

# Sistemi di imprese e di imprese-lavoratori che coevolvono, con jESOF

Pietro Terna

Dipartimento di Scienze economiche e finanziarie G.Prato, Università di Torino  
pietro.terna@unito.it

## 1. INTRODUZIONE

jESOF, *java Enterprise Simulator Open Foundation*, evoluzione del simulatore di impresa jES; è uno strumento di simulazione in grado di gestire in parallelo molti modelli, allo scopo di studiare l'evoluzione di più sistemi tra loro interconnessi. jESOF e jES si fondano su Swarm ([www.swarm.org](http://www.swarm.org)), sia come biblioteca base di funzioni per la simulazione, sia come protocollo per la costruzione dei modelli.

I modelli qui presentati pongono al centro dell'attenzione l'interazione tra componenti ed hanno la doppia valenza di rappresentare esempi illustrativi del metodo della simulazione ad agenti, applicabile in modo semplificato tramite jESOF, e di individuare primi risultati che offrono materiale di riflessione anche per una interpretazione normativa di questo tipo di ricerche.

L'interazione è fondata su relazioni produttive, spaziali e di fiducia. I sistemi oggetto di studio evolvono con la modificazione delle interazioni tra le parti, consentendo l'emergenza di fenomeni grazie alla flessibilità della simulazione ad agenti. Si tratta infatti di una metodologia che non richiede assunzioni a priori invasive rispetto alla dinamica che l'esperimento artificiale svilupperà, aderendo così alle caratteristiche di uno schema di vita artificiale. Schema che invece difficilmente può essere riprodotto con sistemi ad equazioni o con la simulazione di processo ad eventi discreti, intrinsecamente più rigida, in quanto sequenza di eventi e di operazioni conseguenti.

Una prima applicazione, semplice, riguarda un sistema di imprese interagenti che devono organizzarsi per colmare i vuoti produttivi presenti nel sistema a causa della mancanza di competenze che a volta a volta si manifestano per alcune fasi della lavorazione necessaria per la produzione.

Una seconda applicazione estende il modello all'interazione tra sistema del lavoro e sistema delle imprese, con proprietà emergenti in relazione alla distribuzione delle competenze professionali dei lavoratori. Si sottolinea che due popolazioni che coevolvono, con interazioni puntuali tra i due sistemi, non sono trattabili agevolmente se non con modelli di simulazione ad agenti.

## 2. LO STRUMENTO: DA JES A JESOF

Si introduce brevemente jES (*java Enterprise Simulator*: si veda anche Terna, 2002), affiancato da *jES Open Foundation* o jESOF (versione multi-modello del precedente) come metodologia completa di simulazione ad agenti. I codici informatici di jES e di

jESOF sono disponibili all'indirizzo <http://web.econ.unito.it/terna/jes>. Sia jES sia jESOF sono accompagnati da documenti tecnici di riferimento: rispettivamente *How to Use the Java Enterprise Simulator (jES) Program* e *How to Use the jES Open Foundation Program to Discover Complexity in Multi-Stratum Models*.

L'analogia più illuminante per comunicare scopo e contenuti di jES è quella dei simulatori di volo disponibili per i nostri computer. Esattamente come un simulatore di volo riproduce gli effetti di ciò che noi "giocatori" comunichiamo al computer, generando tutte le interazioni che il complesso modello dell'aereo in volo richiede, jES riproduce le conseguenze degli ordini, contenenti le "ricette" da eseguire, che sono immessi nel modello, con tutte le interazioni che ne conseguono nella struttura descritta; struttura di una singola azienda, di un sistema di aziende; di un sistema di entità di diversa natura.

Se chi interagisce con il simulatore di volo commette errori, l'aereo precipita; se sceglie la migliore traiettoria di volo possibile, l'aereo compie perfettamente e nel modo migliore il percorso richiesto: dunque il simulatore riproduce le conseguenze delle scelte, non propone scelte. Allo stesso modo jES non è un ottimizzatore per le scelte di impresa; non è un gestore degli ordinativi, né uno schedatore, secondo il gergo gestionale tratto dall'inglese.

