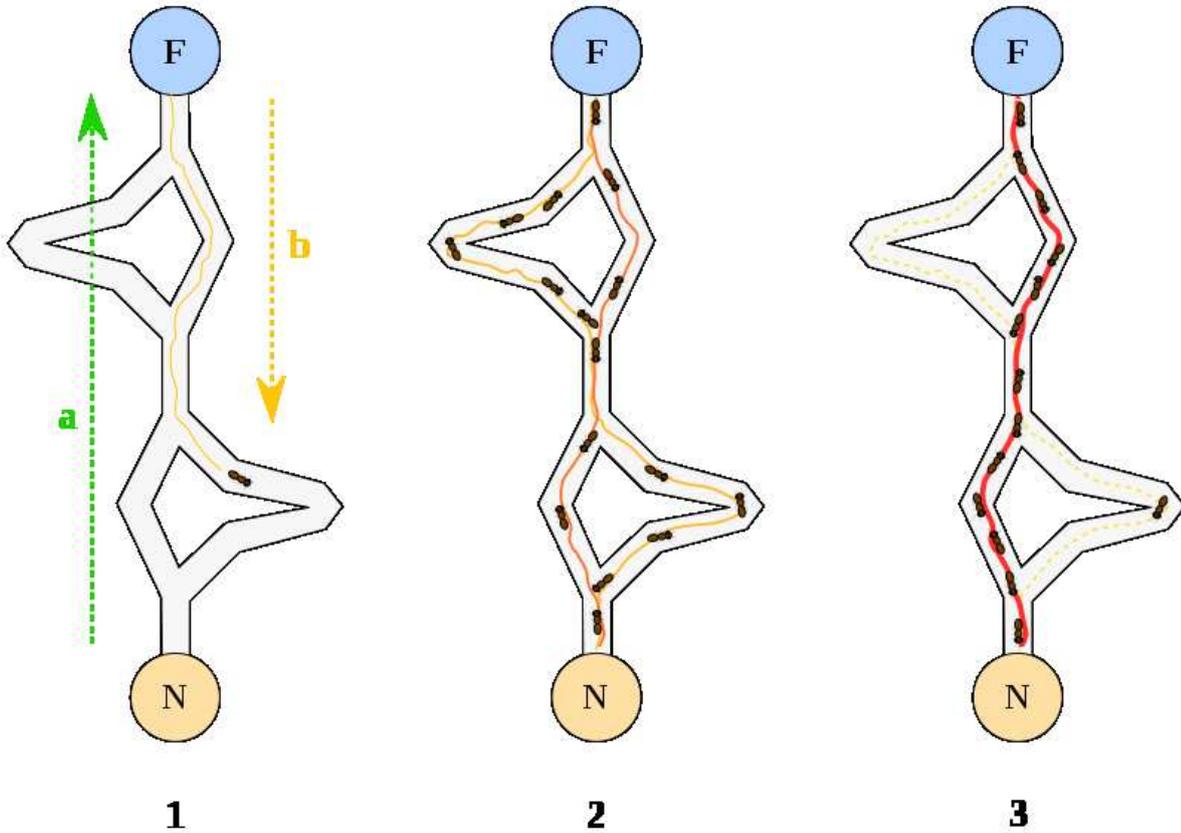


Figura 5 -Ricerca percorso migliore (Johann Dréo-2006)



- 1) la prima formica trova la fonte di cibo (F), tramite un percorso qualsiasi (a), poi ritorna al formicaio (N) lasciando dietro di sé una scia di feromone (b);
- 2) Le altre formiche percorrono, indiscriminatamente, quattro percorsi possibili, ma il rafforzamento della pista rende più attraente il percorso più breve;
- 3) le formiche percorrono il percorso più breve, le lunghe porzioni di altri percorsi perdono la loro scia di feromone;

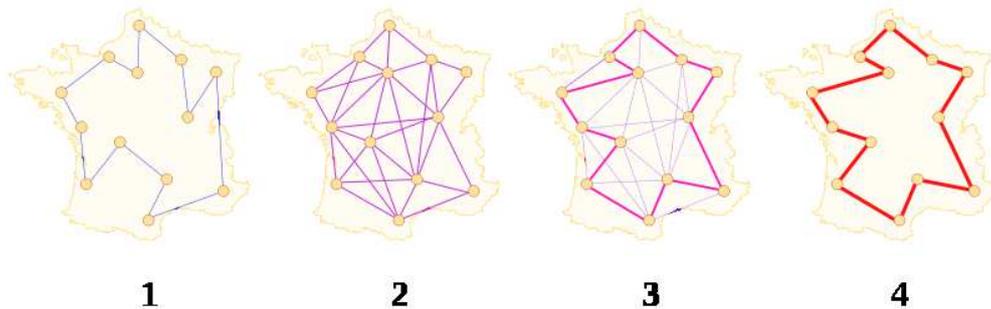
L'idea originale da allora si è diversificata per risolvere una classe più ampia di problemi, facendo quindi emergere diversi algoritmi attingendo da vari aspetti del comportamento delle formiche.

Il primo algoritmo delle colonie di formiche proposto è l'*Ant System* il cui scopo è risolvere il problema del commesso viaggiatore, dove l'obiettivo è quello di trovare la via più breve per collegare una serie di città.

L'algoritmo generale è relativamente semplice ed è basato su un insieme di formiche, ciascuna delle quali attraversa un percorso tra quelli possibili costruendo le componenti della soluzione in modo incrementale. In ogni fase, la formica sceglie di spostarsi da una città all'altra secondo alcune regole:

- più una città è distante, meno possibilità ha di essere scelta (la "*visibilità*");
- più l'intensità del percorso di feromone situato sul crinale tra due città è maggiore, più ha possibilità di essere scelto;
- una volta completato il suo percorso, la formica deposita, su tutti i bordi attraversati, più feromone se il percorso è breve;
- una formica può visitare un vincolo una ed una sola volta;
- i percorsi di feromone evaporano ad ogni iterazione.

Figura 6- Esempio problema del commesso viaggiatore (Nojhan-2006)



- 1) una formica sceglie un percorso e ne crea uno col feromone;
- 2) tutte le formiche percorrono un certo numero di tracce, depositando ognuna una quantità di feromone proporzionale alla qualità del percorso;
- 3) ogni bordo del miglior percorso è più rinforzato rispetto agli altri;
- 4) l'evaporazione rimuove le soluzioni peggiori

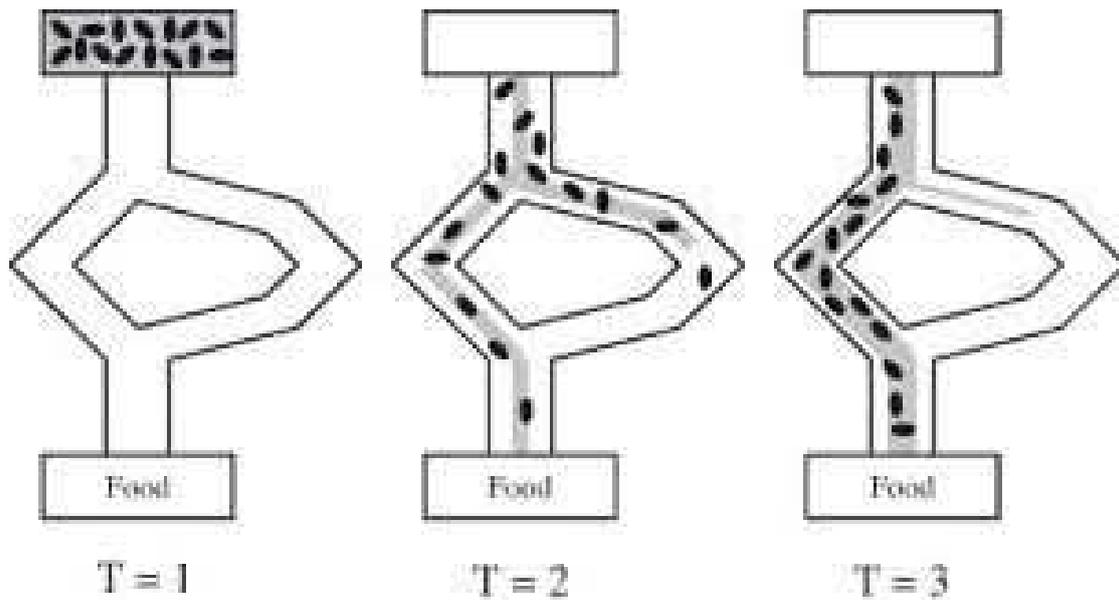
Non è facile dare una definizione precisa di ciò che è o ciò che non è un algoritmo delle colonie di formiche.

In modo molto generale, gli algoritmi delle colonie di formiche sono considerati come delle metaeuristiche di popolazione, dove ogni soluzione è rappresentata da un movimento della formica sullo spazio di ricerca. Le formiche segnano le migliori soluzioni, e tengono conto delle marcature precedenti per ottimizzare la loro ricerca.

Essi possono essere considerati come algoritmi probabilistici multi-agente che utilizzano una distribuzione di probabilità implicita per la transizione tra ogni iterazione. Nelle loro versioni per problemi combinatori, usano una costruzione iterativa di soluzioni.

La differenza sostanziale fra gli algoritmi delle colonie di formiche ed altri processi metaeuristici simili è proprio il suo aspetto costruttivo. Infatti, nei problemi combinatori, è possibile che alla fine sarà trovata la soluzione migliore, anche se nessuna formica l'avrà utilizzata in modo efficiente. Così, nell'esempio del problema del commesso viaggiatore, non è necessario che una formica effettivamente faccia il percorso più breve: può essere costruito da segmenti rinforzati delle soluzioni migliori.

Figura 7: Descrizione comportamento formiche



All'istante T=1 le formiche partono dallo stesso punto di partenza. In T=2 attuano entrambi i percorsi per arrivare alla fonte di cibo. Le formiche che trovano un percorso migliore rilasciano feromone per permettere in T=3 a tutta la colonia di percorrere lo stesso percorso.

Il comportamento collettivo degli insetti sociali rimane una fonte di ispirazione per i ricercatori. L'ampia varietà di algoritmi (per l'ottimizzazione e non) che sostengono l'auto-organizzazione nei sistemi biologici ha dato origine al concetto di "swarm intelligence", che è un quadro molto generale, al cui interno vi sono anche gli algoritmi delle colonie di formiche.

2. Applicazione progettuale

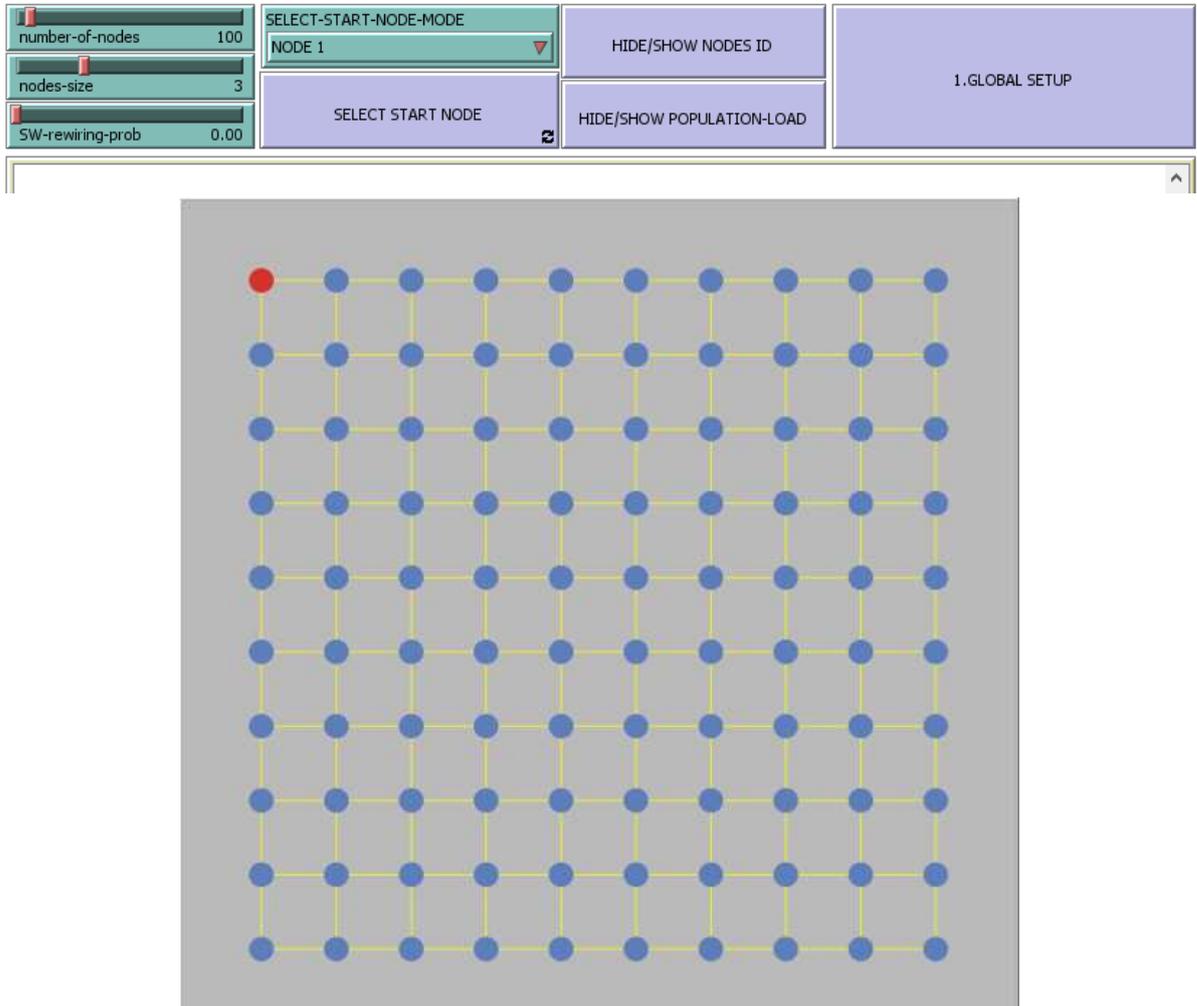
Dopo aver analizzato i principali aspetti sull'utilizzo dell'algoritmo di ottimizzazione della colonia di formiche, si procede con l'applicazione del suddetto algoritmo in un caso ideale.

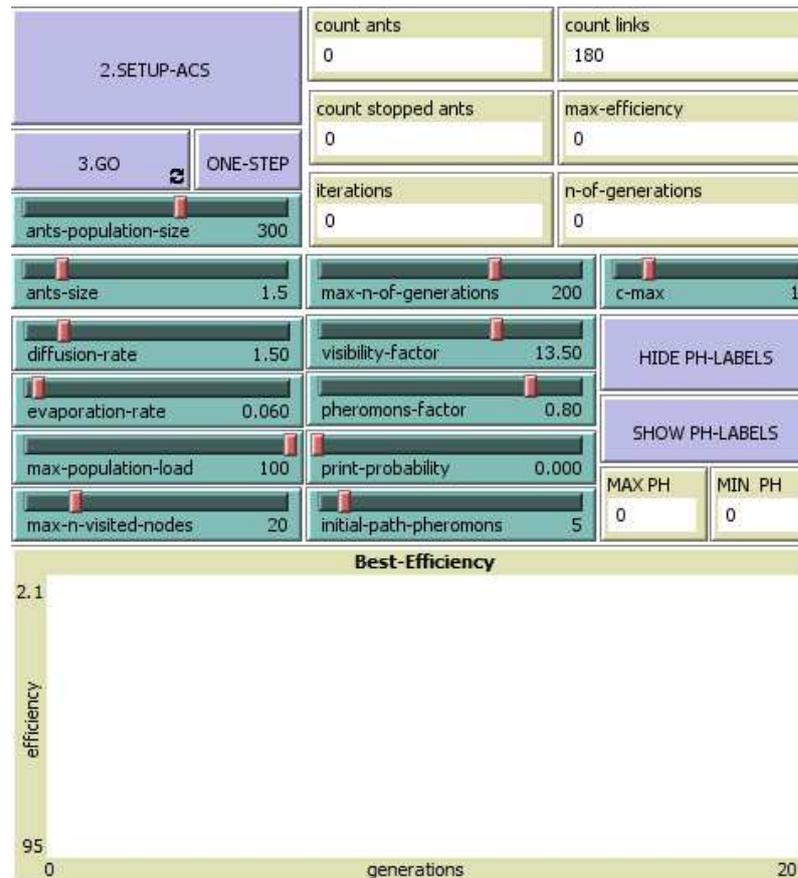
Per l'applicazione si è utilizzato il software Netlogo, sviluppato originariamente da Uri Wilensky nel 1999 presso il Center for Connected Learning and Computer-Based Modeling della Northwestern University, il quale offre un ambiente di sviluppo ideale per la realizzazione di modelli di simulazioni ad agenti, di networks di sistemi dinamici.

Sulla base della letteratura scientifica esistente e con la collaborazione del Prof. Pluchino del DFA dell'Università di Catania, è stato messo a punto un codice che simulasse gli effetti dell'ACS applicato a un problema di Vehicle Routing tramite la modifica di differenti parametri di controllo accuratamente studiati.

