# Solving simulation optimization problems on grid computing systems

YOUSEF RAJAEI TABRIZI

Universidad de La Laguna

January 25, 2016

Escuela Superior de Ingeniería y Tecnología
Universidad de La Laguna

# Diagramas of expriments

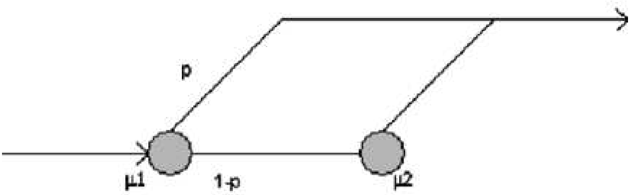



Fig. 1. Two phases, Coxian distribution.

Figure : frequencia de la aparecion de los barcos

# Índice

# Índice

# Índice

# Índice

# Motivación

## Definición

*The optimal assignment of berth slots and cranes to shipping services is the central logistics problem at modern marine container terminals and should be formulated to specifically account for its stochastic nature. We use a computational grid to solve a major seaport logistic problem by a simulation optimization approach centred around a queuing network model of the logistic process of interest. We emphasize the power of grid computing for the simulation optimization studies and we design and implement an algorithm for distributing the computational load to parallel processors. Performance of the algorithm is demonstrated numerically using real-sized problem instances. 2006 Elsevier B.V. All rights reserved.*

# Objetivos

## Ejemplo

- Using technics of simulation for optimization

# Objetivos

## Ejemplo

- Using technics of simulation for optimization
- Using techincs of parallelization

# Fundamentos Teóricos

In the sequel we illustrate the proposed queuing model in which we eliminate some details from the model description of Legato and Mazza [12] because they are not significant for the purpose of this paper. We assume that the occurrence of a delay-time spent at roadstead by an incoming vessel, due to lack of berth slots, is represented as a special case of the phase type Cox's distribution. This assumption can be accepted for the following considerations. Usually, an incoming vessel receives almost immediately the required slots for berthing. In this case, with a very high probability (p), the elapsing interval from "arrival to port" and "berthing time completion" results in a very short time (l1, on average), due to the relatively fast operations for vessel positioning along the berth. With a very low probability (1 p), an arrived vessel is delayed at roadstead due to unavailability of the required berth slots: once this happens, then a very long interval (l1 + l2, on average) elapses from "arrival to port" and "berthing time" (Fig. 1).

# Procedimiento experimental

In this paper, we focus on the case where incoming vessels (customers) belonging to one out of a set of maritime shipping "services" (customer classes) can be berthed into shared segments of berth. Each berth segment (out of "n") corresponds to a fixed number of vessel positions for berthing and is equipped with one or several cranes; in Fig. 3 it is represented as a multi-server queue with a finite number of waiting places. The delay-time at roadstead, for berth assignment operations, is represented by a single server queue with unlimited waiting places. In the berth planning problem, we should answer the following questions: • how many segments do we organise for active shipping services? • how many cranes – out of the total, fixed, number of available ones – do we allocate for each of the organised segments? • to which segment do we forward incoming vessels, provided that we may base this decision on some suitable attributes shared by any given subset of the active services? Answers to these questions are provided by the simulation optimization approach.

# Diagramas of expriments

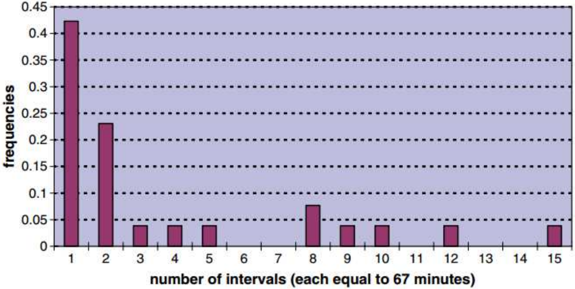



Fig. 2. Empirical probability mass function for waiting times at roadstead.

Figure : frequencia de la aparecion de los barcos

# Diagramas of expriments

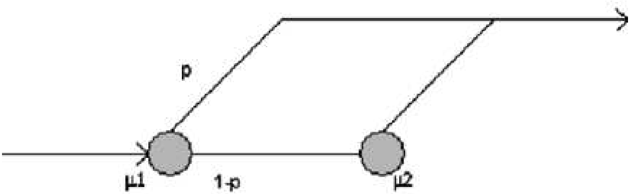



Fig. 1. Two phases, Coxian distribution.