Con il modello invece si intende "far funzionare" l'azienda o l'organizzazione simulata che opererà nel nostro computer con l'azione indipendente di tutte le componenti; nel modello gli eventi accadono in modo indipendente, generando interazioni anche imprevedibili tra atti produttivi e unità produttive, secondo le modalità proprie dei contesti in cui si sviluppa la complessità. Numerose applicazioni sono presentate in Terna *et al.* (2006).

Nei modelli realizzati con jES e jESOF esiste una netta distinzione tra cose da fare e chi le fa. Possiamo introdurre un mini dizionario e una semplice classificazione.

Mini dizionario del modello:

1. unità: una struttura produttiva, eventualmente corrispondente ad una cellula produttiva in una struttura più vasta; una unità è in grado di svolgere almeno uno dei passi indicati nelle ricette;
2. ordine: un oggetto rappresentante un bene da produrre, merce o servizio, oppure una generica azione da compiere, ogni ordine contiene le informazioni necessarie per realizzarlo, in una ricetta;
3. ricetta: una sequenza di passi, espressi con un formalismo numerico, da compiere per realizzare un ordine; si fa riferimento alla ricetta di cucina (*recipe*, in inglese), come prototipo della descrizione dei componenti, delle quantità, delle sequenze, necessari per ottenere un risultato.

Classificazione delle componenti di un modello:

1. che cosa fare: lato del modello che descrive gli ordini da eseguire, contenenti le ricette, con un adeguato formalismo;

2. chi fa che cosa: lato del modello che descrive le unità in grado di eseguire i passi delle ricette contenute negli ordini, con adeguato formalismo;
3. la sequenza temporale degli accadimenti: anche in questo caso con adeguato formalismo.

Sulla base del dizionario e della classificazione che precedono, è ora agevole chiarire che ogni applicazione con jES o jESOF deve gestire una triplice fonte di informazioni: le ricette, le unità e il lancio degli ordini contenenti le ricette.

Un punto molto importante da chiarire per jES e per jESOF è quello dell'origine degli ordini: casuale (utilizzando un "generatore di ordini casuali") o programmata (utilizzando un "distillatore di ordini").

Il generatore di ordini funziona sulla base di tre informazioni fornite dall'utente: i numeri che contraddistinguono i passi produttivi utilizzabili per costruire le ricette (le fasi di lavorazione delle unità); la lunghezza delle ricette, da 1 al massimo indicato; le ripetizioni per ogni singolo passo, per variare le durate esecutive, da 1 al massimo indicato.

(Se si vogliono lanciare ordini con passi non eseguibili da nessuna delle unità esistenti, occorre utilizzare, vedi sotto, la strada del distillatore di ordini, che li estrae da un archivio: in jES un ordine con un passo non eseguibile da nessuna unità produce un errore, trattandosi di una situazione incoerente; in jESOF invece la situazione è gestita).

Esempi di ricette, con unità che dispongono di un dizionario di fasi produttive {1, 2, 4, 7, 9}, lunghezza massima 4, ripetizioni massime 2:

"1 1 7 2 2" - tre passi con due ripetizioni;

"9 7 1 4" - quattro passi, senza ripetizioni;

"2" - un solo passo, senza ripetizioni;

"2 2 2 2" - due passi con due ripetizioni, oppure quattro passi senza ripetizioni, oppure ... .

Il "distillatore di ordini" consente invece la produzione di ordini programmata utilizzando un repertorio di ricette, fondandosi su una sequenza di eventi distinta in due possibili parti (entrambe devono essere scritte; eventualmente quella non utilizzata riporterà soltanto una riga vuota): si tratta di una sequenza iniziale che sarà seguita una sola volta e di una sequenza continua, che sarà eseguita ripetutamente, dopo quella iniziale. E' ovvio che se quella iniziale è vuota, ci sarà solo l'esecuzione ripetitiva e viceversa. La simulazione può essere arrestata o manualmente da noi, con il bottone *Stop*, oppure ad un ciclo predefinito.