Di seguito sono riportate delle immagini dell'ambiente di lavoro:

Figura 8- Ambiente di lavoro Netlogo con relativi parametri





- ants-population-size: Numero di formiche presenti durante una simulazione;
- ants-size: Grandezza delle formiche durante una simulazione (Valore fissato);
- diffusion-rate: Parametro di diffusione del feromone (Valore fissato);
- evaporation-rate: Parametro di evaporazione del feromone dei percorsi peggiori (Valore fissato);
- max-population-load: Valore massimo della popolazione che gravita sui vari nodi;
- max-n-visited-nodes: Numero massimo di nodi visitabili dalle formiche in una simulazione;
- max-n-of-generations: Numero massimo di generazioni di formiche in una simulazione (Valore fissato);
- visibility-factor: Parametro che rappresenta la probabilità che la formica attraversi un arco. L'inverso di una distanza (Valore fissato);
- pheromons-factor: Quantità di feromone su un arco (Valore fissato);
- initial-path-pheromons: Quantità di feromone su un arco alla prima iterazione (Valore Fissato);

3. Calcolo del percorso migliore in una rete di trasporto collettivo tramite ACS: Descrizione programma

Lo scopo del programma sarà quello di trovare il percorso migliore in una rete di trasporto collettivo ideale, descritto dal parametro della “best efficiency”, ovvero la migliore efficienza tra tutti i percorsi trovati per diversi scenari.

L'ambiente di sviluppo del programma utilizzato può essere diviso in tre parti:

- Dati di input;
- Interfaccia grafica;
- Dati di output.

I dati di input ci permettono di settare l'interfaccia grafica per il lancio delle varie simulazioni. Il setting dei dati viene attuato tramite l'interazione con il sistema attraverso pulsanti e sliders per modificare i parametri di controllo. I primi parametri di cui disponiamo sono:

- Number of nodes: permette la scelta del numero di nodi all'interno dell'interfaccia grafica;
- Nodes size: con cui possiamo scegliere la grandezza dei nodi;
- Select start node mode: permette la scelta del nodo di partenza e di arrivo.

Attraverso simulazioni preliminari si è optato di fissare i valori del numero di nodi e della grandezza dei nodi rispettivamente a 100 e 3.

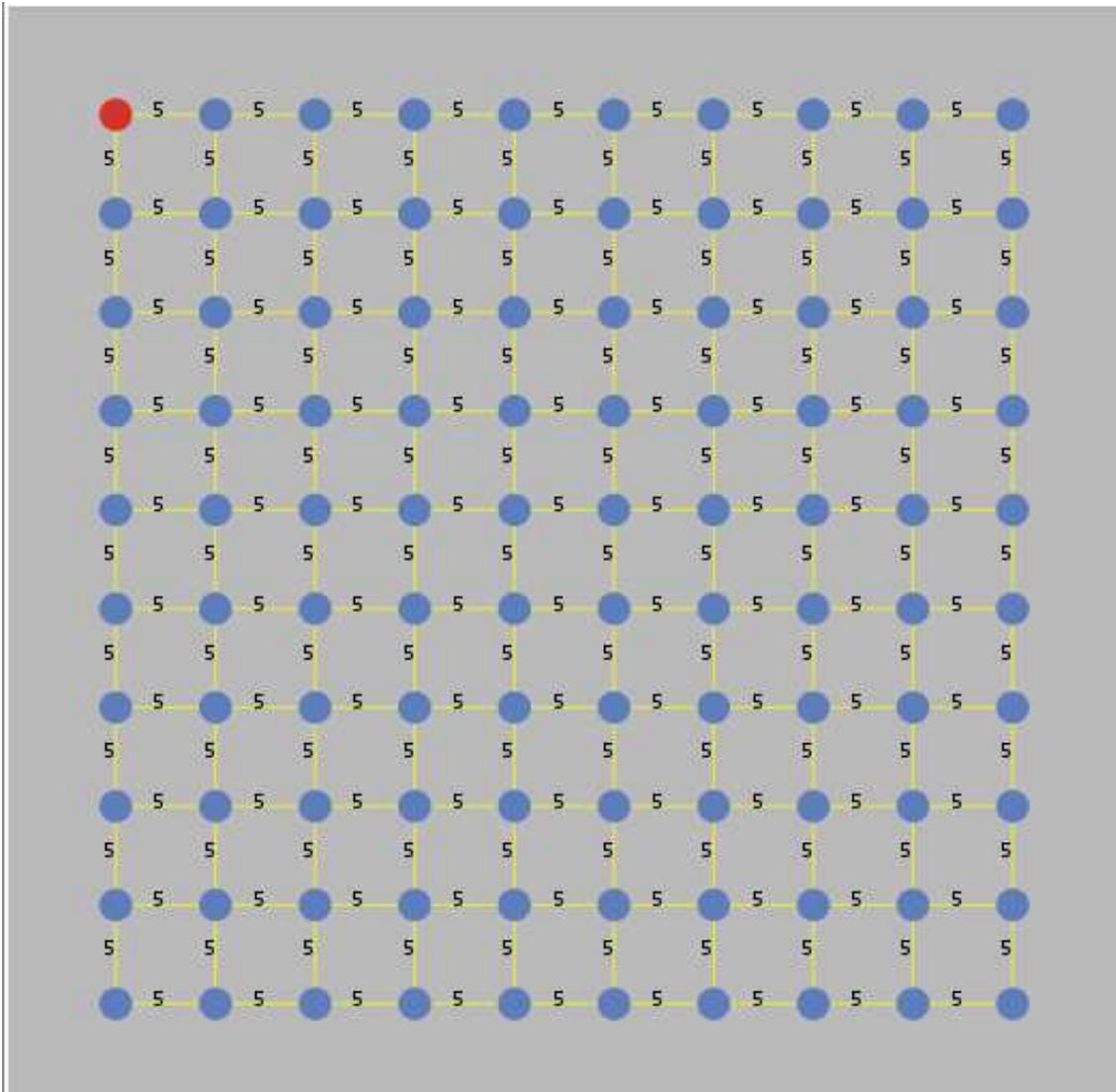
Per quanto riguarda la posizione del nodo di inizio e fine percorso, per simulare idealmente il problema del Vehicle Routing in un caso ideale, e una sua possibile soluzione, per le aree a bassa domanda, si è scelto il nodo 1, che corrisponde al nodo in alto a sinistra.

Figura 9- Dati di Input

The image shows two sets of input controls. The left set consists of two sliders: the top one is labeled 'number-of-nodes' with a value of 100, and the bottom one is labeled 'nodes-size' with a value of 3. The right set consists of a dropdown menu labeled 'SELECT-START-NODE-MODE' with 'NODE 1' selected, and a button labeled 'SELECT START NODE' with a small icon in the bottom right corner.

Una volta fissati i primi parametri di controllo si procede al setup delle formiche sull'interfaccia grafica, che rappresenta una rete ideale stradale di forma quadrata, i cui archi possono fare parte del percorso della linea di trasporto collettivo. Inizialmente, fissato un numero di formiche tramite il setup, esse graviteranno tutte sul nodo di inizio e di fine percorso, contrassegnato in rosso. Notiamo che nei nodi sarà presente un numero fisso, che corrisponde alla quantità di feromone presente all'inizio del ciclo, il cui comando associato è l'initial path pheromones, che si è scelto di fissare a 5 per ogni simulazione.

Figura 10- Interfaccia grafica al setup di una generica simulazione



Completato il setup e scelto il numero delle formiche che faranno parte della colonia (Ants population size), ogni formica esplora la rete con dei criteri e delle regole ben precise. Ogni formica deve decidere su quale dei 4 nodi adiacenti dovrà spostarsi, quindi la generica formica α , sul generico nodo I , calcola la probabilità di spostarsi al nodo J . Il passaggio sui vari modi è attuato tramite la costruzione di una probabilità, dettata dalla seguente legge:

$$P_{IJ} = \frac{\tau_{IJ}}{\sum_{K=0}^N \tau_{IK}} \cdot \eta_{IJ}$$

dove il primo termine rappresenta l'intensità di feromone (Pheromons Factor) su un arco ed il secondo termine rappresenta la visibilità (Visibility Factor) di un nodo J dal nodo I , ovvero l'inverso della distanza tra i due nodi. Il pheromon factor è un ulteriore parametro che si è scelto di fissare a 0,80 per ogni scenario. L'intensità di feromone viene normalizzata rispetto alla somma di tutte le intensità di feromone in modo tale da ottenere una probabilità compresa tra 0 ed 1.

La distanza calcolata non è una distanza fisica, ma a sua volta è definita dalla legge:

$$d_{IJ} = \lambda_{MAX} + \lambda_{cJ} - \lambda_c = \frac{1}{\eta_{IJ}}$$

La distanza sarà sempre un numero maggiore di 1, e sarà data dalla differenza tra il moltiplicatore combinato λ_c calcolato dalla formica in tutto il suo percorso compiuto e il moltiplicatore combinato λ_{cJ} considerando anche il meccanismo associato al nodo J .

Entrambi i moltiplicatori combinati danno il valore della popolazione che gravita su un nodo durante la simulazione. Il moltiplicatore λ_{MAX} è una costante che rende la distanza sempre maggiore di uno. Più piccola è la distanza calcolata, maggiore è la Visibility, maggiore sarà la probabilità che una formica attraversi un dato arco. Una volta definita la distanza, moltiplicando la visibilità per il pheromon factor i valori verranno normalizzati per ottenere la probabilità.

Se la probabilità calcolata è maggiore di R (numero random compreso tra 0 e 1) allora la formica si sposta al nodo J , altrimenti ripete l'operazione per il nodo successivo. Se la formica non si sposta sul nodo successivo, quindi la probabilità risulta bassa per tutti gli archi disponibili, la formica esce dal ciclo. Si ripete la stessa operazione fino a quando tutte le formiche non sono uscite dal ciclo, quindi fino alla fine di un'iterazione.

Durante il processo di ricerca del best path, ogni formica è vincolata da un numero massimo di nodi che potrà visitare durante ogni iterazione, fattore che si è deciso di sfruttare come variabile per i vari scenari studiati.

Figura 11- Processo di ricerca delle formiche

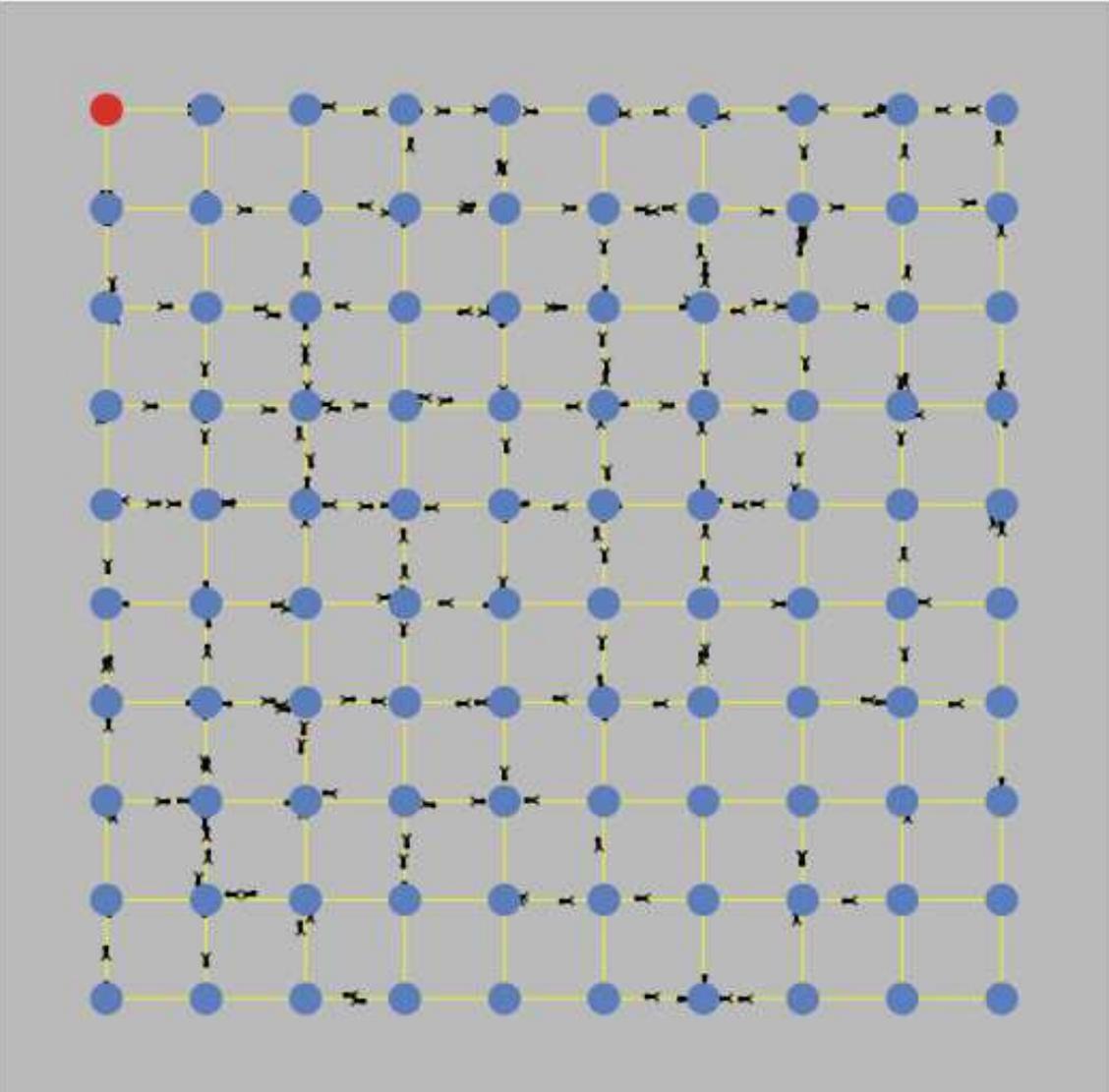
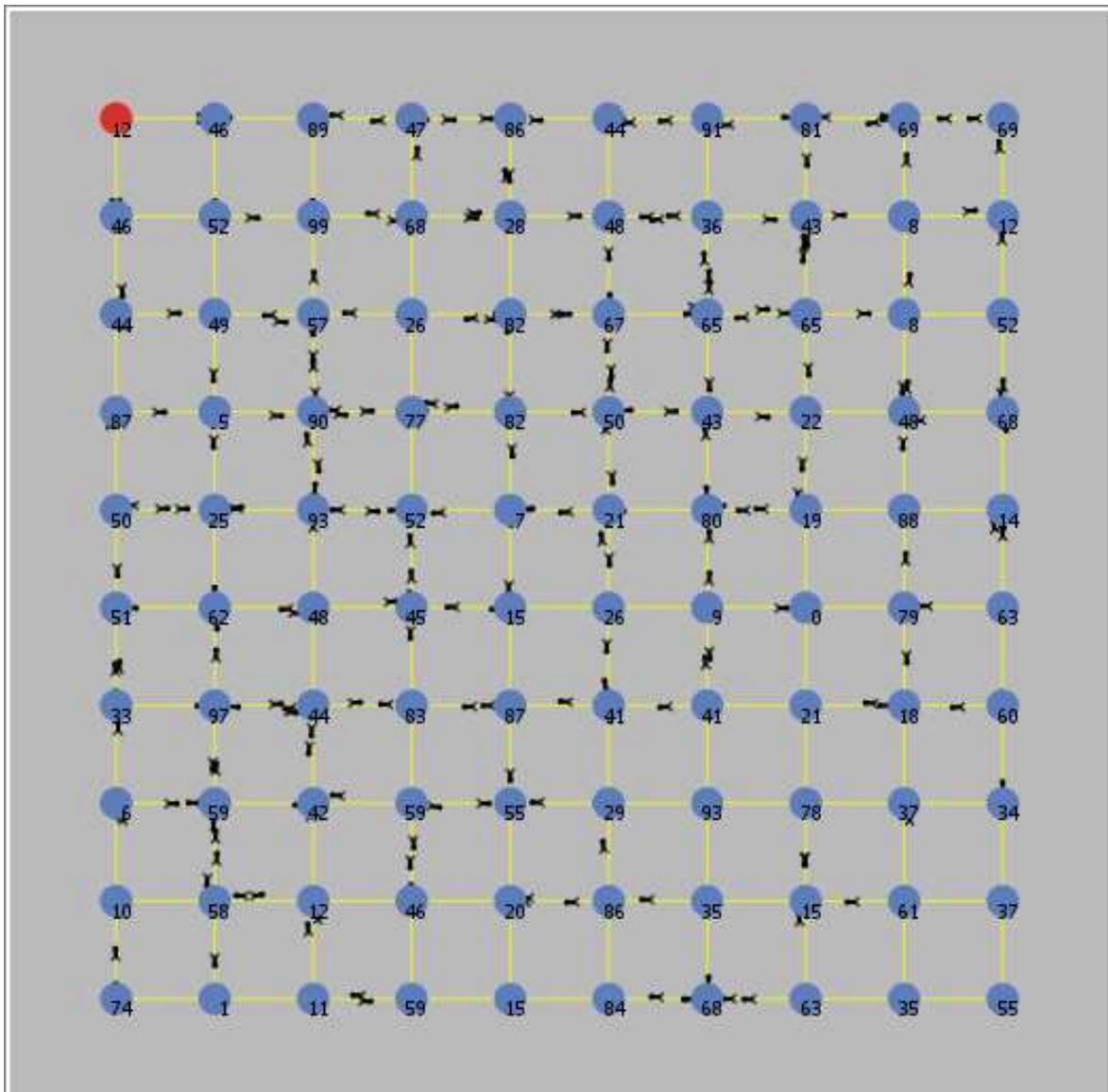


Figura 12- Processo di ricerca delle formiche. I numeri sui nodi rappresentano la quantità di popolazione che gravita sui nodi stessi durante una generica simulazione.