Figure : frequencia de la aparecion de los barcos

# A combined procedure for simulation optimization problems

## Ejemplo

- *Algorithm description*

# A combined procedure for simulation optimization problems

## Ejemplo

- *Algorithm description*
- *Grid enabled versions for the SARP algorithm*

# A combined procedure for simulation optimization problems

## Ejemplo

- *Algorithm description*
- *Grid enabled versions for the SARP algorithm*
- *Algorithm 2. SARP*

## Ejemplo

1. *With the aim of developing grid versions of the SARP algorithm described above (in the sequel G-SARP), we have adopted a master/worker approach [16,4,19]. Let p be the number of workers. At iteration k, the master creates p + 1 perturbations of a same configuration of the system (i.e. p + 1 neighbours of the current solution), keeps for itself the first generated configuration to be estimated and sends the others to the workers (one configuration for each worker). Each processor, including the master, must decide if its own configuration can be accepted or not. If a processor accepts its own solution (according to the acceptance criterion of SARP), then sends it to the master. If the master has new solutions to examine, it selects the next configuration, otherwise it keeps the old one. There are different rules for selection; for example, the master can choose the configuration with the best estimated performance (best strategy) or it can make a random choice (random strategy). We have implemented both, and we present in*

## Ejemplo

1. *With the aim of developing grid versions of the SARP algorithm described above (in the sequel G-SARP), we have adopted a master/worker approach [16,4,19]. Let p be the number of workers. At iteration k, the master creates $p+1$ perturbations of a same configuration of the system (i.e. $p+1$ neighbours of the current solution), keeps for itself the first generated configuration to be estimated and sends the others to the workers (one configuration for each worker). Each processor, including the master, must decide if its own configuration can be accepted or not. If a processor accepts its own solution (according to the acceptance criterion of SARP), then sends it to the master. If the master has new solutions to examine, it selects the next configuration, otherwise it keeps the old one. There are different rules for selection; for example, the master can choose the configuration with the best estimated performance (best strategy) or it can make a random choice (random strategy). We have implemented both, and we present in*

# Computational experiments

| **Tiempo ($\pm$ 0.001 s)** | **Velocidad ($\pm$ 0.1 m/s)** |
|---|---|
| 1.234 | 67.8 |
| 2.345 | 78.9 |
| 3.456 | 89.1 |
| 4.567 | 91.2 |

Table : Resultados experimentales de tiempo (s) y velocidad (m/s)

# Diagramas of expriments

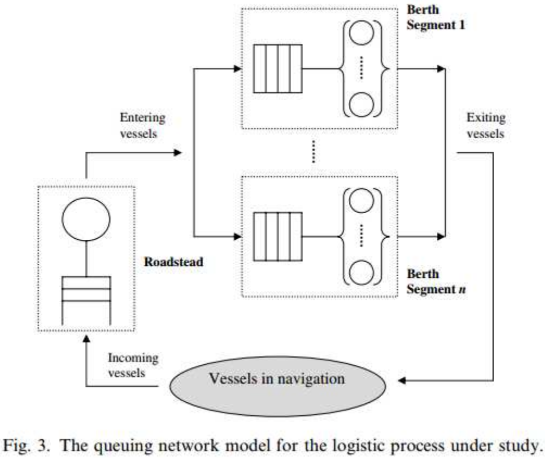
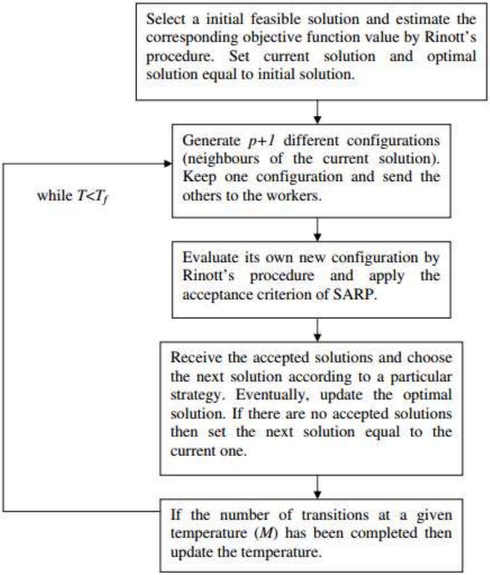



Figure : imagen de la muelle

# Diagramas of expriments



# Conclusiones

## Ejemplo

1. *We have designed and implemented on a grid platform a simulation optimization algorithm that requires a limited amount of communication and a minimal synchronization among computing resources. It could be used for a cost-effective solution of logistics decision problems at seaport container terminals. Numerical evidence shows that as the number of workers increases, the quality of the final solution improves. A crucial role is played by the optimal compromise between searching the entire feasible region (exploration) and locally searching promising sub-regions (exploitation).*

# Conclusiones

## Ejemplo

1. *We have designed and implemented on a grid platform a simulation optimization algorithm that requires a limited amount of communication and a minimal synchronization among computing resources. It could be used for a cost-effective solution of logistics decision problems at seaport container terminals. Numerical evidence shows that as the number of workers increases, the quality of the final solution improves. A crucial role is played by the optimal compromise between searching the entire feasible region (exploration) and locally searching promising sub-regions (exploitation).*
2. *Nasim3*

# Bibliografía


- CTAN. *http : //www.ctan.org/*
- Tantau, Till. *User's Guide to the* BEAMER *Class, Version 3.06, 2005* *http : //ctang.tug.org/tex − archive/macros/latex/contrib/beamer*