Passando a jESOF, lavoriamo in un mondo normalmente a più strati o livelli, ciascuno con suoi ordini (e relative ricette), con sue unità e con un generatore di ordini o un distillatore di ordini. Le fasi produttive descritte nelle ricette sono valide per tutti gli strati e le ricette, spostandosi da una unità ad un'altra possono cambiare strato. Pertanto una ricetta che preveda le fasi 3, poi 33, poi 2 può muoversi dallo strato 2 in

cui c'è una unità in grado di svolgere la fase produttiva 3, allo stesso strato in cui un'altra unità svolge la fase 33 e poi andare allo strato 0 dove una unità svolge la fase 2.

Come descritto in fig. 1, ogni strato della simulazione corrisponde ad un modello, generato secondo il protocollo Swarm citato all'inizio. Ogni unità possiede una matrice e l'ambiente nel suo insieme possiede matrici generali; tutte le matrici sono accessibili durante la simulazione, tramite i passi computazionali.

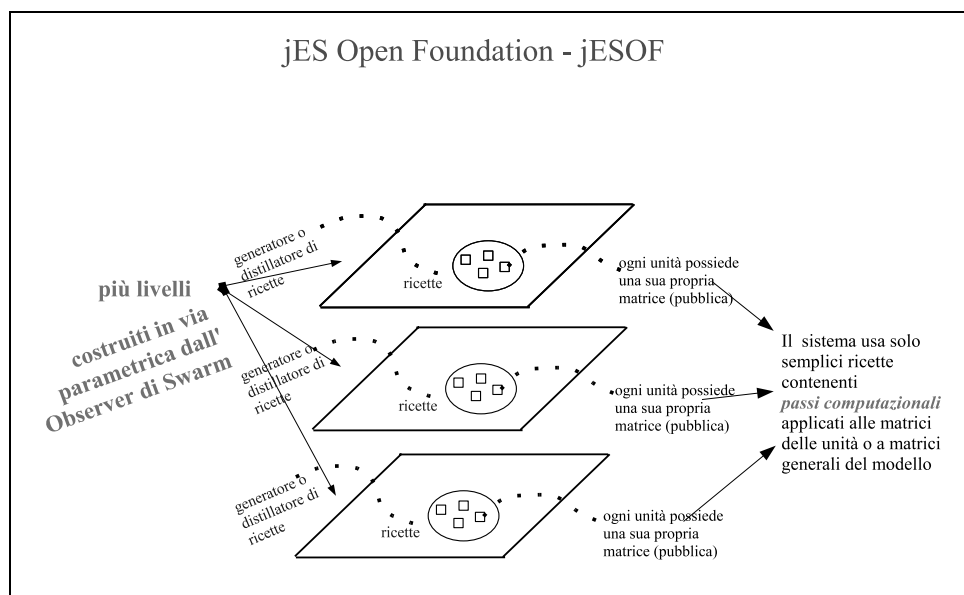


Fig. 1. Il mondo multistrato delle simulazioni con jESOF

I passi computazionali sono specificazioni contenute nelle ricette che, nel momento in cui sono in una unità (esattamente nel momento in cui avranno terminato la fase di lavorazione prevista in quella unità), lanciano una sequenza di elaborazione specificata in un metodo Java; l'elaborazione non richiede altre informazioni oltre a quelle contenuto nelle matrici dette.

In jESOF è prevista la rappresentazione degli spazi degli strati, come dislocazione spaziale, propria o metaforica espressa in qualche metrica, delle unità; gli spazi possono essere considerati come toroidali. Su quelle superfici le unità sono rappresentate con delle sbarre (se richieste) di istogrammi verso sinistra, sotto, verso destra, con vari significati di rappresentazione dello stato di ciascuna unità; possono inoltre essere contornate da un'area costante o crescente di "visibilità": è possibile introdurre l'opzione che solo le unità che hanno in comune, anche su strati diversi, uno o più punto della superficie di visibilità, sono tra loro intervisibili. Le ricette, con i relativi passi computazionali, possono in questo caso muoversi solo tra unità che siano tra loro reciprocamente visibili: il significato di questi spazi di visibilità può essere un

indice della distanza entro cui si collabora in un distretto geograficamente stabilito, una rappresentazione metaforica dei legami di fiducia in una rete anche non connessa fisicamente, ecc.

