

A perspective from the past to the future

○

A tool for huge ABMs

○○○

The scheme of an ABM

Timing and Results

00

Final remarks

○○

Looking at Repast4Py

Eleonora Priori, Pietro Terna

IRES, online—November 17th, 2022

A perspective from the past to the future

○○○

A tool for huge ABMs

三

The scheme of an ABM

Timing and Results

00

Final remarks

00

A perspective from the past to the future

The past

The future

A tool for huge ABMs

Repast4Py

The scheme of an ABM

The basics

The code

Timing and Results

Final remarks

A letter from the past

Siro Lombardini to Ferdinando Di Fenizio, 1962, 1/3

Ires ISTITUTO RICERCHE ECONOMICO-SOCIALI « ALDO VALENTE »
Torino
via Reggio 55, tel. 44 663
via Maria Fittoria 18

Torino, 26 gennaio 1962

16 DISCUSSION

Chiarissimo
Prof. Ferdinando di Fenizio
Via Appiani, 1
M. I. I. A. N. O.

M I L A N O

Caro Professore.

è stata ultimata la valutazione dei parametri del modello che l'IRES ha concepito ed elaborato, per l'analisi delle prospettive di sviluppo dell'economia torinese, nel quadro di un complesso di ricerche svolte, oltre che dall'IRES, dalla Société d'Economie et de Mathématique Appliquées di Parigi e dall'Italconsorti di Roma. Stiamo ora utilizzando il modello per l'analisi delle implicazioni delle diverse ipotesi sullo sviluppo delle imprese motrici e delle industrie autonome e delle diverse congettive sugli andamenti delle altre grandezze considerate esogene.

In attesa che lo studio sia compiuto, ci è gradito inviarLe la prima stesura della prima parte della ricerca (ora ultimata), nella speranza di conoscere le Sue osservazioni critiche e i Suoi suggerimenti che contiamo di utilizzare nella messa a punto definitiva dello studio. (Il dattilo scritto è ancora a circolazione privata).

La ringraziamo vivamente in anticipo per questa Sua preziosa collaborazione e La preghiamo di gradire il nostro cordiale saluto.

Prothomisus

Siro Lombardini to Ferdinando Di Fenizio, 1962, 2/3

INDICE

CAPITOLO I

- INTERDIPENDENZE SPAZIALI ED INTERDIPENDENZE SETTORIALI NELLE ANALISI REGIONALI

1. Considerazioni introduttive all'analisi delle prospettive economiche di una regione pag. 1
 2. Alcuni criteri per la delimitazione della regione pag. 4
 3. Interdipendenze settoriali e interdipendenze spaziali pag. 5
 4. Relazioni tra l'attività industriale e l'attività agricola pag. 8

CABITOL 9-11

- SCHEMA CONCETTUALE PER LA CLASSIFICAZIONE DEI SETTORI

1. Il settore industriale. Le industrie autonome pag. 14
 2. Le imprese motrici e le imprese complementari..... pag. 17
 3. Il commercio pag. 18
 4. La valutazione degli scambi del commercio con l'estero
e dei flussi interni del settore pag. 23
 5. Investimenti autonomi ed indotti pag. 26
 6. Gli effetti degli investimenti e la loro analisi pag. 29
 7. L'attività finanziaria pag. 36
 8. Le famiglie: i loro redditi e i loro consumi pag. 38

CAPITOLO III

- IL MODELLO -

1. I settori, le variabili e le relazioni del modello pag. 42
 2. La matrice dei flussi per la provincia di Torino pag. 48
 3. Considerazioni introduttive per la determinazione delle diverse variabili del modello pag. 52
 4. Il sistema di equazioni del modello pag. 57
 5. Il sistema di equazioni per la regione di Torino pag. 62

Siro Lombardini to Ferdinando Di Fenizio, 1962, 3/3

UN MODELLO ECONOMETRICO PER L' ANALISI DELLE PROSPETTIVE DI SVILUPPO

CAPITOLO I

INTERDIPENDENZE SPAZIALI ED INTERDIPENDENZE SETTORIALI NELLE ANALISI REGIONALI

1. Considerazioni introduttive all'analisi delle prospettive economiche di una regione

Le prospettive di sviluppo dell'economia di una regione dipendono:

- a) dalle tendenze spontanee quali risultano dall'analisi dell'attuale struttura nell'ipotesi di costanza dei coefficienti tecnici, delle preferenze dei consumatori e delle caratteristiche socio-istituzionali;

b) dagli sviluppi dell'economia di altre regioni o paesi;

c) dagli interventi delle Pubbliche Amministrazioni;

d) da eventi esogeni (come il progresso tecnico, i mutamenti nei gusti, le possibilità e le prospettive della politica sindacale, ecc.).