Alla fine della prima iterazione, sarà rimasta un'unica formica che possiede il massimo valore di efficienza tale da definire il percorso migliore (Best Ant). La “migliore formica”, a cui è associato il λ_c minore deposita il feromone sul generico arco IJ , e la nuova quantità di feromone depositato sui vari archi verrà calcolata secondo la legge:

$$\Delta\tau_{IJ}^a(t) = \frac{Q}{L^a}$$

dove Q è una costante che determina la quantità di feromone depositata, scalata in funzione della lunghezza dell'intero percorso (L^a) compiuto dalla formica.

Dopo il rilascio, si aggiorna la Matrice dei Feromoni secondo la legge:

$$\tau_{IJ}(t + 1) = (1 - \rho)\tau_{IJ}(t) + \Delta\tau_{IJ}^a(t)$$

dove ρ è una costante, compresa tra 0 e 1 che regola l'evaporazione del feromone (0 nessuna evaporazione, 1 evaporazione completa). È possibile implementare questa equazione con altri due parametri α e β , che rappresentano due costanti che scalano l'importanza relativa nella scelta del percorso dell'intensità di feromone e della visibilità (olfatto e vista).

Per ogni iterazione, resterà traccia della best ant in quanto aumenterà la quantità di feromone di un parametro chiamato “Diffusion Rate” in maniera proporzionale. Si è scelto di fissare questo parametro a un valore di 1,50.

Tramite i dati di output è possibile leggere, per ogni iterazione, il valore della best efficiency, ovvero il valore massimo dell'efficienza tra tutti i percorsi migliori trovati durante la simulazione, e il numero della best ant che ha trovato il best path.

Figura 13- Output della prima iterazione di una generica simulazione

```
observer: "n-of-generations = 1"  
(ant 191): "Best ant n.191"  
(ant 191): " path-length = 20"  
(ant 191): " efficiency = 328.0311722992191"
```

Ad ogni iterazione corrisponderanno best ants e best paths diversi. Qualora dopo una generica iterazione il best path risultasse con un'efficienza minore rispetto al best path dell'iterazione precedente il feromone della migliore formica dell'iterazione precedente diminuisce di una certa quantità dettata dal parametro dell' "Evaporation Rate", che si è scelto di fissare per ogni iterazione a un valore di 0,060.

Ogni percorso migliore sarà evidenziato all'interno dell'interfaccia grafica in verde, mentre ogni percorso con un'efficienza minore rispetto a quello dell'iterazione precedente sarà evidenziato in rosso.

Gli stessi dati possono essere visualizzati in un grafico che mette in relazione l'efficienza di ogni iterazione con il numero di generazioni di formiche che si è scelto di definire in una data simulazione.

Figura 14- Grafico dati di output. I picchi corrispondono ai percorsi migliori delle varie iterazioni

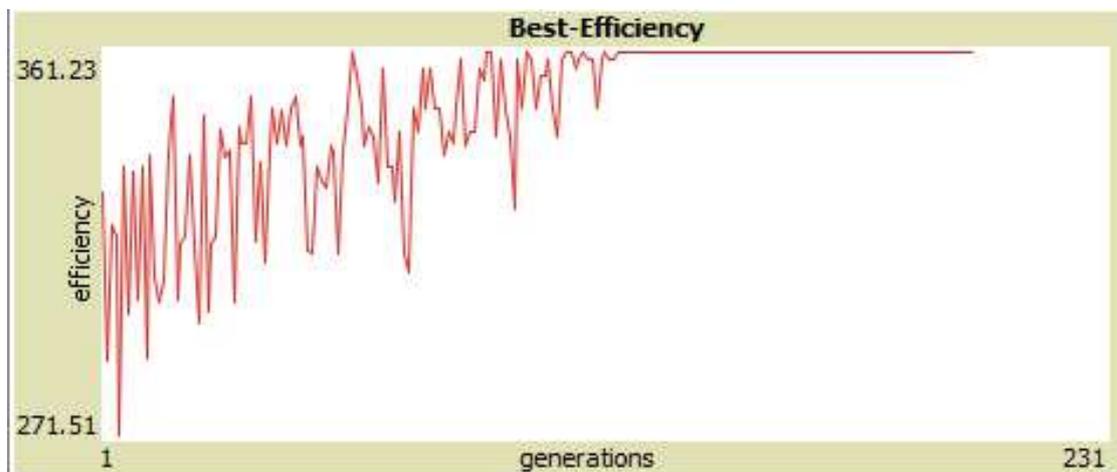
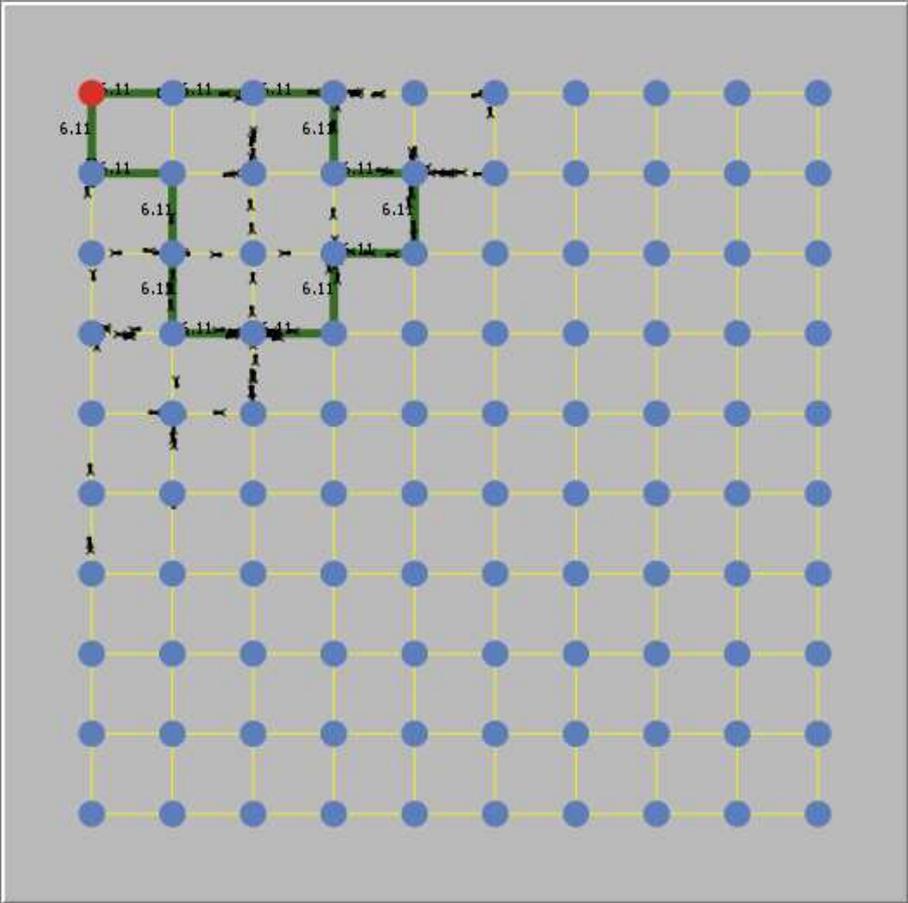


Figura 15- Dettagli dei percorsi migliori definiti in più iterazioni



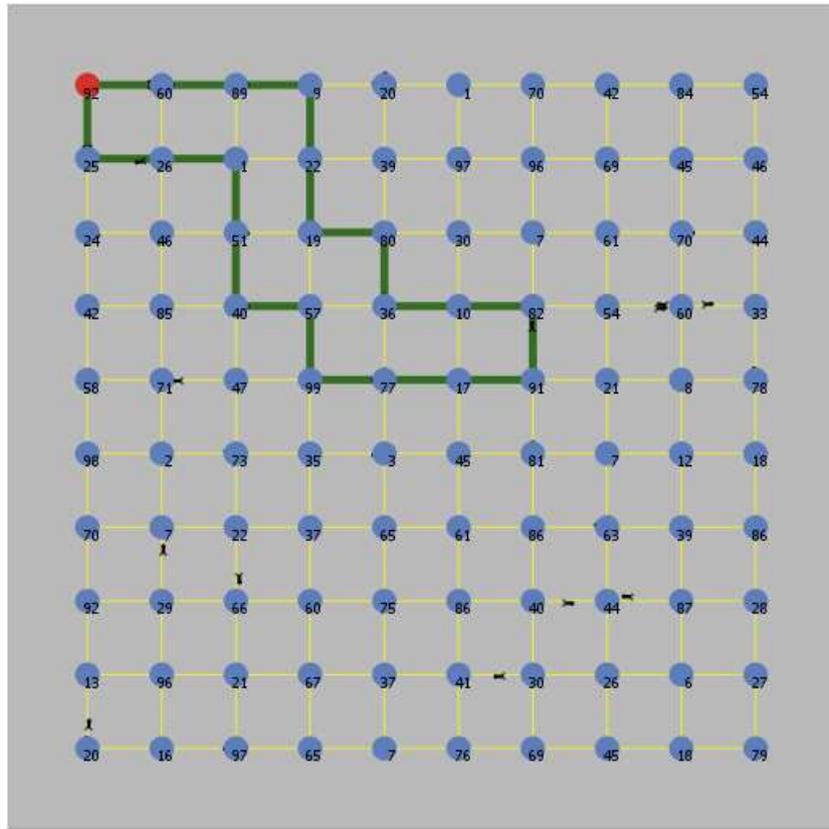
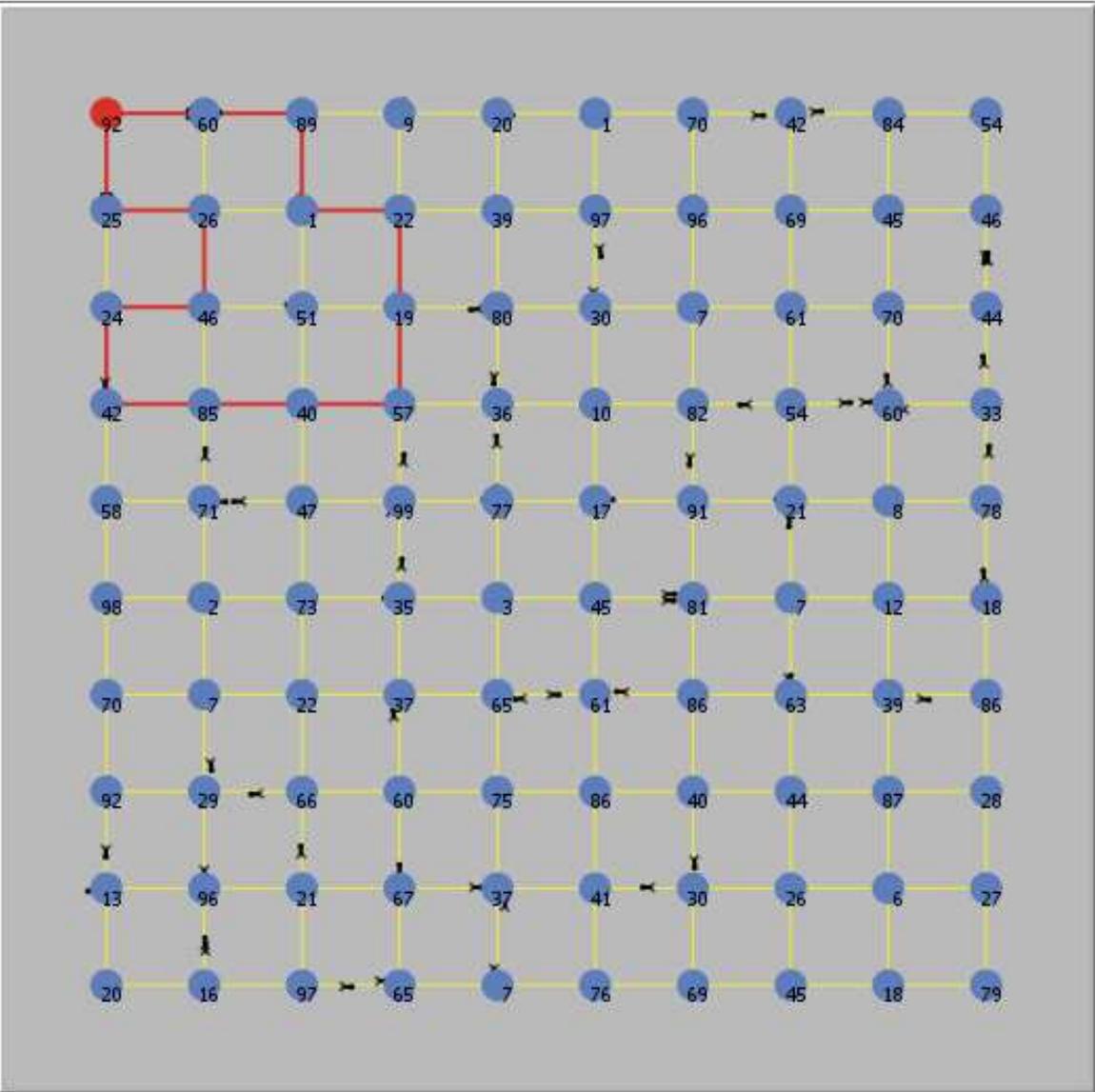


Figura 16- Dettaglio percorso migliore con un'efficienza minore del precedente



4. Simulazioni e Risultati

Una volta descritte le funzioni del programma, si è scelto di attuare tre simulazioni modificando dei parametri descritti di seguito:

- max-n-visited-nodes: ovvero il numero massimo che una colonia di formiche può visitare durante una simulazione;
- max-population-load: ovvero il numero massimo della popolazione che può gravitare sui nodi durante una simulazione;
- ants-population-size: ovvero il numero totale di formiche in una simulazione.

Ogni parametro variabile è stato modificato tre volte. Tramite il comando “Export” presente su Netlogo si è proceduto all'estrazione dei dati delle varie simulazioni per poi successivamente elaborarli in fogli di lavoro per la costruzione di tre grafici.

Lo scopo delle simulazioni è stato trovare un trend crescente, descritto dal grafico in basso a destra dell'ambiente di sviluppo, per definire la variazione della massima efficienza dei percorsi migliori in funzione dei tre parametri sopracitati.