Seguono ora due esperimenti descritti con riferimento alla metodologia enunciata, che consente di descrivere, come si farà qui di seguito, in modo analitico, le ipotesi costruttive dei modelli, senza che sia necessario fare riferimento al codice informatico sottostante.

### 3. CICLO ECONOMICO ED INTERAZIONE TRA IMPRESE

Con jESOF si presenta un esperimento di evoluzione di imprese che cooperano nella produzione sotto la condizione di reciproca visibilità o fiducia, determinando un ciclo economico plausibile. Per riprodurre l'esperimento si utilizzi il contenuto della cartella apps/firstEvolApplication nella distribuzione di jESOF. L'applicazione si esegue utilizzando i file con nome formato da un numero, in particolare per riprodurre gli esempi menzionati si debbono usare i file '5.2' e '5.3' (si tratta di file eseguibili).

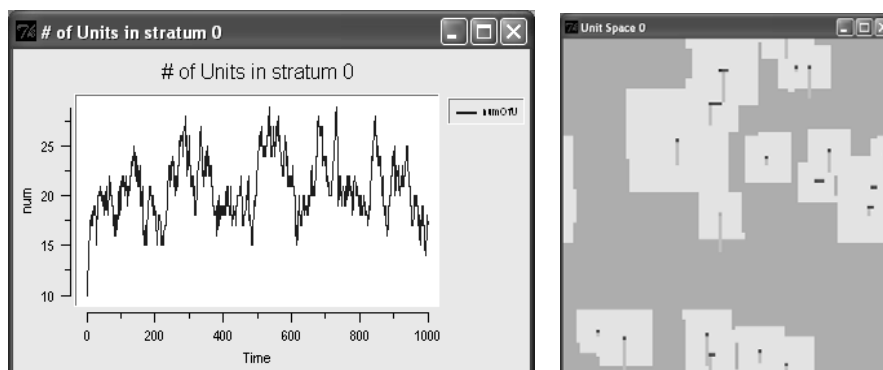


Fig. 2. Il ciclo produttivo e la configurazione spaziale delle imprese nel caso 5.2.

Le unità produttive sono inizialmente 10 al massimo, trattandosi di unità di 5 tipi diversi (la diversità corrisponde alla fase della produzione che l'unità sa trattare, da 1 a 5), e sono potenzialmente due per tipo, con probabilità di creazione iniziale delle unità pari a 0,8. Ad ogni ciclo nasce una nuova unità, di uno dei 5 tipi, con probabilità 0,8. Le unità nascono con uno spazio di visibilità compreso casualmente tra 0 e 200 punti intorno ad esse (il mondo è formato da 100x100), con una crescita di 10 punti per ciclo della simulazione. Due unità scambiano se le loro aree di visibilità hanno almeno un punto in comune. Gli spazi di visibilità sono rappresentati in grigio chiaro nelle figg. 2 e 3.

Il generatore casuale di ordini produce ordini, uno per ciclo, con ricette costruite con il dizionario delle 5 fasi produttive che le unità sanno trattare; le ricette sono lunghe da 1

a 5 passi e ciascun passo può richiedere 1 o 2 cicli per essere svolto; la ricetta più breve richiede un ciclo per essere elaborata, la più lunga, 10.

Ogni unità è visualizzata dal simulatore con tre barre (eventualmente assenti) che sono utilizzate per rappresentare i seguenti stati interni dell'unità: la barra a sinistra indica la coda di attesa di prodotti, che l'unità ha ricevuto, dal generatore di ordini o da altre unità e che elaborerà nei cicli successivi, uno per ciclo; la barra in basso indica il numero di cicli in cui l'unità è stata inattiva, senza ordini da lavorare; non appena svolge un ciclo di lavorazione, la barra è azzerata; la barra a destra indica il numero di prodotti che l'unità non riesce a consegnare ad altre unità per la lavorazione, per mancanza di visibilità (se la ricerca è ultimata, il prodotto è semplicemente cancellato dall'unità, nella presunzione che ad ogni ordine corrisponde un acquisto).