Diverse sono le modalità dell'analisi a seconda delle ipotesi che vengono formulate circa gli interventi della Pubblica Amministrazione:

- a) Si può assumere che gli interventi si svolgeranno secondo le modalità già in atto. Si tratta allora di un'analisi economica in senso stretto.

A program toward the future

From the PROGRAMMA ANNUALE DI RICERCA DELL'IRIS PIEMONTE 2022

LINEA DI RICERCA TRASVERSALE - F: INNOVAZIONE E COMPETITIVITÀ DEL SISTEMA PRODUTTIVO PIEMONTESE



SCHEDA DELL'ATTIVITÀ N. 30

STUDI PREPARATORI PER UN MODELLO STOCK-FLUSSI CONSISTENTE PER SCENARI MACROECONOMICI REGIONALI MICRO-FONDATI

La modellistica ad agenti si sta da tempo affermando come uno dei formalismi più promettenti per l'analisi dei fenomeni socioeconomici e, oltre il livello accademico, questo approccio si sta diffondendo anche nei centri studi delle istituzioni e degli istituti di credito. I vantaggi che derivano dall'applicazione di questa metodologia sono diversi ed ampiamente documentati in letteratura. Fra questi è particolarmente rilevante la possibilità di sviluppare delle simulazioni di sistemi reali "a partire dal basso" per tener conto di almeno due aspetti che si riscontrano in tutti i sistemi socioeconomici: l'eterogeneità degli agenti, sia sotto il profilo delle loro dotazioni sia rispetto alla varietà dei loro comportamenti, e l'interazione fra soggetti eterogenei a diversa scala (micro-meso-macro). Non meno rilevante è la possibilità di poter tener conto della dimensione territoriale. Questa metodologia si rivela quindi particolarmente versatile nei suoi impieghi, come ad esempio la valutazione ex-ante degli impatti di alcune policy, lo studio delle dinamiche territoriali, l'analisi scenariale della macroeconomia regionale, l'analisi *just-in-time* su specifici settori produttivi o aggregati sociali.

OBIETTIVI: Nel 2022, si propone di avviare gli studi preparatori per la specificazione di un modello multi-agente stock-flussi consistente che sia rappresentativo della dinamica del sistema socioeconomico piemontese. Nella fattispecie si procederà alla progettazione di un modello di base che preveda l'interazione domanda-offerta tra imprese, famiglie ed istituti di credito, con la prospettiva di poter integrare successivamente moduli specifici per gli altri settori dell'economia, fra cui la bilancia commerciale. L'attività preparatoria terrà in particolar modo conto della georeferenziazione e della possibilità d'integrazione con altri strumenti di modellazione già presenti in istituto, fra cui il modello PROMETEIA per l'analisi d'impatto settoriale, il modello PIEMOD per la microsimulazione fiscale ed il modello STRUDEL per l'evoluzione demografica.

METODO DI LAVORO: Analisi della più recente letteratura specializzata in campo economico e di esperienze simili a livello nazionale ed internazionale. Definizione della scala territoriale. Predisposizioni di un prototipo. Raccolta, sistematizzazione e analisi delle informazioni cross-section e time-series disponibili per alimentare e calibrare il modello.

A perspective from the past to the future



A tool for huge ABMs



The scheme of an ABM



Timing and Results



Final remarks



URLs (1/2)

Repast

<https://repast.github.io/repast4py.site/index.html>



Repast installation source

<https://pypi.org/project/repast4py/>



repast4py 1.0.5b1
pip install repast4py

URLs (2/2)