Di seguito vengono riportati i risultati tramite degli screenshot dell'ambiente di sviluppo e i vari grafici delle simulazioni:

Figura 17- Best Path (max-n-visited-nodes=20): Percorso migliore definito dalla simulazione con numero massimo di nodi fissato a 20

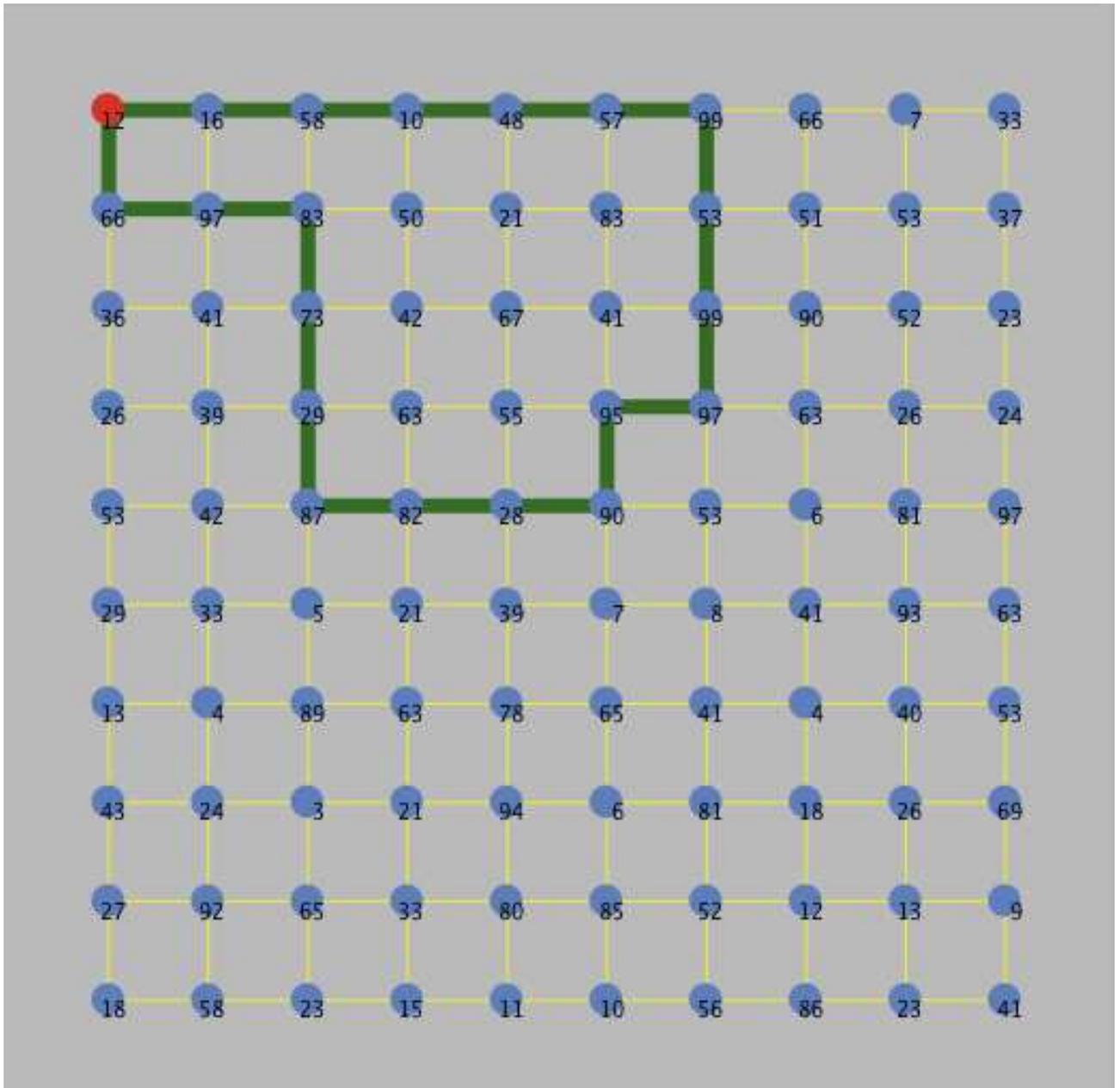


Figura 18- Parametri fissati per la simulazione e valori di output finali (max-n-visited-nodes=20)

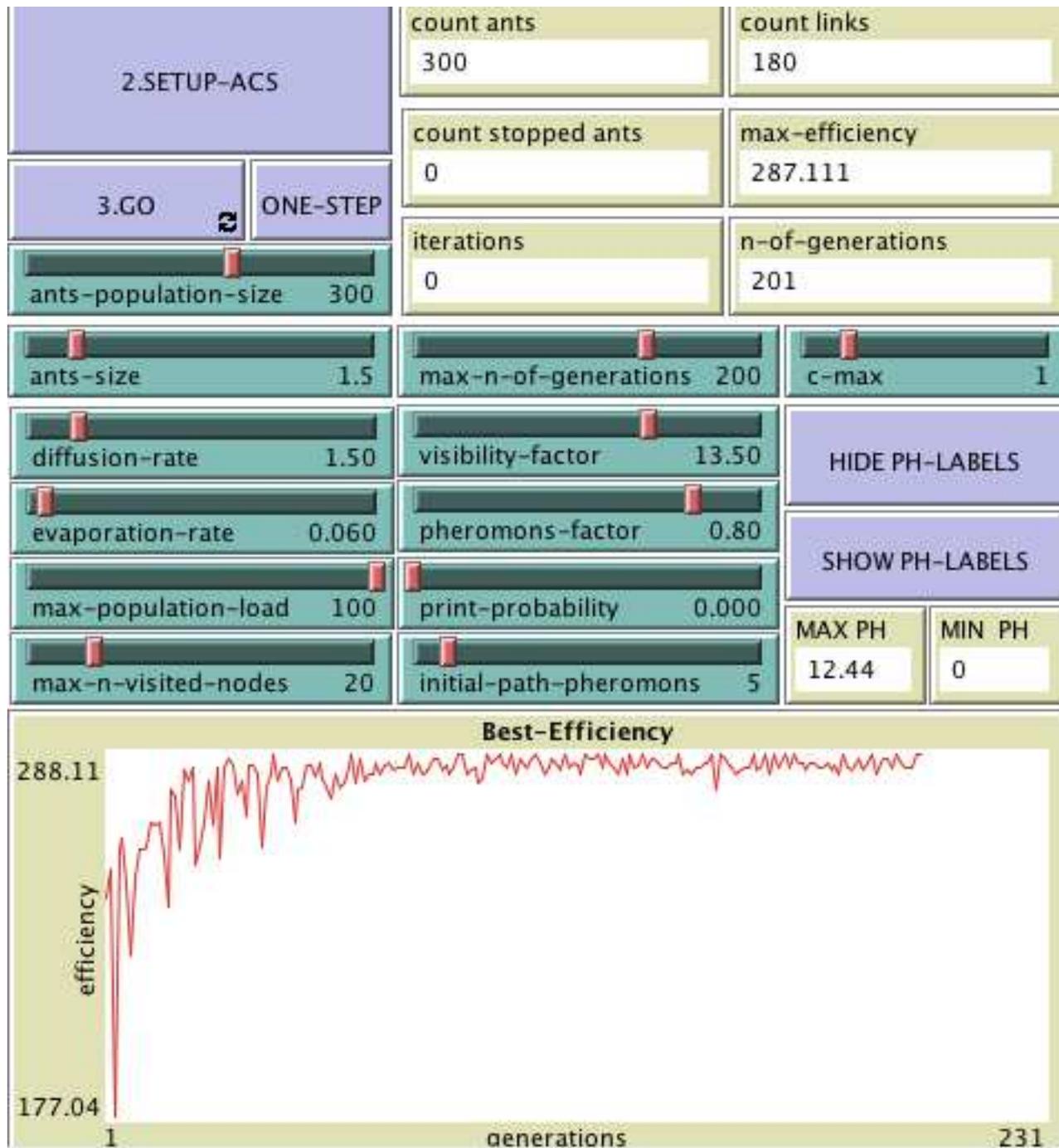


Figura 21- Best Path (max-n-visited-nodes=35): Percorso migliore definito dalla simulazione con numero massimo di nodi fissato a 35

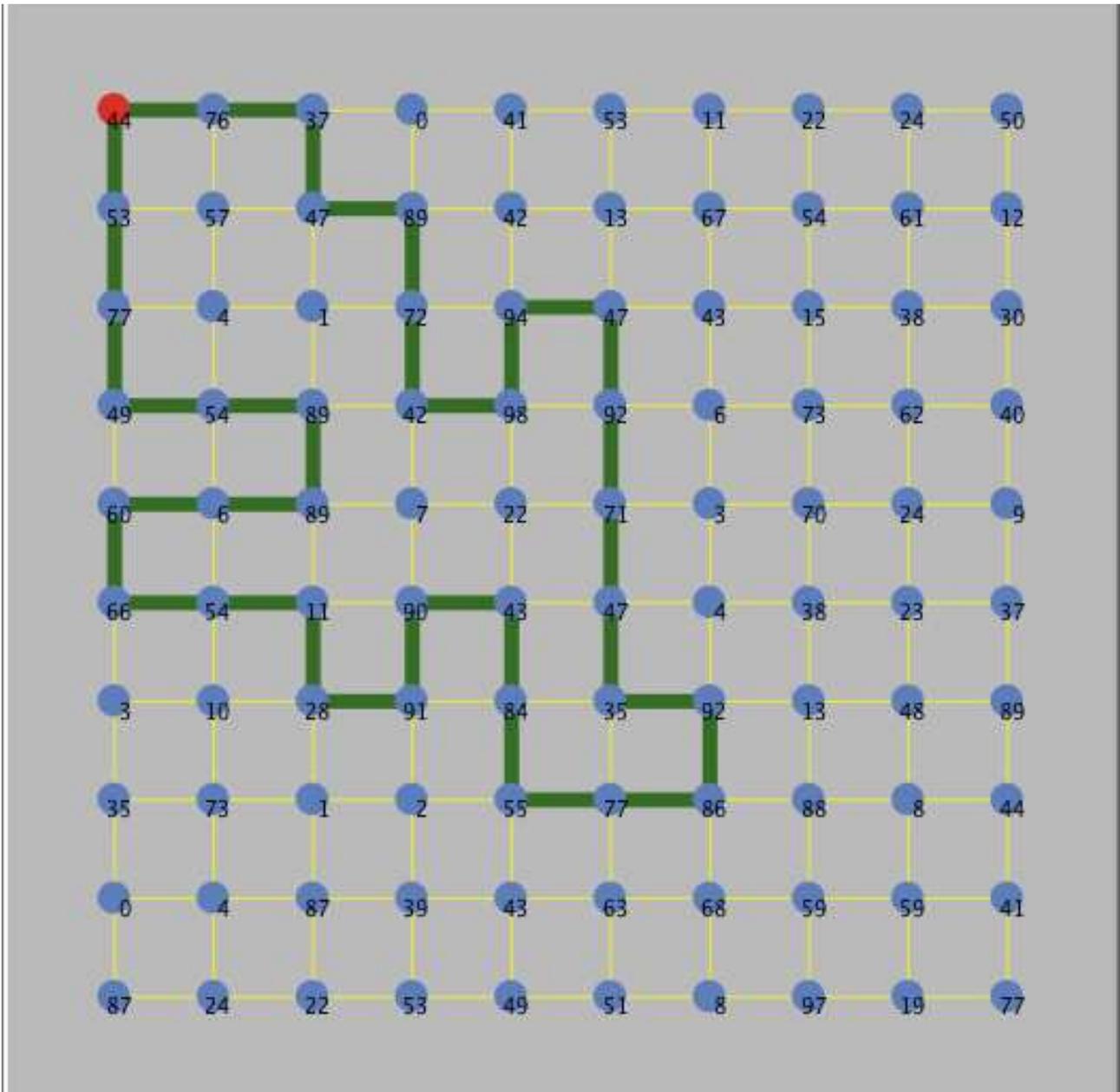


Figura 22- Parametri fissati per la simulazione e valori di output finali (max-n-visited-nodes=35)

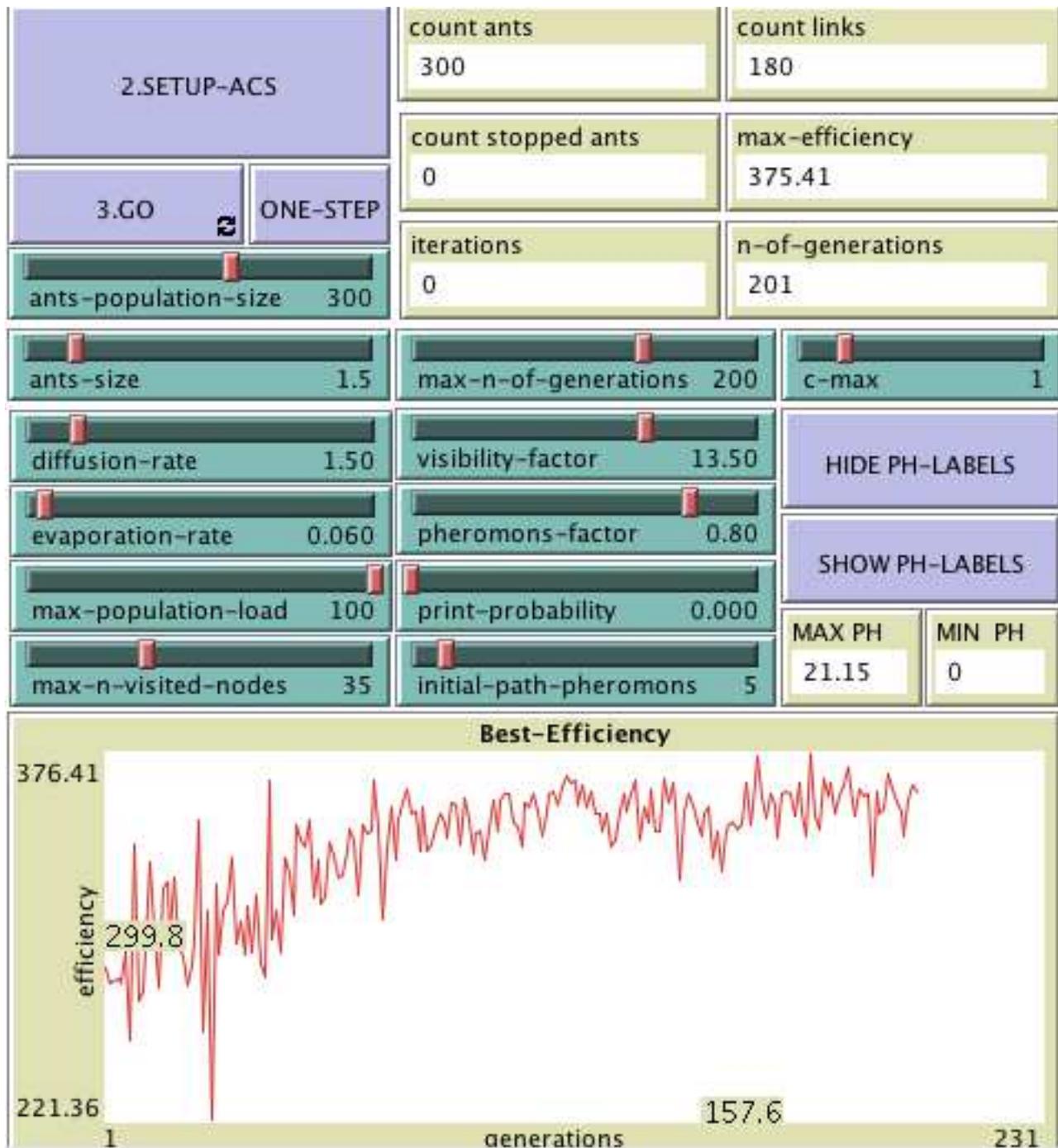


Figura 23 - Best Path (max-n-visited-nodes=15): Percorso migliore definito dalla simulazione con numero massimo di nodi fissato a 15

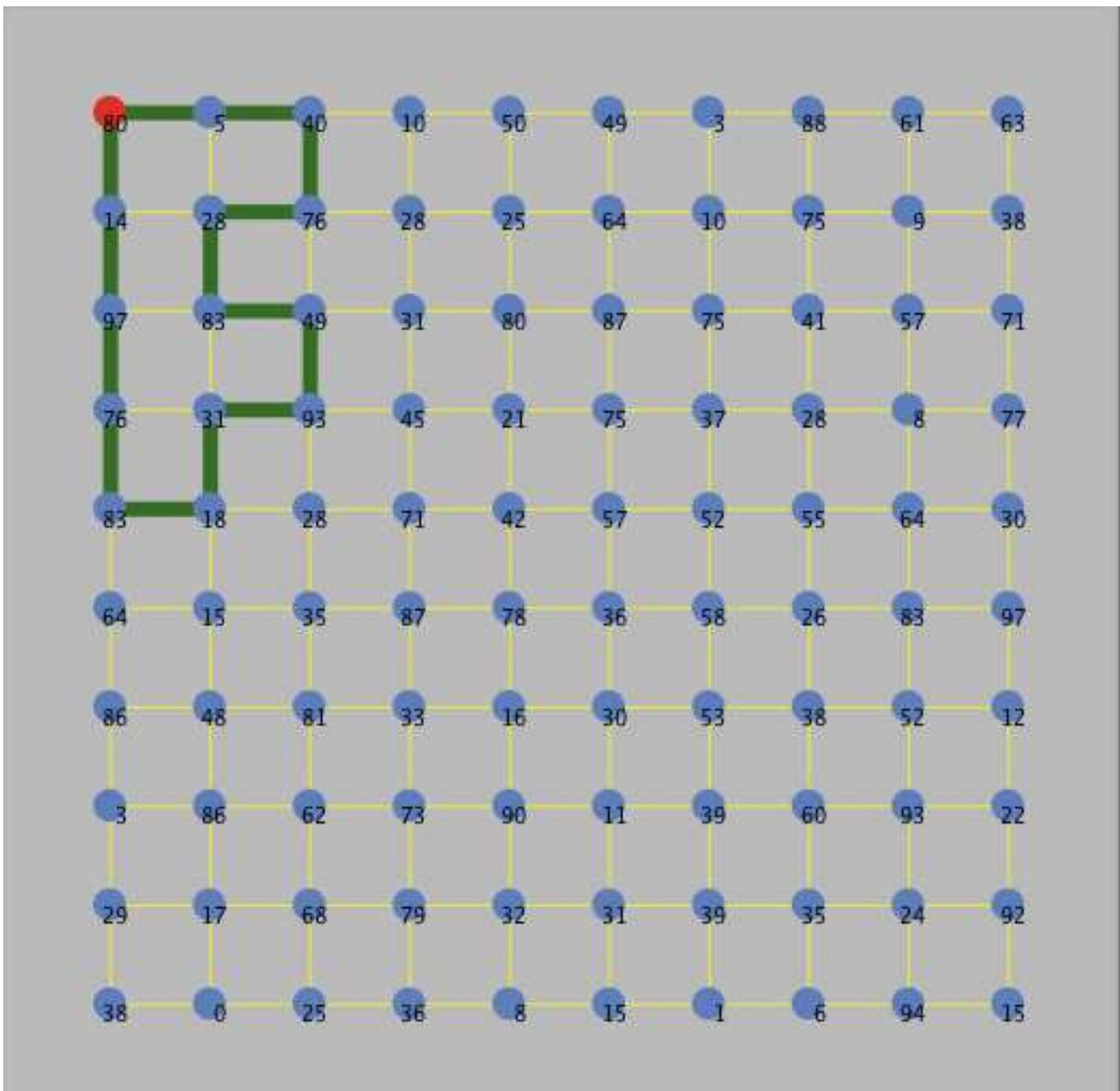


Figura 24- Parametri fissati per la simulazione e valori di output finali (max-n-visited-nodes=15)