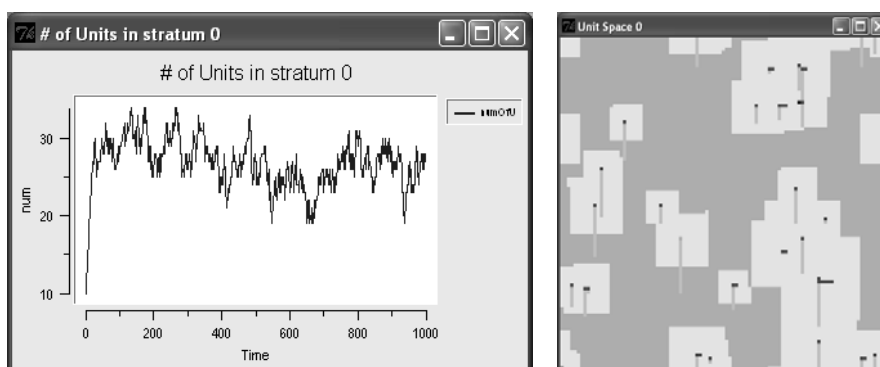


Fig. 3. Il ciclo produttivo e la configurazione spaziale delle imprese nel caso 5.3.

Se una unità non ha ordini da lavorare per 10 cicli, o non riesce a consegnare più di 10 prodotti, esce dalla produzione; in entrambi i casi tutti prodotti presenti nell'unità sono persi. 10 cicli di inattività totale possono essere fatti corrispondere a tre mesi di inattività totale nel mondo reale, come congettura ragionevole di sopravvivenza di un'azienda con totale assenza di lavoro. I 1000 cicli della simulazione sono quindi pari a 100 trimestri, cioè 25 anni.

Lo spazio di visibilità rappresenta sia la fiducia conquistata restando sul mercato, sia l'ampliarsi territoriale delle relazioni. La prima grandezza in particolare spiega la formazione dei distretti.

I risultati della simulazioni (fig. 2) mostrano come, utilizzando i parametri illustrati, si formano ampie aree produttive e si producono oscillazioni marcate, da 15 a 30 unità circa, del numero delle imprese attive, con 5 cicli economici nell'arco dei 25 "anni" considerati, il che è plausibile da un punto di vista empirico.

I risultati riportati in fig. 3, invece, si basano su una sopravvivenza di 15 cicli in caso di inattività totale, simulando un intervento a più alto rischio del sistema bancario, ma

si imposta anche una maggiore rigidità del sistema introducendo una crescita della visibilità di 5 punti a ciclo invece di 10. Il risultato è una netta stabilizzazione del ciclo.

Uno sviluppo interessante del modello è l'introduzione esplicita del sistema bancario, per valutare gli effetti sul bilancio delle banche della seconda politica, potenzialmente a più elevato rischio.

Già comunque nell'impostazione qui presentata il modello consente di valutare il ruolo della fiducia e del capitale sociale (Zuckerman, 2003) non solo nella costruzione dei sistemi di relazioni da cui dipendono i distretti, ma anche nella formazione del ciclo economico, in quest'ultimo caso accanto al ruolo delle istituzioni finanziarie.

Dal punto di vista della relazione tra modello di simulazione e analisi normativa, il risultato emergente pare di grande interesse, suggerendo l'adozione di misure che consentano la crescita della fiducia e della collaborazione nel sistema produttivo come intervento di politica economica.

#### 4. UN MODELLO DI COEVOLUZIONE TRA IL SISTEMA DEL LAVORO E QUELLO DELLE IMPRESE

Si introduce ora la coevoluzione di due sistemi: quello delle imprese e quello dei lavoratori, presentato in quattro versioni. I file relativi si trovano nella distribuzione di jESOF, in quattro cartelle con nome simile tranne il carattere finale: apps/workers\_skills\_firms/workers\_skills\_firms\_v\_1/ (oppure finale 2).