WL in GitHub

<https://github.com/terna/winners-losers>



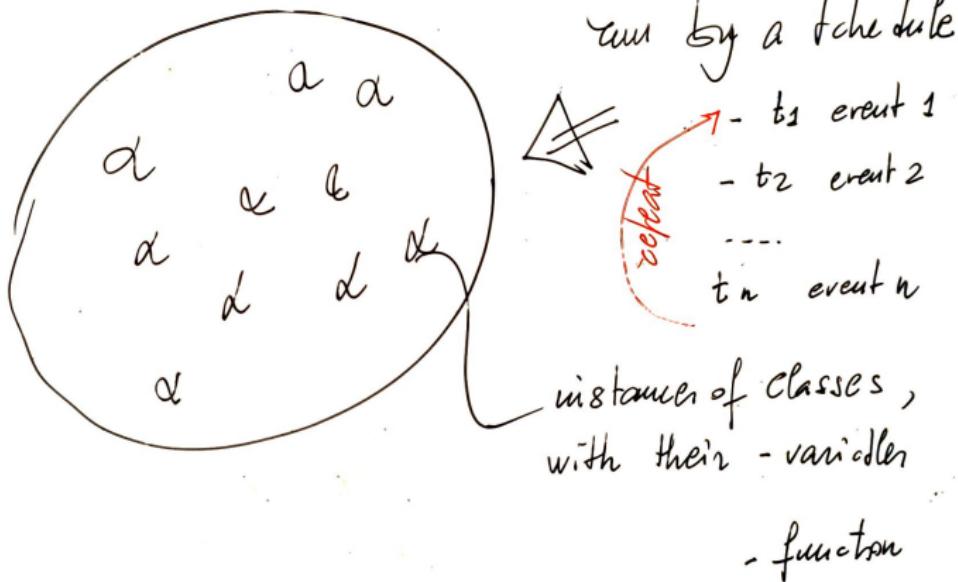
WL in viewer

<https://nbviewer.org/github/terna/winners-losers/blob/main/winners-losers.ipynb>