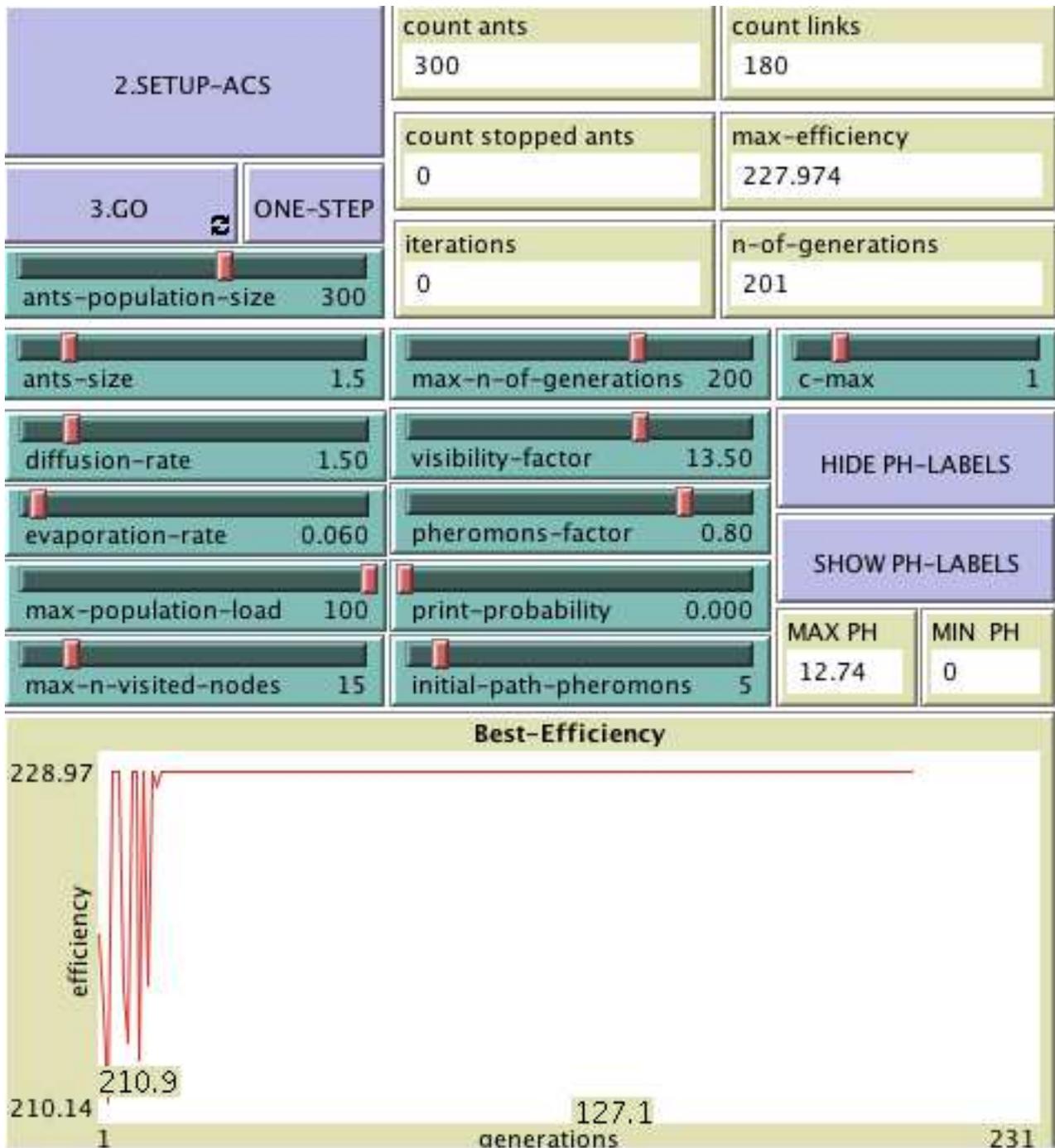
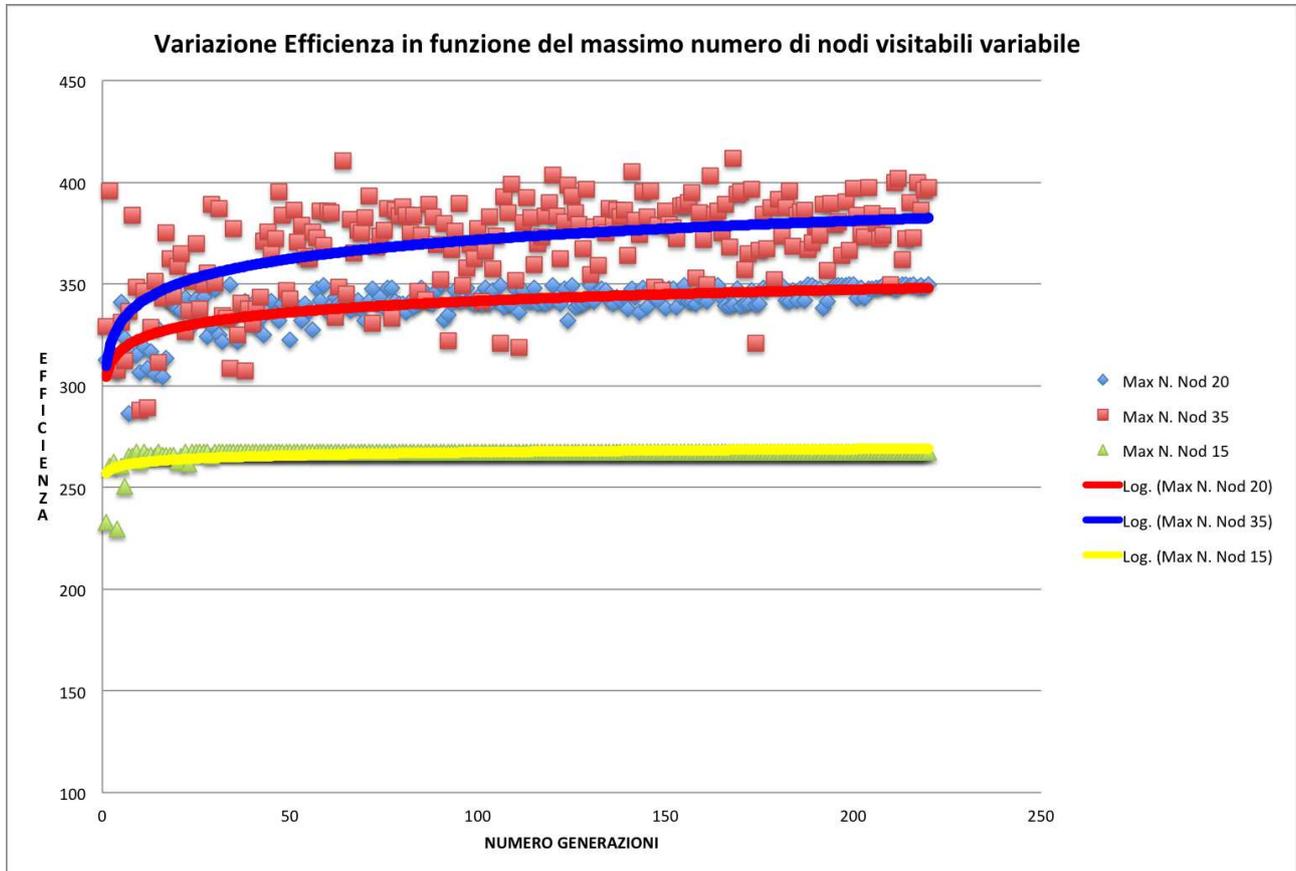


Grafico 1- Dati complessivi max-n-visited-nodes



Come si evince dal grafico e dalle figure precedenti, aumentando il numero di nodi visitabili, aumenta la possibilità data alla formiche di creare un percorso migliore più lungo. L'efficienza tenderà a un valore asintotico con curve più ripide all'aumentare del parametro considerato.

Figura 25 - Best Path (max-population-load=100): Percorso migliore definito dalla simulazione con numero massimo della popolazione in un nodo fissato a 100

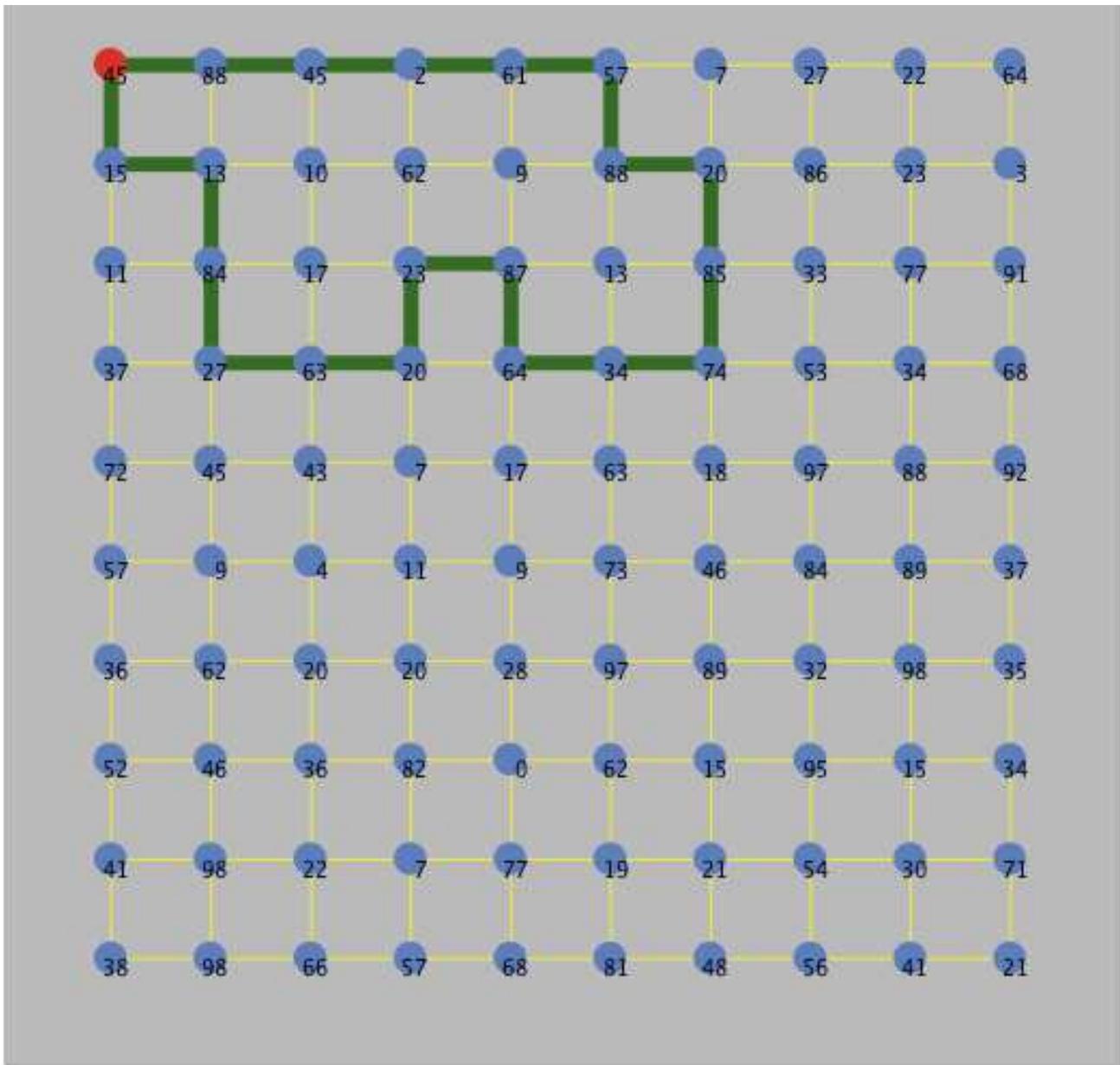


Figura 26- Parametri fissati per la simulazione e valori di output finali (max-population-load=100)

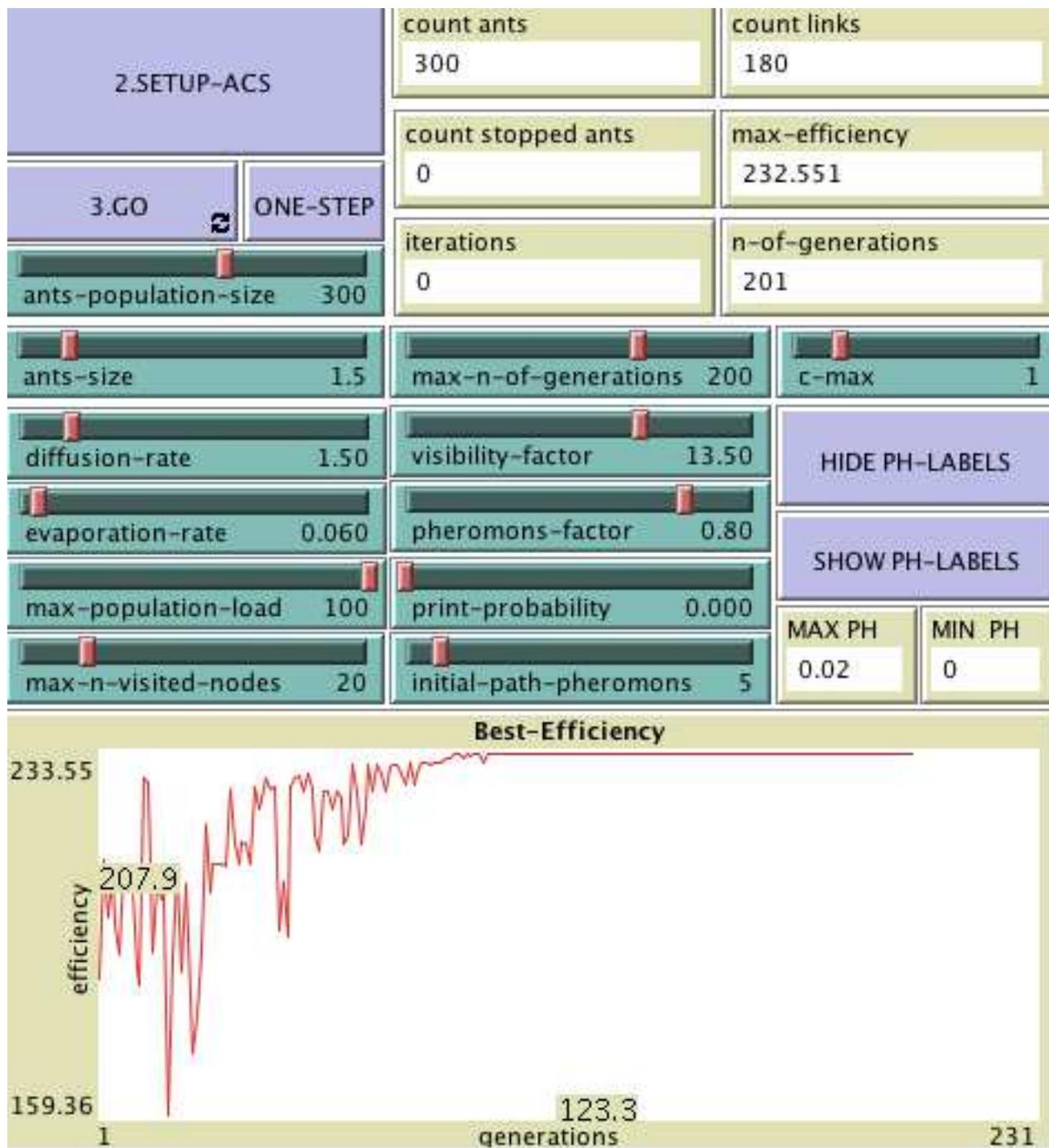


Figura 27 - Best Path (max-population-load=80): Percorso migliore definito dalla simulazione con numero massimo della popolazione in un nodo fissato a 80