Nel simulatore opera un generatore casuale di ordini che accompagna lo strato 4, con la produzione di ordini, uno per ciclo, contenenti ricette fondate sul dizionario a tre attività coerente con le unità esistenti e potenziali; le ricette sono lunghe da 1 a 5 passi e ciascun passo può richiedere 1 o 2 cicli per essere svolto; come sopra, la ricetta più breve richiede un ciclo per essere elaborata; la più lunga 10. Nel simulatore opera anche un distillatore di ordini, che accompagna lo strato 0, ma opera anche con riferimento alle unità degli altri strati, 4 compreso.

In tutto abbiamo 5 strati, con cinque modelli dal punto di vista Swarm, di cui importanti sono il primo (numero 0) e l'ultimo (numero 4). Gli strati 1, 2 e 3 sono dedicati alla visualizzazione dei sottogruppi di lavoratori con competenza professionale 1, 2 o 3; i lavoratori sono anche rappresentati tutti insieme nello strato 0. Le imprese, che popolano lo strato 4, sono di 3 tipi (coincidenti con la fase di lavorazione che sono in grado di compiere) e sono inizialmente in numero di 3 per tipo con probabilità di generazione pari a 0,8; nei cicli successivi la probabilità di nascita di una impresa ad ogni ciclo è 0,2; le imprese sopravvivono senza lavoro per 15 cicli e possono al massimo accumulare 10 prodotti che non riescono a consegnare. Se i 15 cicli sono fatti corrispondere a 3 mesi come nel paragrafo precedente, i 3000 cicli della simulazione equivalgono a 200 trimestri, quindi a 50 anni.

I lavoratori, che popolano lo strato 0, sono anch'essi di tre tipi, corrispondenti alla rispettiva competenza professionale; inizialmente sono 50 per tipo con probabilità di

creazione 1; è anche prevista la generazione di un individuo a ciclo - con probabilità 0,9 – che però non rappresenta la fonte importante di presenza di nuovi lavoratori nello strato, dovuta invece a ordini di nascita di lavoratori, indirizzati a quelli esistenti, con la ricetta che indica la comparsa di un nuovo soggetto. Questi ordini di duplicazione delle unità rappresentanti i lavoratori sono lanciati in numero di 10 per tipo di attività in ogni ciclo. Il passo computazionale utilizzato può essere di due tipi: la copia rappresentante il nuovo lavoratore nello strato zero (e che avrà anche una copia di rappresentazione in uno degli strati 1, 2 oppure 3) avviene in prossimità del lavoratore copiato oppure in una posizione a caso: ciò riproduce una situazione di concentrazione o di diffusione di competenze. Gli ordini di copia, che si realizzano con probabilità 0,9, possono essere ricevuti più volte dalla stessa unità-lavoratore in ogni ciclo.