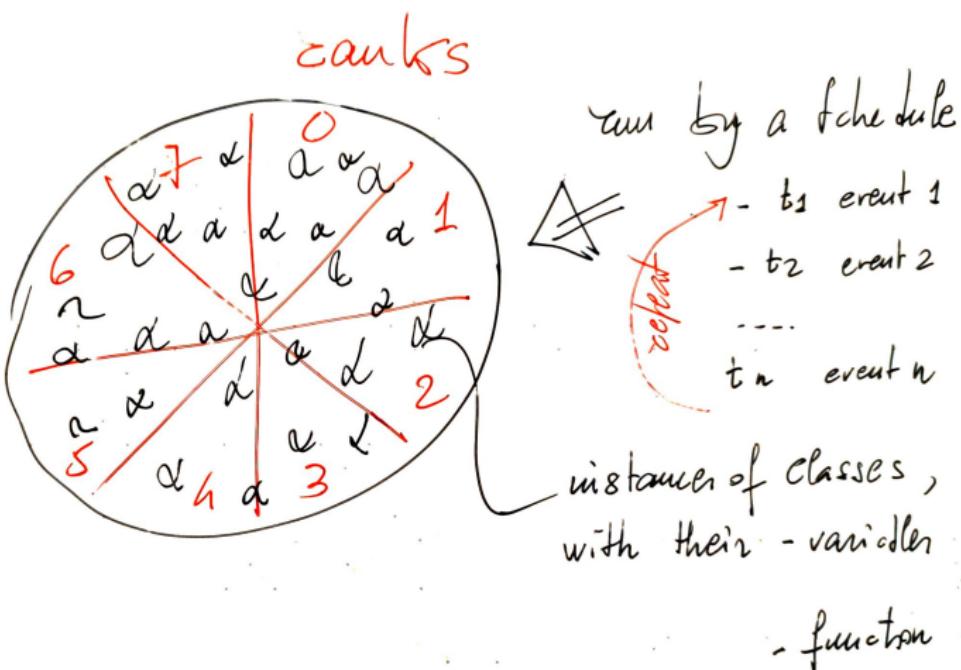




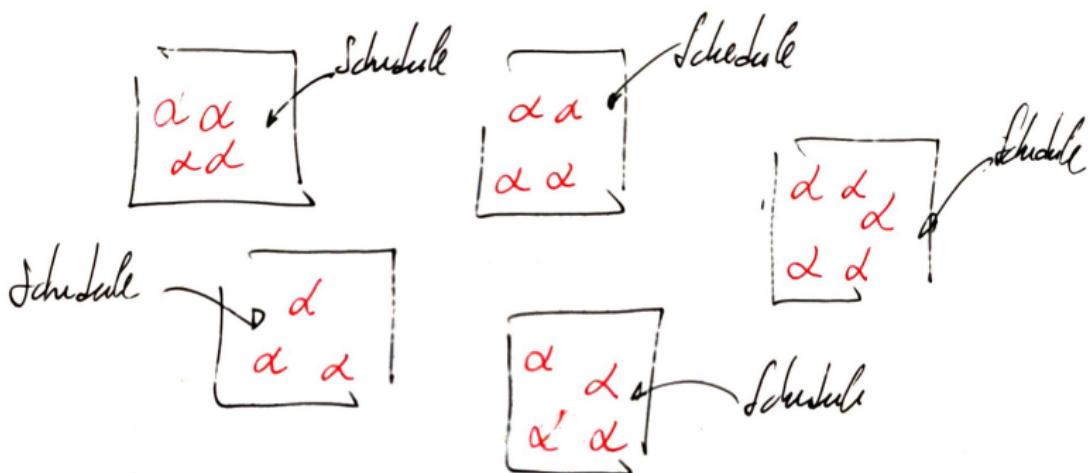
Agents

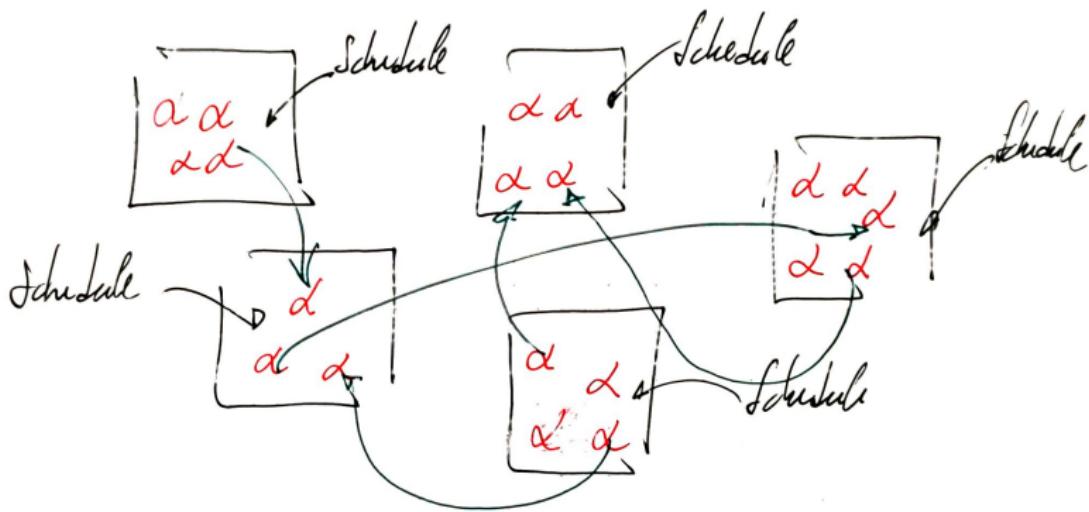


Agents and ranks (1/3)



Agents and ranks (2/3)





The model

class Model:

三

The Model class encapsulates the simulation, and is responsible for initialization (scheduling events, creating and the grid the agents inhabit IF ANY), and the overall behavior of the model.

Args:

params: the simulation input parameters

三

```
global params  
PARAMS = params
```

```
def __init__(self, params: Dict):
```

The schedule / runner

```

runner.schedule_repeating_event(0,      1, self.counter)
runner.schedule_repeating_event(0.1,    1, self.agentsChoosingCounterpart)
runner.schedule_repeating_event(0.11,   1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.12,   1, self.agentsExchangingInTheirRanks)
runner.schedule_repeating_event(0.2,    1, self.sync)
runner.schedule_repeating_event(0.21,   1, self.ghostsExchangingInDifferentRanks)
runner.schedule_repeating_event(0.22,   1,\                                self.agentsHavingExchangedWithGhostsPreparingTheirOwnGhosts)
runner.schedule_repeating_event(0.23,   1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.24,   1, self.messengerGhostsReportingOccuredExchange)
"""

schedule_stop(at)
Schedules the execution of this schedule to stop at the specified tick.

Parameters
at (float) ? the tick at which the schedule will stop.
"""
runner.schedule_stop(params['stop.at'])

runner.schedule_end_event(self.finish)

```

agentsChoosingCounterpart

```
runner.schedule_repeating_event(0,      1, self.counter)

=> runner.schedule_repeating_event(0.1,   1, self.agentsChoosingCounterpart)

runner.schedule_repeating_event(0.11,   1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.12,   1, self.agentsExchangingInTheirRanks)
runner.schedule_repeating_event(0.2,    1, self.sync)
runner.schedule_repeating_event(0.21,   1, self.ghostsExchangingInDifferentRanks)
runner.schedule_repeating_event(0.22,   1,\                                self.agentsHavingExchangedWithGhostsPreparingTheirOwnGhosts)
runner.schedule_repeating_event(0.23,   1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.24,   1, self.messengerGhostsReportingOccuredExchange
"""
schedule_stop(at)
Schedules the execution of this schedule to stop at the specified tick.

Parameters
at (float) ? the tick at which the schedule will stop.
"""
runner.schedule_stop(params['stop.at'])

runner.schedule_end_event(self.finish)
```