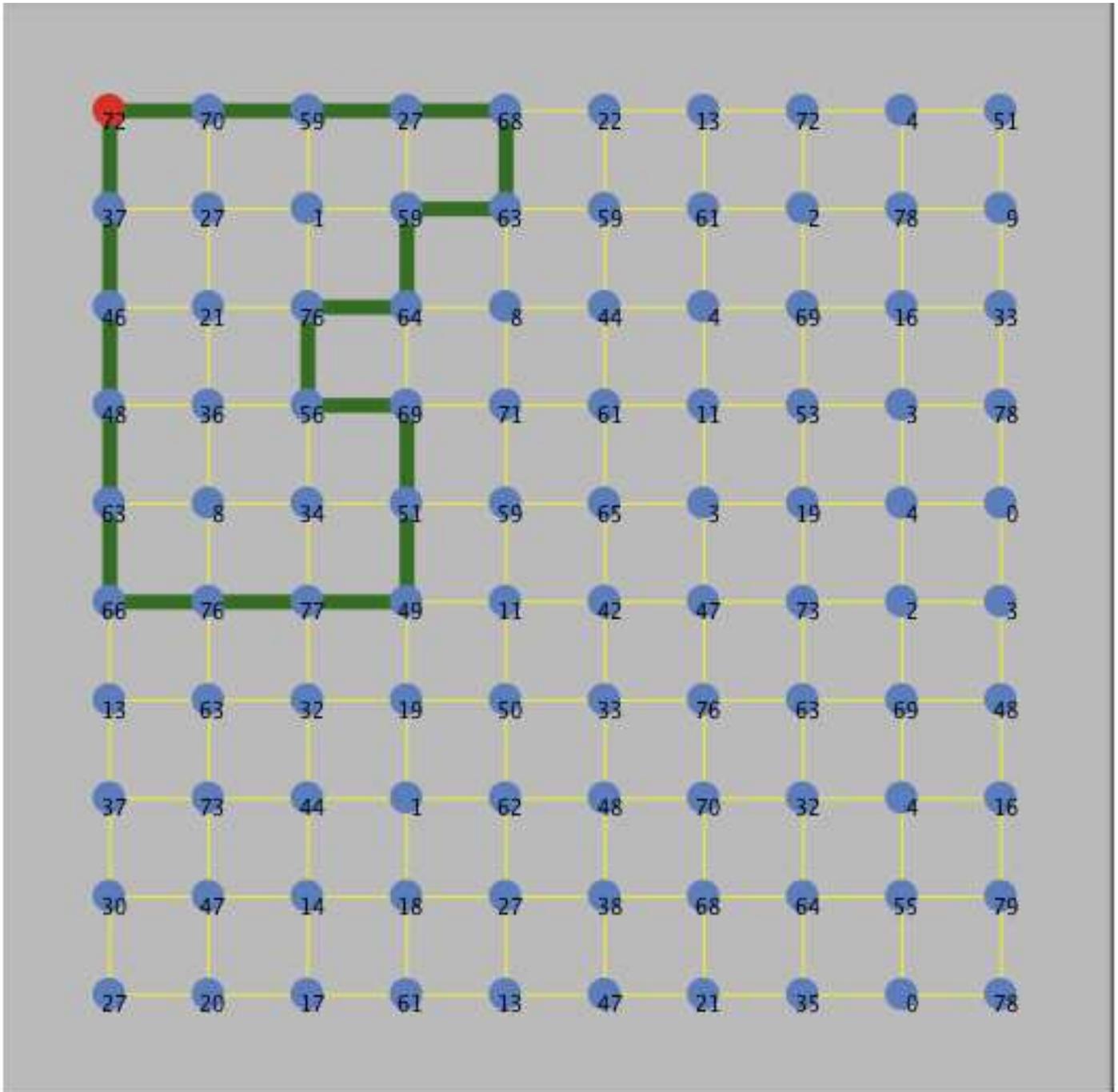


Figura 28- Parametri fissati per la simulazione e valori di output finali (max-population-load=80)

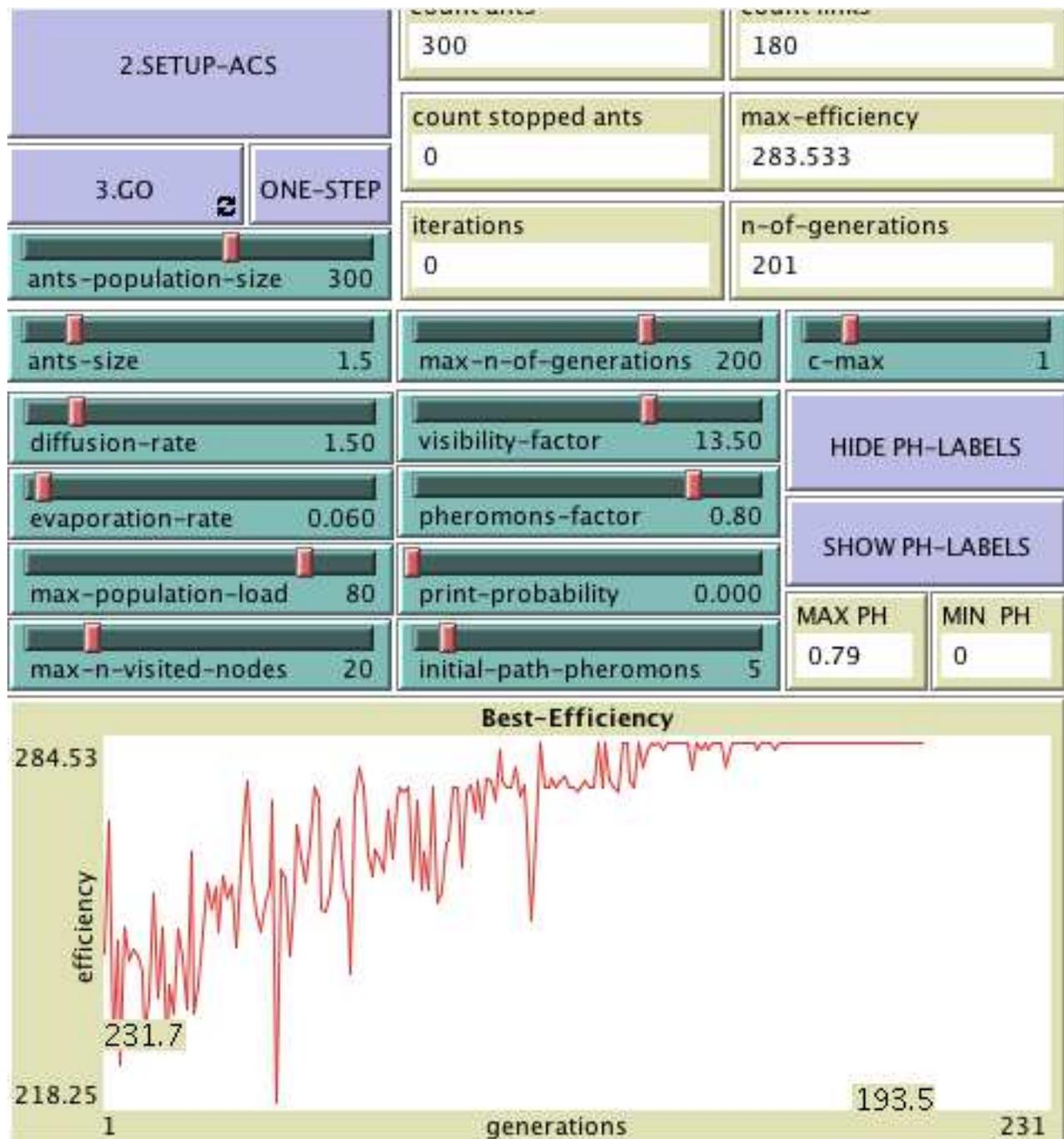


Figura 29 - Best Path (max-population-load=60): Percorso migliore definito dalla simulazione con numero massimo della popolazione in un nodo fissato a 60

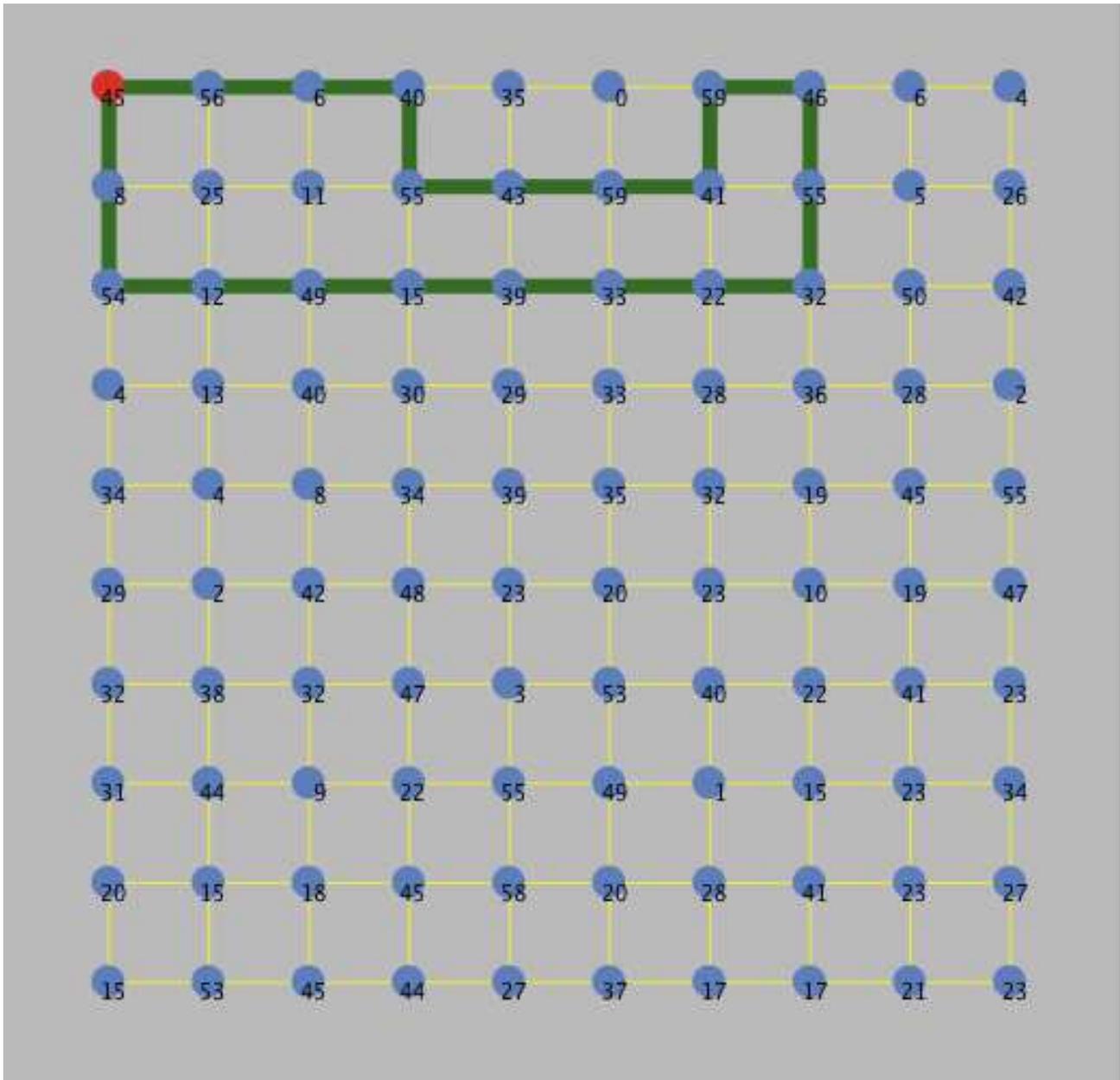


Figura 30- Parametri fissati per la simulazione e valori di output finali (max-population-load=60)

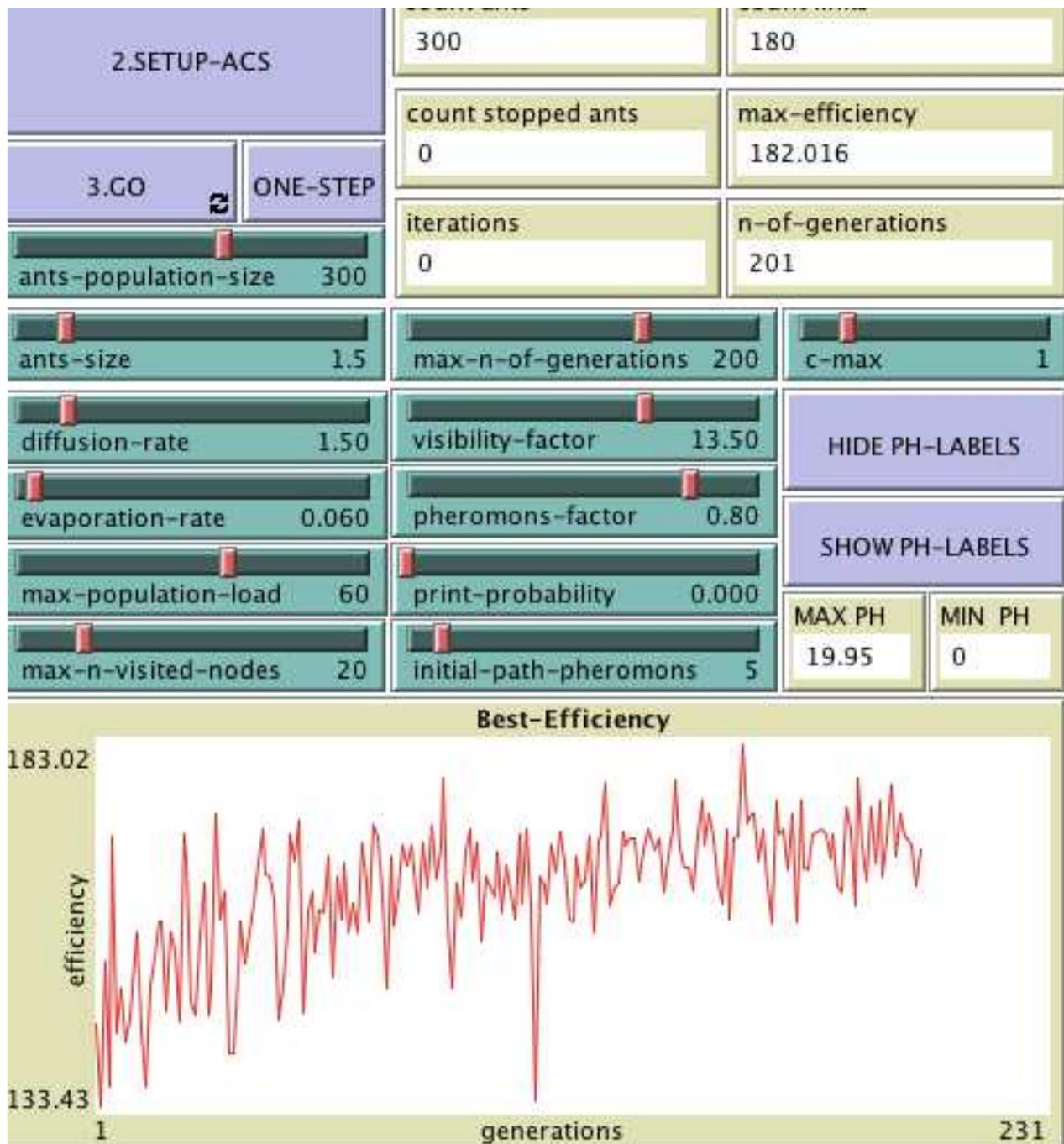
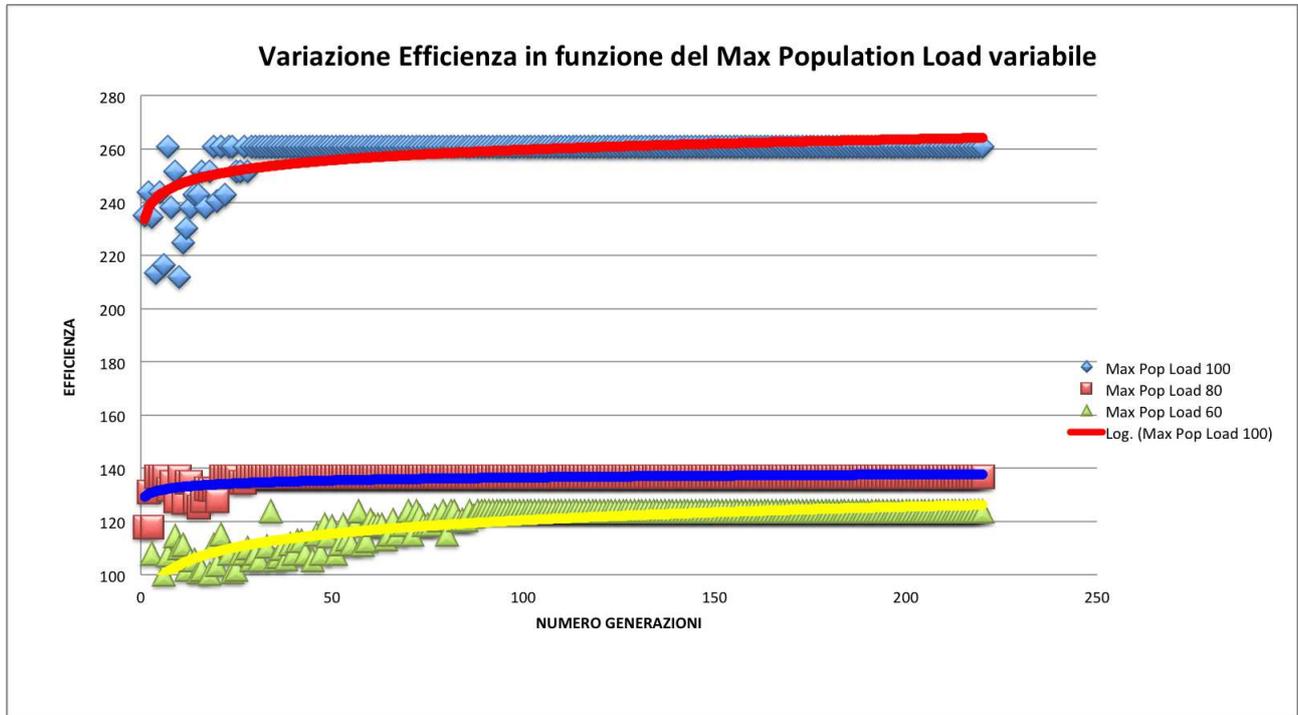


Grafico 2- Dati complessivi max-population-load



Dai valori raccolti notiamo che diminuendo la quantità di popolazione che gravita sui nodi da un valore di 100 a 80, la massima efficienza aumenta, quindi il trend raggiunto è maggiore, per poi arrivare ad un valore asintotico.