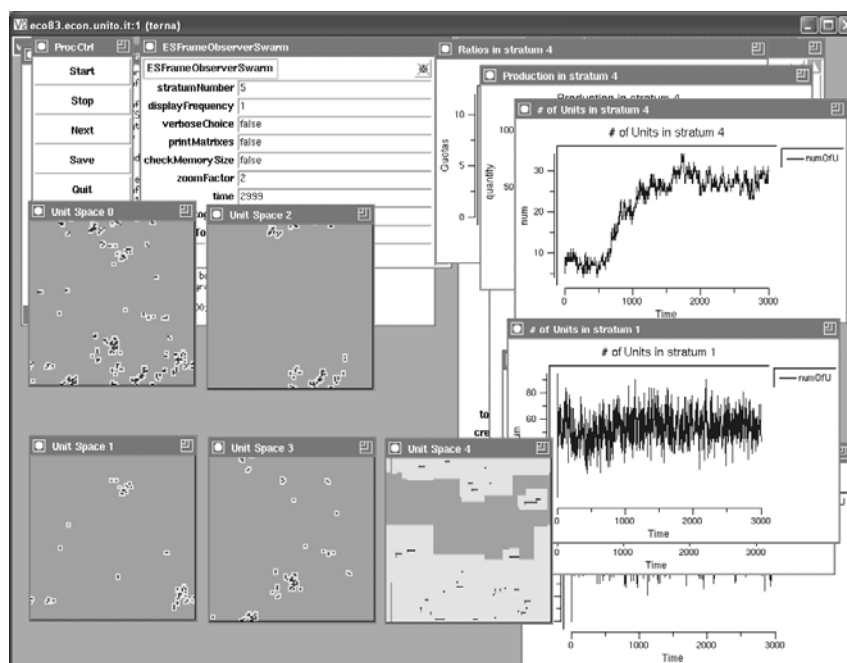


Fig. 4. Versione 1 della simulazione lavoratori-imprese.

Le aree di visibilità delle imprese hanno una dimensione iniziale generata a caso tra 0 e 200 punti, in un mondo formato da 100x100 punti, e si incrementano di 0,5 punti ad ogni ciclo. Quelle dei lavoratori sono invece fisse, di 8 punti (quindi le otto posizioni adiacenti l'unità-lavoratore, se si raffigura il tutto su una carta a quadretti).

Il legame impresa-lavoratori è realizzato lanciando 10 ordini per tipo di impresa con una ricetta che prevede che il primo passo sia svolto da un'impresa e il secondo da un lavoratore dotato del profilo professione indicato nella ricetta stessa e coerente con la fase produttiva della unità-impresa che ha svolto il primo passo; il lavoratore è individuato solo se visibile dall'impresa, vale a dire se almeno un punto della sua area di visibilità (di 9 punti, quello dell'unità più 8) si interseca con l'area di visibilità



dell'impresa. Quando l'ordine arriva all'unità lavoratore opera un passo computazionale che aggiunge una unità nella matrice di memoria dell'impresa, nella posizione in cui si conteggiano i lavoratori assunti, e una unità nella matrice di memoria del lavoratore, nella posizione in cui si conteggiano le giornate di lavoro. L'assunzione dura un ciclo di calcolo e poi deve essere ripetuta. Come sopra per gli ordini che inducono la comparsa di nuovi lavoratori, anche in questo caso una impresa o un lavoratore possono ricevere più di un ordine nello stesso ciclo (essendo più elevata la numerosità dei lavoratori, questo secondo effetto è poco probabile).

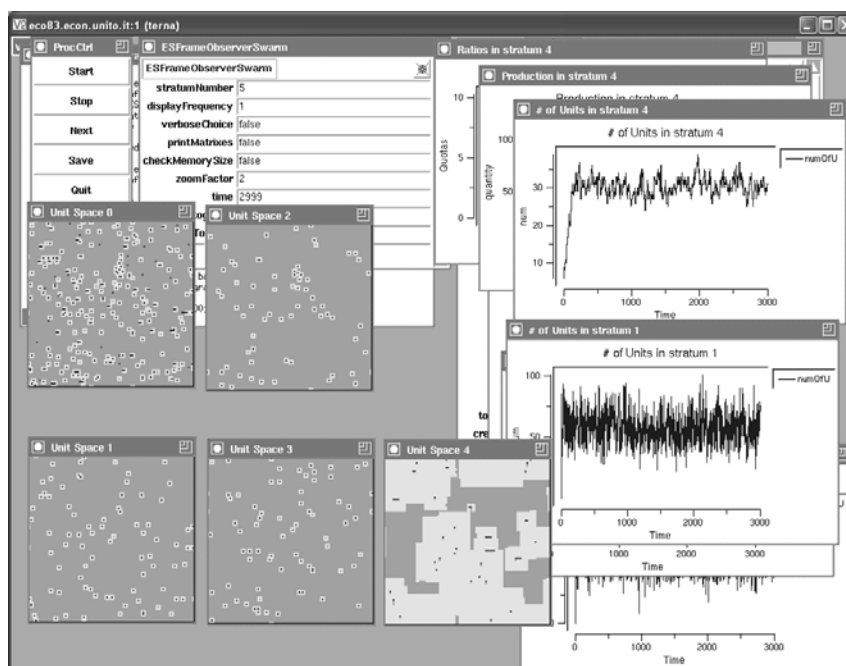


Fig. 5. Versione 2 della simulazione lavoratori-impreses.