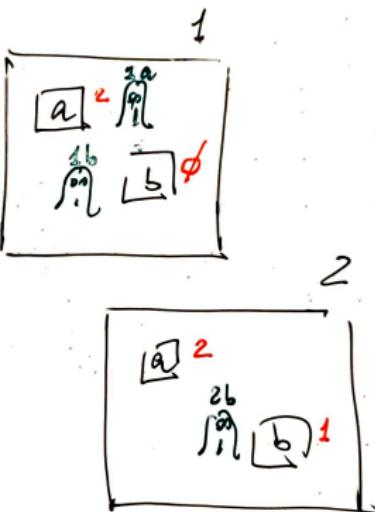
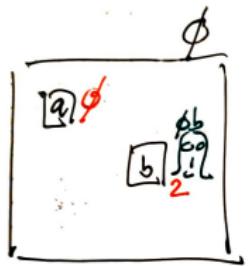
○

○○○

○○

○○

agentsChoosingCounterpart => choosingRankAndCreatingItsGhostIfAny()



agentsSendingTheirGhosts

```
runner.schedule_repeating_event(0,      1, self.counter)
runner.schedule_repeating_event(0.1,    1, self.agentsChoosingCounterpart)

=> runner.schedule_repeating_event(0.11,  1, self.agentsSendingTheirGhosts)

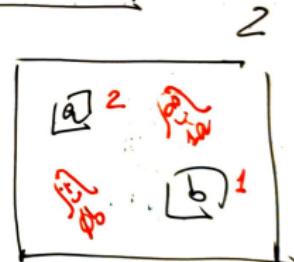
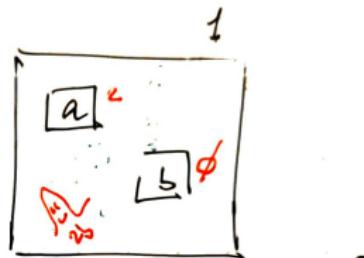
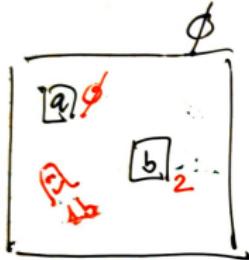
runner.schedule_repeating_event(0.12,  1, self.agentsExchangingInTheirRanks)
runner.schedule_repeating_event(0.2,   1, self.sync)
runner.schedule_repeating_event(0.21,  1, self.ghostsExchangingInDifferentRanks)
runner.schedule_repeating_event(0.22,  1,
                                self.agentsHavingExchangedWithGhostsPreparingTheirOwnGhosts)
runner.schedule_repeating_event(0.23,  1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.24,  1, self.messengerGhostsReportingOccuredExchange)
"""

schedule_stop(at)
Schedules the execution of this schedule to stop at the specified tick.

Parameters
at (float) ? the tick at which the schedule will stop.
"""
runner.schedule_stop(params['stop.at'])

runner.schedule_end_event(self.finish)
```

agents Sending Their Ghosts THROUGH broadcastGhostRequests, request_agents



0 req. 1b
 1 req. 2b
 2 req. 1a
 2 req. φb

agentsExchangingInTheirRanks

```
runner.schedule_repeating_event(0,      1, self.counter)
runner.schedule_repeating_event(0.1,    1, self.agentsChoosingCounterpart)
runner.schedule_repeating_event(0.11,   1, self.agentsSendingTheirGhosts)

=> runner.schedule_repeating_event(0.12, 1, self.agentsExchangingInTheirRanks)

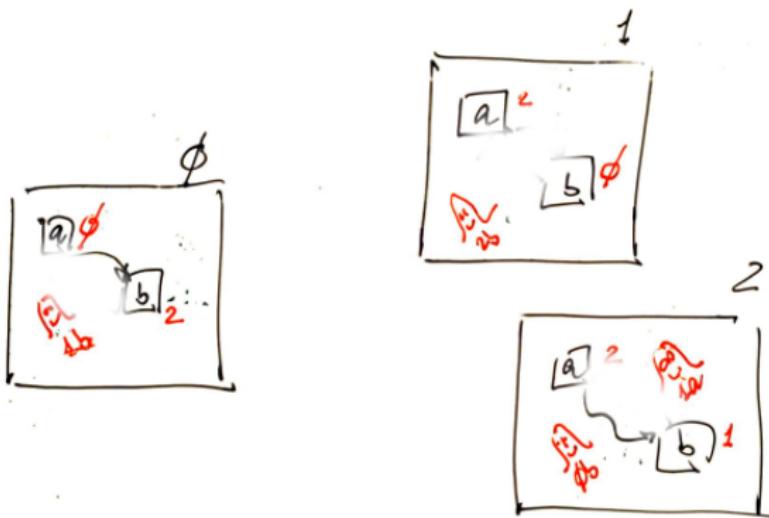
runner.schedule_repeating_event(0.2,   1, self.sync)
runner.schedule_repeating_event(0.21,  1, self.ghostsExchangingInDifferentRanks)
runner.schedule_repeating_event(0.22,  1,
                                self.agentsHavingExchangedWithGhostsPreparingTheirOwnGhosts)
runner.schedule_repeating_event(0.23,  1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.24,  1, self.messengerGhostsReportingOccuredExchange)
"""

schedule_stop(at)
Schedules the execution of this schedule to stop at the specified tick.

Parameters
at (float) ? the tick at which the schedule will stop.
"""
runner.schedule_stop(params['stop.at'])

runner.schedule_end_event(self.finish)
```