Figura 31 - Best Path (ant-population-size=300): Percorso migliore definito dalla simulazione con numero massimo della popolazione di formiche a 300

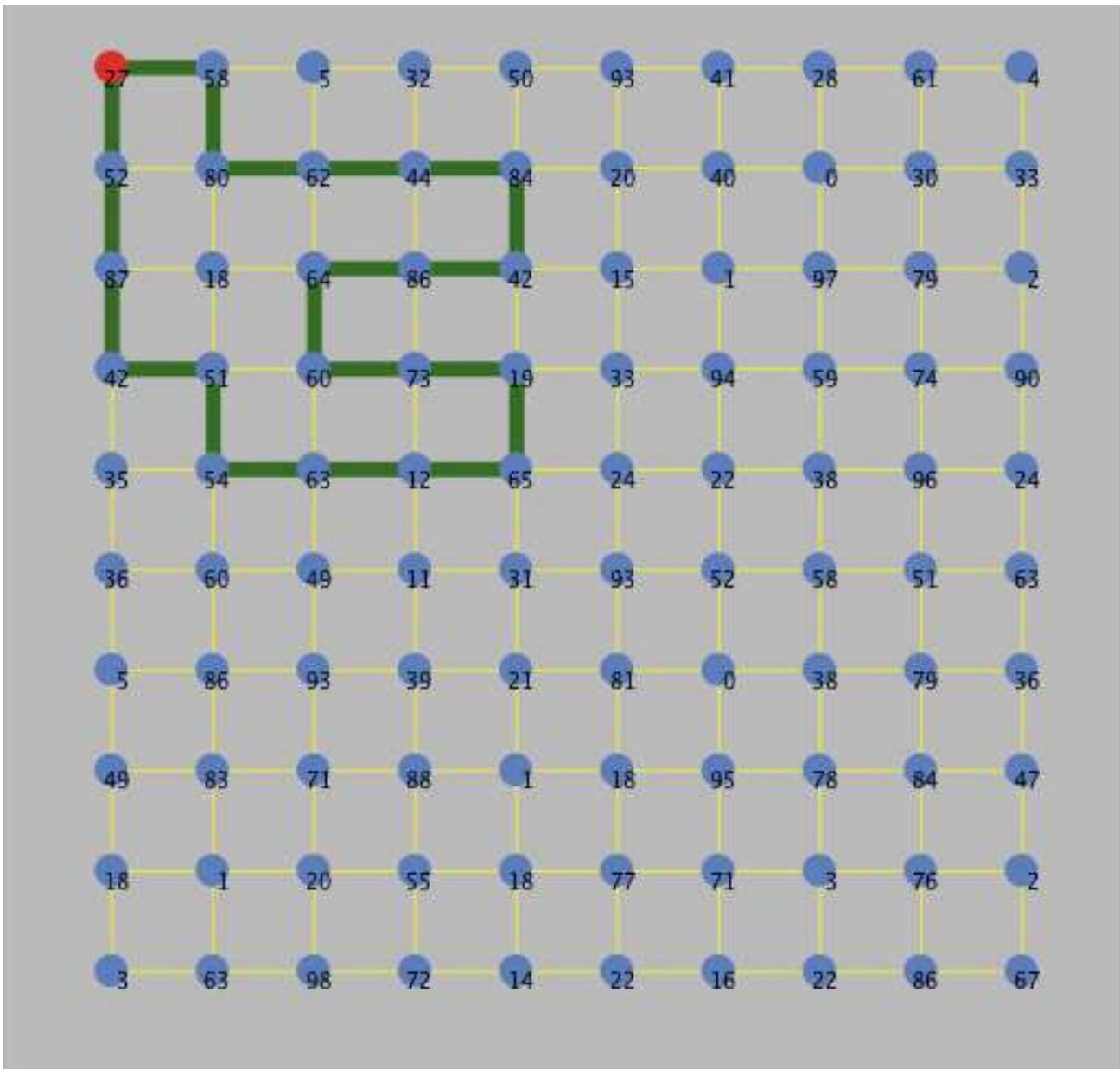


Figura 32- Parametri fissati per la simulazione e valori di output finali (ant-population-size=300)

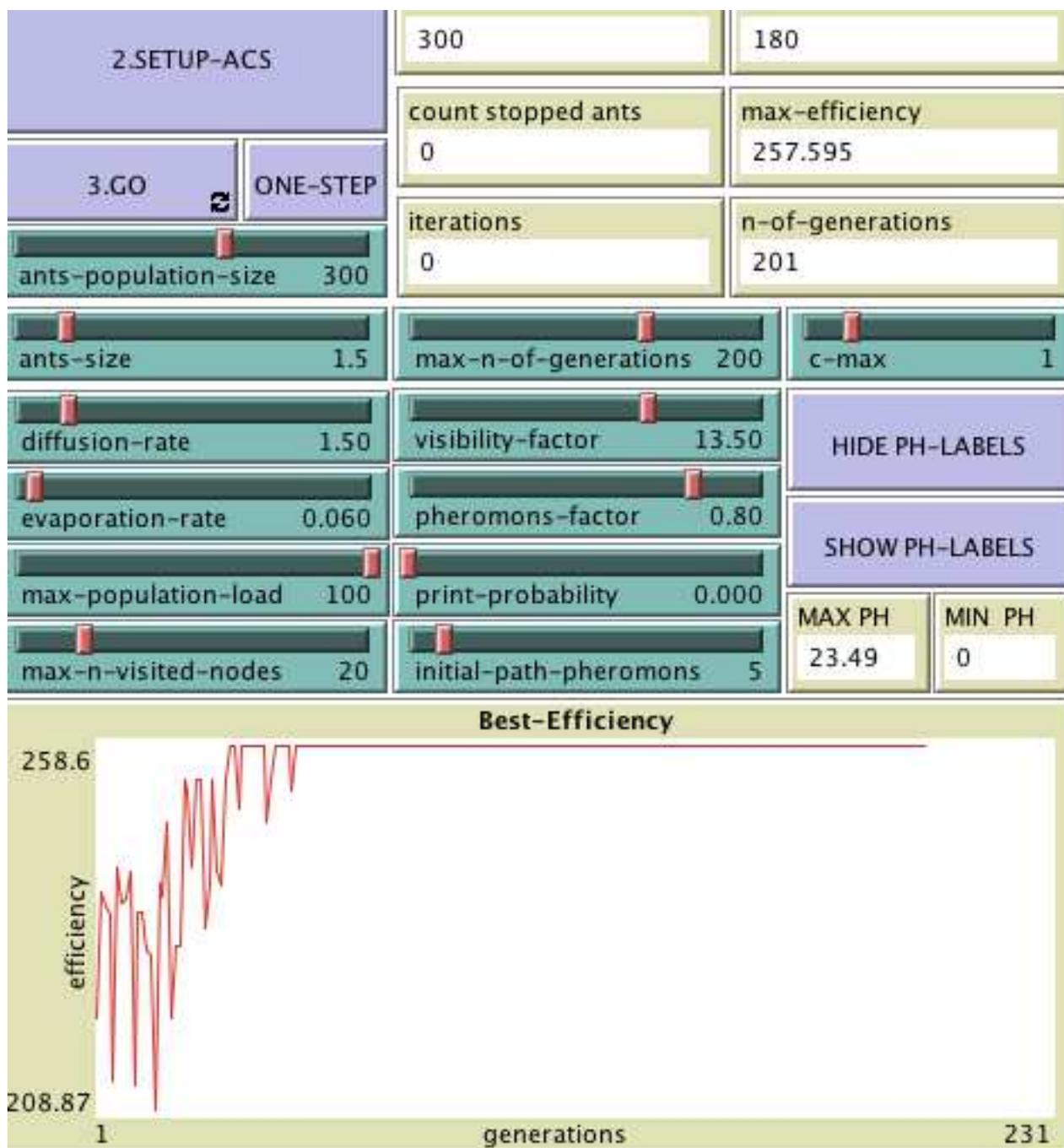


Figura 33 - Best Path (ant-population-size=400): Percorso migliore definito dalla simulazione con numero massimo della popolazione di formiche a 400

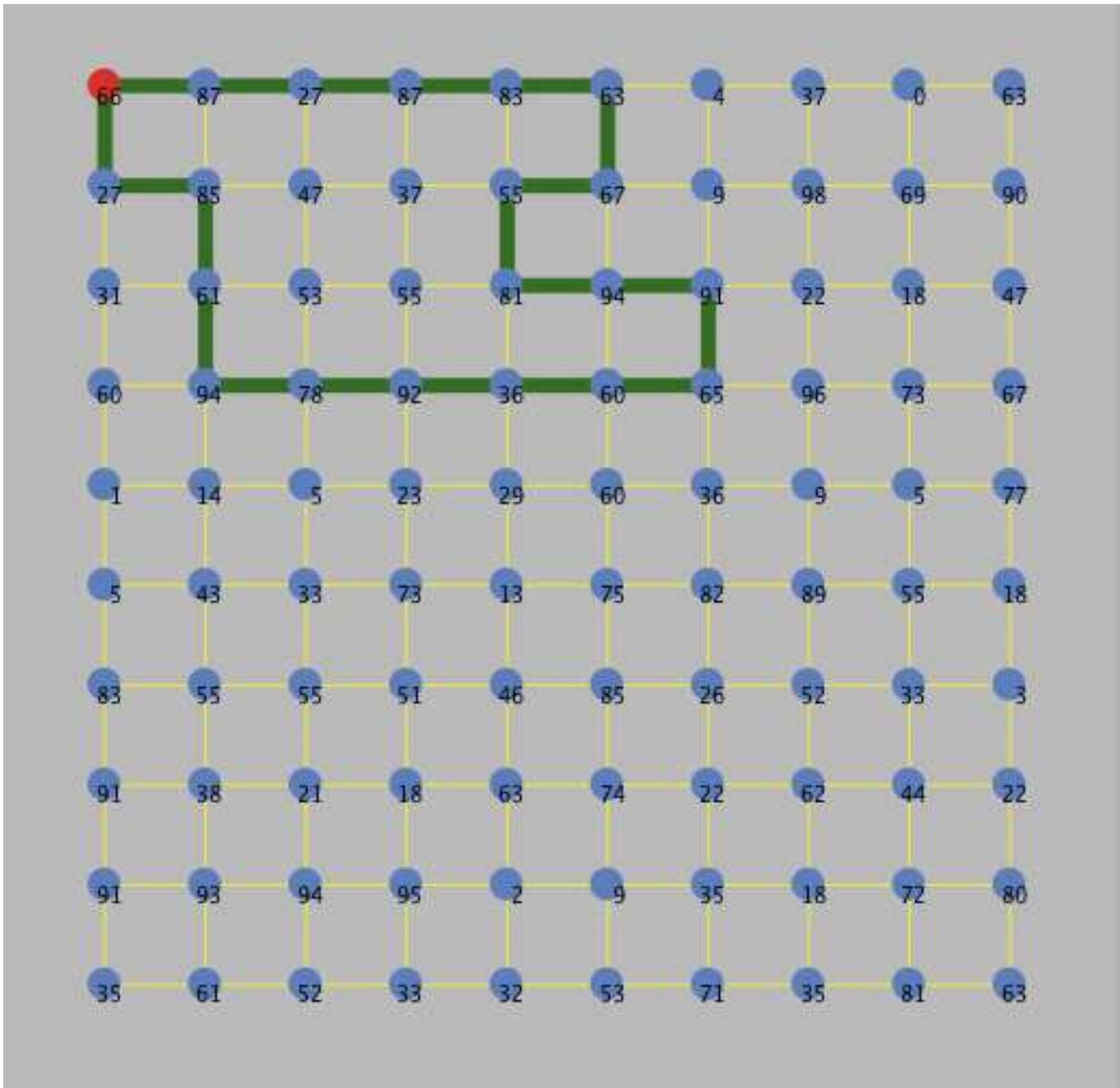


Figura 34- Parametri fissati per la simulazione e valori di output finali (ant-population-size=400)

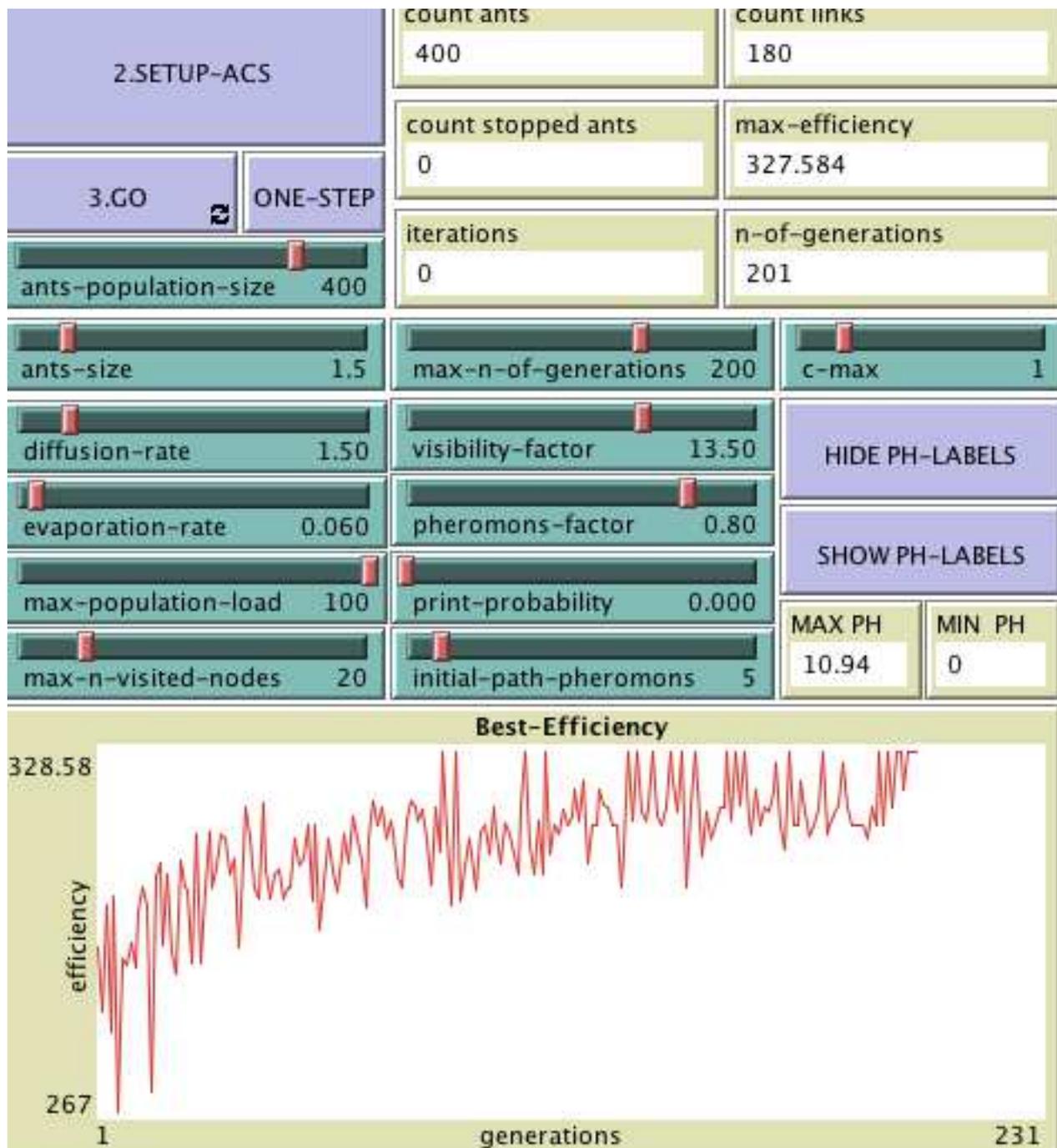


Figura 35 - Best Path (ant-population-size=200): Percorso migliore definito dalla simulazione con numero massimo della popolazione di formiche a 200