Ad ogni ciclo è quindi lanciato un ordine di utilizzazione di lavoro per ciascuno dei tre tipi di unità-impresa. Qualsiasi sia l'impresa che lo riceve (a caso), il passo computazionale contenuto si applica a tutte le imprese similari, con probabilità 1,0. Il passo computazionale opera sottraendo una unità dalla casella della matrice di memoria di ogni unità, dove sono totalizzati i lavoratori assunti, determinando il "consumo di lavoro" da parte dell'impresa; se il totalizzatore scende al di sotto di -10, l'unità-impresa è cancellata. Analogamente per le unità-lavoratore accade lo stesso tipo di elaborazione, ma la soglia al di sotto della quale avviene la cancellazione è -5. Il meccanismo di interazione tra imprese e lavoratori è quindi piuttosto allentato e prevede che si utilizzi forza lavoro comunque, anche se l'unità-impresa non ha lavoro da svolgere, il che non è privo di realismo.

La differenza costitutiva delle versioni qui presentate è la seguente:

- versione 1: i nuovi lavoratori sono collocati accanto alla posizione originaria del lavoratore copiato, secondo una diffusione delle competenze per prossimità, con un orientamento professionale per imitazione; la disposizione delle unità di lavoro nella fig. 4 è diretta conseguenza di questa impostazione; le imprese formano un unico distretto collegandosi al suo interno alle isole di competenze di lavoro esistenti; il numero delle imprese aumenta molto lentamente e si stabilizza soltanto in un tempo lungo, di 30 anni sui 50 della simulazione;
- versione 2: i nuovi lavoratori sono collocati casualmente nello spazio; la disposizione delle unità di lavoro nella fig. 5 è diretta conseguenza di questa impostazione; la localizzazione delle imprese è più articolata; il numero delle imprese aumenta rapidissimamente e si colloca in una posizione stazionaria in circa 5 anni dei 50 della simulazione.

Le conclusioni in termini di orientamento professionale, deducibili da questo esperimento particolarmente stilizzato, sono relative alla rischiosità di una eccessiva concentrazione spaziale di competenze di lavoro.

In tutte le versioni elencate compare anche un ciclo complesso relativo all'andamento del numero dei lavoratori; sulla spiegazione dei cicli nel mercato del lavoro, come sfida non risolta, si vedano i recenti lavori di Hall (2005) and Shimer (2005).

Si ripete anche per questo modello la valutazione di carattere normativo, in quanto grazie a questi risultati di simulazione abbiamo una verifica sperimentale artificiale della fragilità di sistemi troppo concentrati su poche competenze professionali e quindi produttive, con un indebolimento della tradizionale visione del distretto come concentrazione di specializzazioni e quindi di efficienza. In termini di dinamica di sviluppo ciò risulta invece negativo secondo il modello, per una minore robustezza strutturale di quel tipo di organizzazione produttiva. Ciò apre ampi spazi di ricerca, sulla sperimentazione della dinamica degli attuali distretti, anche con riferimento, come nel caso precedente, alle interazioni con il sistema finanziario.

#### BIBLIOGRAFIA

HALL R.E. (2005), *Employment Fluctuations with Equilibrium Wage Stickiness*, «American Economic Review», 1, XCV, pp. 50-65.

SHIMER R. (2005), *The Cyclical Behavior of Equilibrium Unemployment and Vacancies*. «American Economic Review», 1, XCV, pp. 25-49.

TERNA P. (2002), *Simulazione ad agenti in contesti di impresa*, in «Sistemi intelligenti»>, 1, XIV, pp. 33-51.

TERNA P., BOERO R., MORINI M., SONNESSA M. (2006), *Modelli per la complessità – La simulazione ad agenti in economia*. il Mulino, Bologna.

ZUCKERMAN E. (2003), *On Networks and Markets by Rauch and Casella, eds.*, in «Journal of Economic Literature», 3, XLI, pp. 545-565.