agentsExchangingInTheirRanks => operatingInItsRank



synchronize

```
runner.schedule_repeating_event(0,      1, self.counter)
runner.schedule_repeating_event(0.1,    1, self.agentsChoosingCounterpart)
runner.schedule_repeating_event(0.11,   1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.12,   1, self.agentsExchangingInTheirRanks)

=>     runner.schedule_repeating_event(0.2,   1, self.sync)

runner.schedule_repeating_event(0.21,  1, self.ghostsExchangingInDifferentRanks)
runner.schedule_repeating_event(0.22,  1,\                                self.agentsHavingExchangedWithGhostsPreparingTheirOwnGhosts)
runner.schedule_repeating_event(0.23,  1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.24,  1, self.messengerGhostsReportingOccuredExchanges)
"""

schedule_stop(at)
Schedules the execution of this schedule to stop at the specified tick.

Parameters
at (float) ? the tick at which the schedule will stop.
"""
runner.schedule_stop(params['stop.at'])

runner.schedule_end_event(self.finish)
```

ghostsExchangingInDifferentRanks

```
runner.schedule_repeating_event(0,      1, self.counter)
runner.schedule_repeating_event(0.1,    1, self.agentsChoosingCounterpart)
runner.schedule_repeating_event(0.11,   1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.12,   1, self.agentsExchangingInTheirRanks)
runner.schedule_repeating_event(0.2,    1, self.sync)

=> runner.schedule_repeating_event(0.21, 1, self.ghostsExchangingInDifferentRanks)

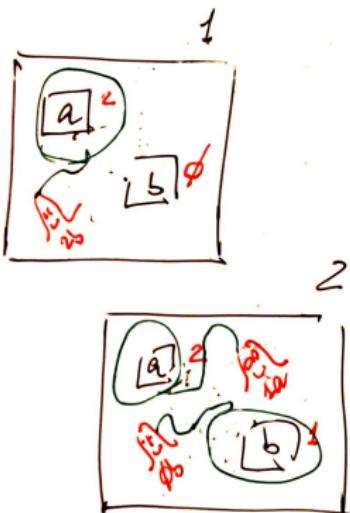
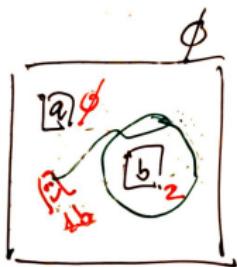
runner.schedule_repeating_event(0.22, 1,\                                self.agentsHavingExchangedWithGhostsPreparingTheirOwnGhosts)
runner.schedule_repeating_event(0.23, 1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.24, 1, self.messengerGhostsReportingOccuredExchange)
"""

schedule_stop(at)
Schedules the execution of this schedule to stop at the specified tick.

Parameters
at (float) ? the tick at which the schedule will stop.
"""
runner.schedule_stop(params['stop.at'])

runner.schedule_end_event(self.finish)
```