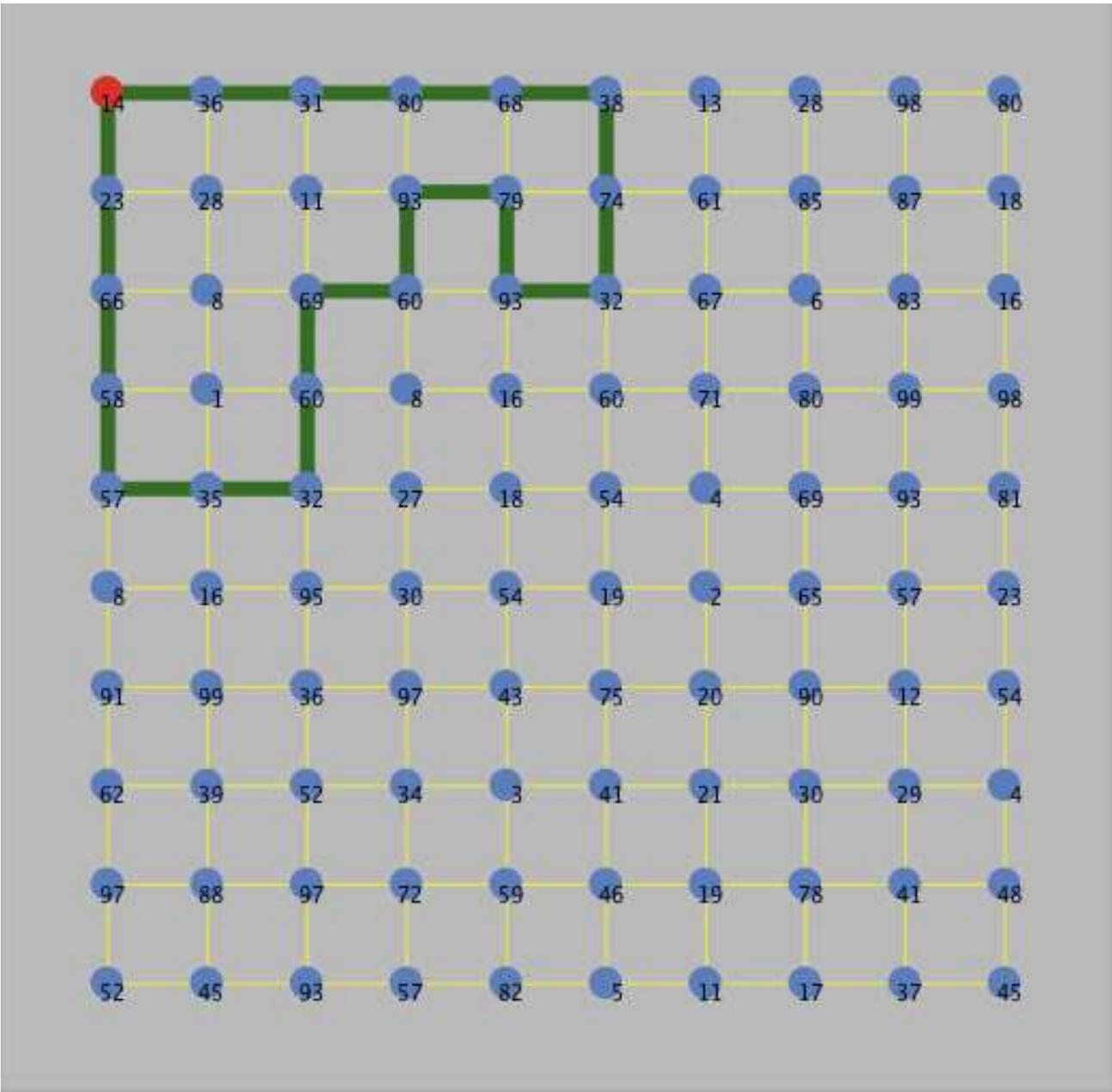


Figura 36- Parametri fissati per la simulazione e valori di output finali (ant-population-size=200)

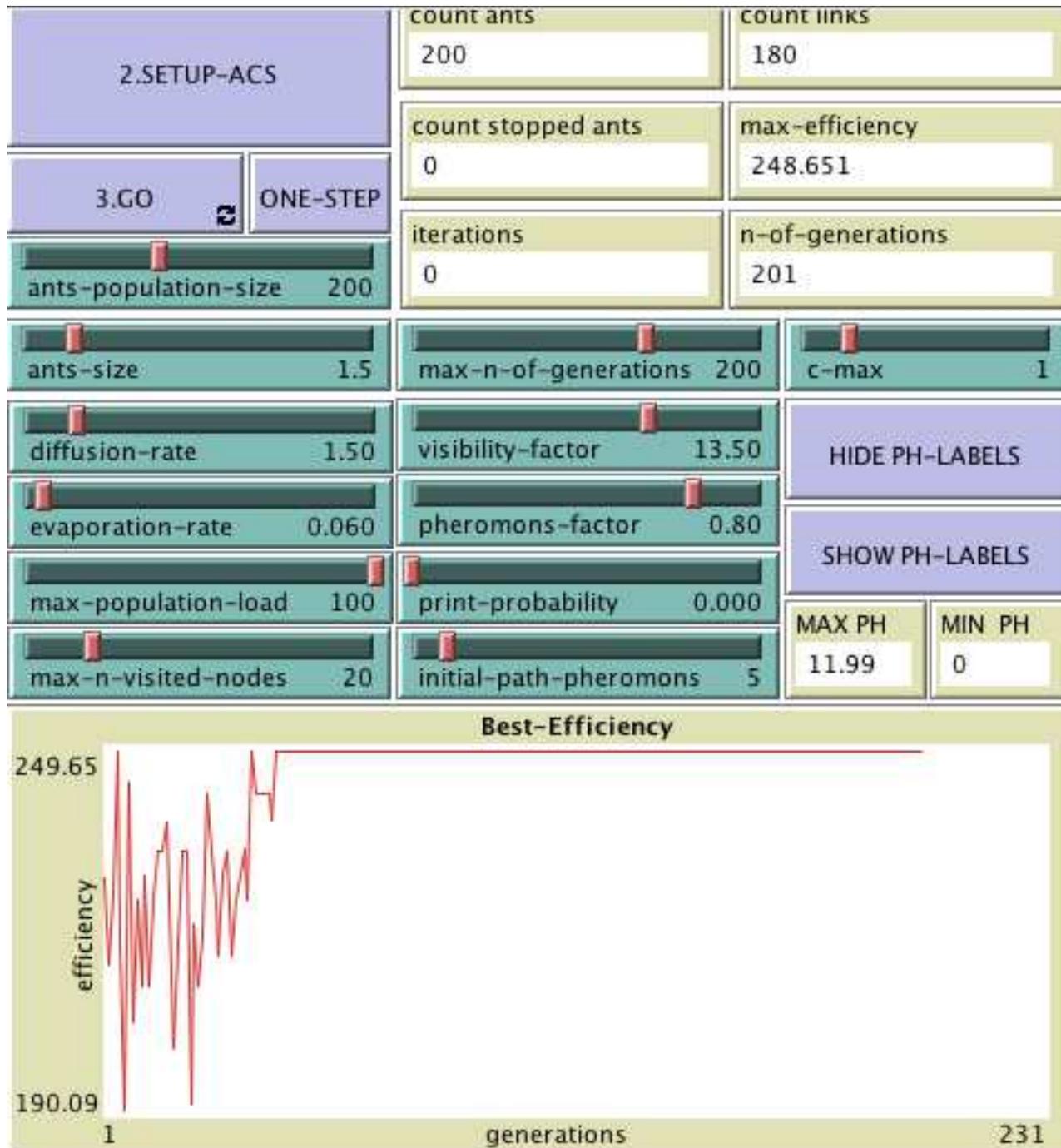
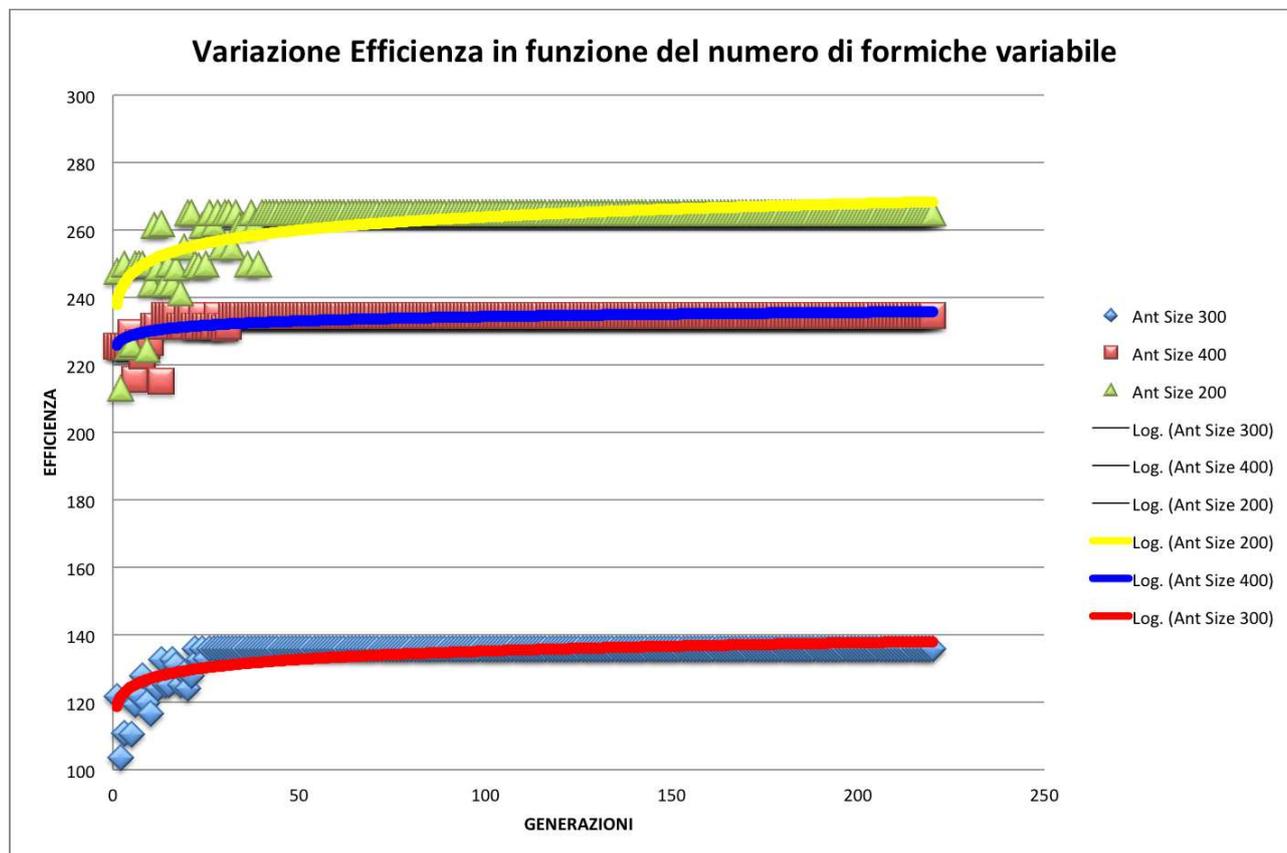


Grafico 3- Dati complessivi ant-population-size



Dai dati raccolti e dal grafico notiamo che aumentando il numero delle formiche presenti in una simulazione da 200 e 300, la massima efficienza tenderà ad un valore asintotico costante dopo un certo numero di generazioni. Al valore di 400 verrà raggiunta un'efficienza elevata con un trend crescente.

Conclusioni

Come abbiamo più volte espresso durante questo lavoro di tesi, il nostro obiettivo era quello di fornire una metodologia per la progettazione e definizione delle linee di feederaggio del trasporto pubblico attraverso l'utilizzo dell'algoritmo ACS. Scopo implicito è stato sicuramente quello di fornire i mezzi per l'edificazione di un nuovo modo con cui guardare alla mobilità, specialmente per le aree a domanda debole, in modo da creare servizi di trasporto pubblico più agevoli e alla portata di tutti.

Oltre a fornire le principali linee guida per la progettazione di una rete di trasporto collettivo, abbiamo applicato i criteri esposti nella letteratura degli algoritmi euristici in un caso ideale, per una prospettiva applicativa nella realtà.

I campi di applicabilità e gli sviluppi futuri di questo lavoro di tesi sono tra i più disparati. Oltre a poter essere applicato ad un qualsiasi contesto urbano, nazionale e internazionale, può anche essere utilizzato in sfere che vanno oltre questo scenario.

Un esempio di scenario alternativo per l'applicazione di tale tipologia è, ad esempio, quello presentato dalle aree a bassa domanda della città di Catania, in particolare dal parcheggio scambiatore Due Obelischi, sottoutilizzato per la scarsa efficienza, che purtroppo va sempre diminuendo, del mezzo di trasporto pubblico a cui il parcheggio è collegato.

Un'ulteriore applicazione nel caso catanese è collegato alla linea linea 504M Stesicoro – Benedettini e alla nuova stazione di San Nullo. La realizzazione di una linea TPL che segua i criteri esposti con l'utilizzo dell'Ant Colony System potrebbe rappresentare un motivo di miglioramento della mobilità del centro storico catanese e avvantaggiare soprattutto gli studenti del Dipartimento di Scienze umanistiche, di Scienze della Formazione e di Giurisprudenza ma anche gli studenti della cittadella universitaria, che utilizzano i trasporti pubblici, e un incentivo per diminuire l'utilizzo del trasporto privato e soprattutto i notevoli flussi di traffico che si concentrano soprattutto nelle ore di punta.

La costruzione di linee di feederaggio rappresenta, quindi, per il territorio catanese una proposta operativa per la rinascita del trasporto pubblico cittadino, che possa consentire spostamenti veloci e sicuri.

Bibliografia

1. Coppola P., *Il servizio di trasporto collettivo*, Università degli Studi di Roma Tor Vergata, Marzo 2014.
2. Consiglio Nazionale degli Ingegneri, *Reti Urbane di Trasporto: Linee guida per l'analisi ed il progetto*, Marzo 2009.
3. Giuseppe Inturri, *Sistemi di trasporto collettivo*, Università degli Studi di Catania, Gennaio 2017.
4. Giuseppe Inturri, *Alcuni criteri di progetto dei sistemi di trasporto collettivo*, Università degli Studi di Catania, Gennaio 2017.
5. Montella B., D'Acerno L., *Metodologie di progettazione dei servizi di trasporto pubblico locale nelle aree a domanda debole*, Dicembre 2009.
6. Hame L., *Demand Responsive Transport: Models and Algorithms*, Aalto University, Maggio 2012.
7. Amisano F., Cassone A., Marchese C., *Trasporto pubblico locale e aree a domanda di mobilità debole in Provincia di Alessandria*, Università del Piemonte "Amedeo Avogadro", Alessandria, Maggio 2011
8. Coppola P., *Modelli di progetto delle reti di trasporto collettivo*, Gennaio 2009.
9. Pranzo M., *Appunti sul Vehicle Routing Problem*, Febbraio 2008.
10. Aldà M., *Algoritmi Euristici per il Vehicle Routing Problem*, Luglio 2004.
11. Baiocchetti M., *Algoritmi euristici ed evolutivi per l'ottimizzazione*, Università degli Studi di Perugia, Febbraio 2009.
12. Pluchino A., *Introduzione alla programmazione di simulazione ad agenti*, Università degli Studi di Catania, Aprile 2012
13. Chivilikhin D., *Ant Colony Optimization: An Introduction*, Aprile 2013.
14. Okpedia : *Algoritmo ACO* (https://www.okpedia.it/algoritmo_aco).
15. Wikipedia: *Algoritmo delle colonie di formiche* (https://it.wikipedia.org/wiki/Algoritmo_delle_colonie_di_formiche).
16. Catay B., *Ant Colony Optimization and its application to the Vehicle Routing Problem with Pickups and Deliveries*, Marzo 2015.
17. Pellonpera T., *Ant colony optimization and the vehicle routing problem*, University of Tampere, Maggio 2015.
18. Rizzoli A.E., Montemanni R., Lucibello E., Gambardella L.M., *Ant colony optimization for real-world vehicle routing problems*, Springer Science + Business Media, Settembre 2007.
19. Mehr H. S., Rahnama M. R., Shokouhi M. A., Mafi E., *Optimization of Main Public Transport Paths Based on Accessibility— Case Study: Mashhad, Iran*, University of Mashhad, Febbraio 2011.

Dichiarazione

Il contenuto di questo documento riflette il punto di vista dell'allievo laureando ingegnere, che è responsabile per la natura e l'accuratezza delle informazioni riportate.

Questo documento è diffuso con il patrocinio del Dipartimento di Ingegneria Civile ed Architettura dell'Università degli Studi di Catania con la finalità prevalente di favorire lo scambio di informazioni e di conoscenze. Il Dipartimento e il relatore non si assumono alcuna responsabilità per i contenuti del documento.

Le opinioni, i risultati e le conclusioni del documento sono quelle espresse dall'allievo e non sono necessariamente condivise dal Dipartimento e dal relatore.