ghostsExchangingInDifferentRanks => actingAsGhost



agentsHavingExchangedWithGhostsPreparingTheirOwnGhosts

```

runner.schedule_repeating_event(0,      1, self.counter)
runner.schedule_repeating_event(0.1,    1, self.agentsChoosingCounterpart)
runner.schedule_repeating_event(0.11,   1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.12,   1, self.agentsExchangingInTheirRanks)
runner.schedule_repeating_event(0.2,    1, self.sync)
runner.schedule_repeating_event(0.21,   1, self.ghostsExchangingInDifferentRanks)

=>     runner.schedule_repeating_event(0.22, 1,\n                                         self.agentsHavingExchangedWithGhostsPreparingTheirOwnGhosts

runner.schedule_repeating_event(0.23, 1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.24, 1, self.messengerGhostsReportingOccuredExchanges)
"""

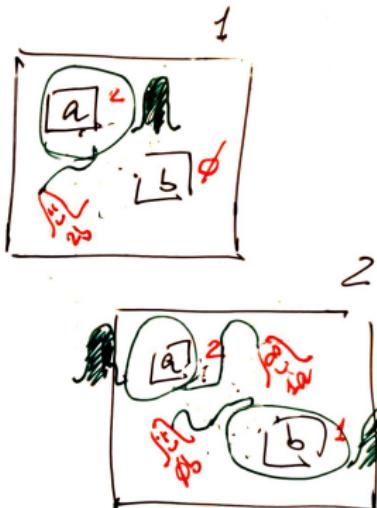
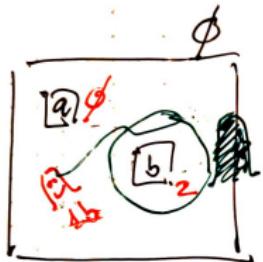
schedule_stop(at)
Schedules the execution of this schedule to stop at the specified tick.

Parameters
at (float) ? the tick at which the schedule will stop.
"""
runner.schedule_stop(params['stop.at'])

runner.schedule_end_event(self.finish)

```

agentsHavingExchangedWithGhostsPreparingTheirOwnGhosts =>
 sendingMyGhostToConcludeTheExchange



agentsSendingTheirGhosts

```
runner.schedule_repeating_event(0,      1, self.counter)
runner.schedule_repeating_event(0.1,    1, self.agentsChoosingCounterpart)
runner.schedule_repeating_event(0.11,   1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.12,   1, self.agentsExchangingInTheirRanks)
runner.schedule_repeating_event(0.2,    1, self.sync)
runner.schedule_repeating_event(0.21,   1, self.ghostsExchangingInDifferentRanks)
runner.schedule_repeating_event(0.22,   1,\                                self.agentsHavingExchangedWithGhostsPreparingTheirOwnGhosts)

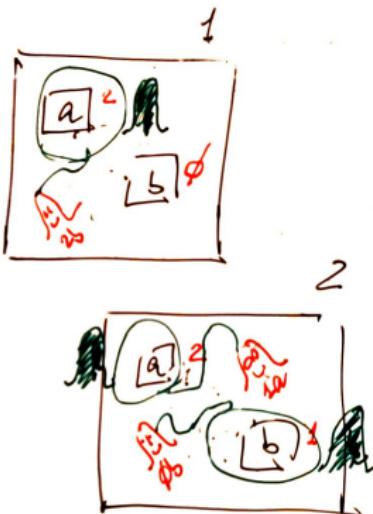
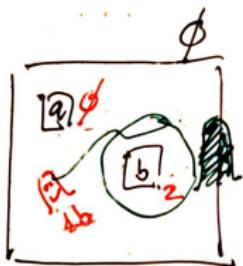
=>     runner.schedule_repeating_event(0.23, 1, self.agentsSendingTheirGhosts)

runner.schedule_repeating_event(0.24, 1, self.messengerGhostsReportingOccuredExchange)
"""
schedule_stop(at)
Schedules the execution of this schedule to stop at the specified tick.

Parameters
at (float) ? the tick at which the schedule will stop.
"""
runner.schedule_stop(params['stop.at'])

runner.schedule_end_event(self.finish)
```

agents Sending Their Ghosts THROUGH broadcastGhostRequests, request_agents



φ req. 2b
 1 req. φ b
 1 req. 2a
 2 req. 1a

messengerGhostsReportingOccuredExchanges

```
runner.schedule_repeating_event(0,      1, self.counter)
runner.schedule_repeating_event(0.1,    1, self.agentsChoosingCounterpart)
runner.schedule_repeating_event(0.11,   1, self.agentsSendingTheirGhosts)
runner.schedule_repeating_event(0.12,   1, self.agentsExchangingInTheirRanks)
runner.schedule_repeating_event(0.2,    1, self.sync)
runner.schedule_repeating_event(0.21,   1, self.ghostsExchangingInDifferentRanks)
runner.schedule_repeating_event(0.22,   1,\                                self.agentsHavingExchangedWithGhostsPreparingTheirOwnGhosts)
runner.schedule_repeating_event(0.23,   1, self.agentsSendingTheirGhosts)

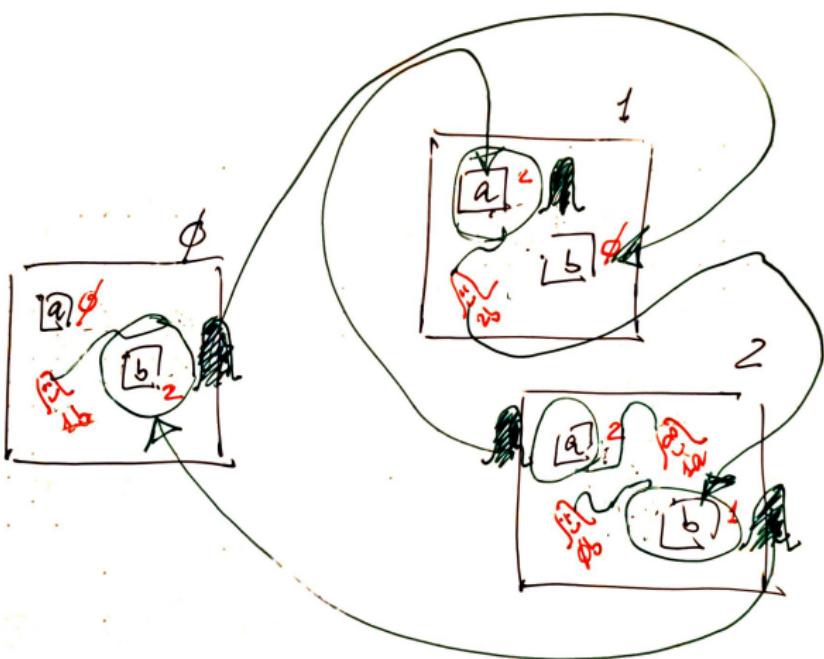
=> runner.schedule_repeating_event(0.24, 1, self.messengerGhostsReportingOccuredExchanges)

"""
schedule_stop(at)
Schedules the execution of this schedule to stop at the specified tick.

Parameters
at (float) ? the tick at which the schedule will stop.
"""
runner.schedule_stop(params['stop.at'])

runner.schedule_end_event(self.finish)
```

messengerGhostsReportingOccuredExchanges => actingAsReportingGhost



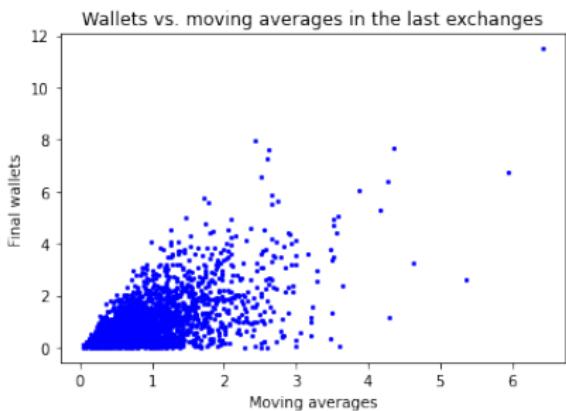
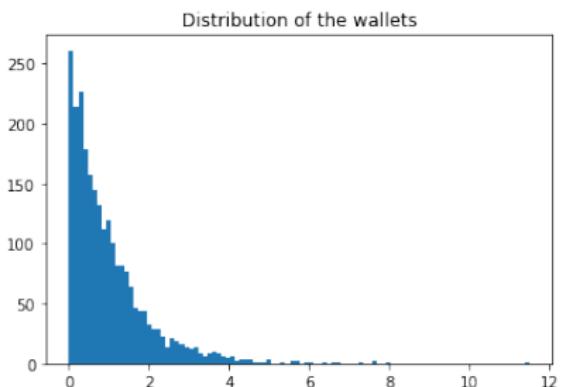
Timing

Agent #: 2400 Cycles: 200

- 1 rank: s. 480.2 transaction # 480000
- 3 ranks without interaction
 - rank0: s. 66.14 transaction # 160000
 - rank1: s. 66.12 transaction # 160000
 - rank2: s. 66.13 transaction # 160000
- 3 ranks with interaction
 - rank0: s. 52.29 transaction # 160366
 - rank1: s. 52.28 transaction # 159853
 - rank2: s. 52.29 transaction # 159781
- using a simple program (no classes, no agents) as par2.1Chakraborti.ipynb in the git
480000 transactions in s. 2.78



Results



Final remarks

- Repast4Py perfectly working also on a unique rank.
 - Probably, no more using SLAPP, <https://terna.github.io/SLAPP/>.



A perspective from the past to the future

○

A tool for huge ABMs

○○○

The scheme of an ABM

A sequence of 15 small circles arranged in two rows. The top row has 4 circles, and the bottom row has 11 circles.

Timing and Results

00

Final remarks

1

Thanks

Slides at <https://terna.to.it/PrioriTerna20221117.pdf>



Pietro Terna home: <https://terna.to.it>

