Accepted Manuscript

Title: A Simulation Based Decision Support System For Logistics Management

Author: M.P. Fanti G. Iacobellis Walter Ukovich V. Boschian G. Georgoulas C. Stylios

PII: S1877-7503(14)00123-9
DOI: http://dx.doi.org/doi:10.1016/j.jocs.2014.10.003
Reference: JOCS 304

To appear in:

Received date: 21-2-2014
Revised date: 26-9-2014
Accepted date: 14-10-2014

Please cite this article as: M.P. Fanti, G. Iacobellis, W. Ukovich, V. Boschian, G. Georgoulas, C. Stylios, A Simulation Based Decision Support System For Logistics Management, *Journal of Computational Science* (2014), http://dx.doi.org/10.1016/j.jocs.2014.10.003

Maria Pia Fanti received the Laurea degree in electronic engineering from the University of Pisa, Italy in 1983 and was a visiting researcher at the Rensselaer Polytechnic Institute of Troy, New York, in 1999. Since 1983 she has been with the Polytechnic of Bari (Italy), where she was Assistant Professor from 1990 till 1998 and an associate professor from 1998 till 2012. Now she is full professor of System and Control Engineering in the Department of Electrical and Information Engineering of the same Polytechnic.
Her research interests include discrete event systems, Petri net, sensor networks, management and modeling of automated manufacturing systems, automatic guided vehicle systems, railway and traffic networks, supply chains, and healthcare systems. She has published around 210+ papers and two textbooks on these topics. She was General Chair of the 2nd IFAC Workshop on Dependable Control of Discrete Systems, the 2010 IEEE Workshop on Health Care Management, and the 2011 IEEE Conference on Automation Science and Engineering. She was member of the International Program Committees of around 30 international conferences.
Prof. Fanti is Associate Editor of the IEEE TRANS. ON SYSTEMS, MAN, AND CYBERNETICS PART A, and Editor of the IEEE TRANS. ON AUTOMATION SCIENCE AND ENGINEERING. She is vice-chair of the IEEE Italy Section, Co-Chair of the IEEE SMC Technical committee on Discrete Event Systems, Chair of the Central & Southern Italy SMC Chapter, Chair of the IEEE RAS Technical committee on Automation in Logistics.

Walter Ukovich is full professor of Operations Research at the University of Trieste, Italy. From 1974 to 1978, he was in charge of the course of Electrical Engineering, and since 1979 he holds the course of Operations Research at the Faculty of Engineering.
He has been a member of the Evaluation Boards of the National Research Council (CNR) of Italy, of the Universities of Trieste, Italy, and of Macerata, Italy, of the Polytechnic of Turin, Italy, and of the University Institute of Architecture of Venice, Italy. His main research interests are in Optimization, Logistics, Transportation, Production Planning and Control and Health Management Systems.
He is the author of over one hundred scientific papers, which appeared in several journals, including Operations Research, Management Science, SIAM Journal on Algebraic and Discrete Methods, SIAM Journal on Optimization, IEEE Transactions on Robotics and Automation, IEEE Transactions on Automatic Control, Automatica, Journal of Optimization Theory and Applications, Naval Research Logistics, Networks, Transportation Research, International Journal of Production Research, International Journal of Production Economics, Computers and Operations Research and European Journal of Operational Research.

Valentina Boschian graduated in management and logistics engineering in 2008, she obtained her Ph.D. at the University of Trieste in information technology engineering in March 2012 focused on the study of multi-actor systems and of the impact of advanced ICT on intermodal logistic system. She is also working since February 2008 on logistic topics and working in different European projects founded in the 7th EU Framework Program in the ICT area. Her experience and knowledge of such topics related to logistics and intermodality allow her to actively work on EU-founded and research project proposals and publications of scientific paper in proceedings of international conferences and in international journals.

Giorgio Iacobellis received the Laurea degree in electronic engineering and the Ph.D. degree in computer science engineering from the Polytechnic of Bari (Italy), in 2004 and 2009, respectively. From February 2005 to February 2006, he was a Research Fellow at Politecnico di Bari, from September 2009 to 2011 he has been a Research Fellow at the University of Trieste, from 2011 to 2012 he has been Experienced Researcher Fellowship (ER2) on Development of Decision Support System DSS at Technological Educational Institute of Epirus ad Arta (Greece). Since the 2012 he is Research fellow at Politecnico di Bari. His research interests include Decision Support System, modeling, simulation, and control of discrete-event systems, Petri nets, supply chains and urban traffic networks, distribution and internal logistics, management of hazardous materials, management of drug distribution systems, and healthcare systems.

George Georgoulas holds a Diploma in Electrical and Computer Engineering (1999, University of Patras), and a Ph.D. in Data Processing (2006, University of Patras). From 2006 to 2008 he worked as a postdoctoral fellow at the Georgia Institute of Technology (Georgia Tech). From 2008 to 2009 he worked as Scientific Associate (non-permanent Assistant Professor) at the Department of Applications of Information Technology in Management and Economics, Technological Educational Institute (TEI) of Ionian Islands and at the Department of Informatics Engineering at the TEI of Epirus. Lately he was an Experienced Researcher Fellow (ER2) under the SAIL Marie Curie Project for the Development of a Decision Support System. Currently he is working as a research fellow at the Laboratory of Knowledge & Intelligent Computing at the Department of Informatics Engineering at the TEI of Epirus.
His expertise and research interests involve areas such as Machine Learning, Computational Intelligence, Data Mining, Evolutionary based Global Optimization and Decision Support Systems for complex environments.

Dr. Chrysostomos Stylios is an Associate Professor at Dept. of Computer Engineering, Technological Educational Institute of Epirus, Greece. He is also research collaborator at Computer Technology Institute & Press "Diophantus", Patras, Greece. He was a visitor assistant professor at Computer Science Dep., University of Ioannina. He received his Ph.D from the Department of Electrical & Computer Engineering, University of Patras (1999) and the diploma in Electrical & Computer Engineering from the Aristotle University of Thessaloniki (1992). He has published over 135 journal and conference papers and book chapters. His main scientific interests include: Fuzzy Cognitive Maps, Soft Computing &, Computational Intelligence Techniques, Signal Processing methods and Decision Support Systems. Prof. Stylios is member of IEEE and member of the TC 8.2 and TC 5.4 of IFAC.

Maria Pia Fanti

Walter ukovich

Giorgio Iacobellis

 Valentina Boschian

 George Georgoulas

 Chrysostomos Stylios

A SIMULATION BASED DECISION SUPPORT SYSTEM FOR LOGISTICS MANAGEMENT

M.P. Fanti (a), G. Iacobellis (a), Ukovich (b), V. Boschian (b), G. Georgoulas(c), C. Stylios(c)

[(a)] Polytechnic of Bari, Via Orabona 4, 70125 Bari, Italy
[(b)] University of Trieste, Via Valerio 10, 34127 Trieste, Italy
[(c)] Technological Educational Institute of Epirus, GR-47100 Arta, Greece

[(a)]{mariapia.fanti, giorgio.iacobellis}@poliba.it, [(b)]{valentina.boschian, walter.ukovich}@di3.units.it
[(c)]georgoul@kic.teiep.com, stylios@teiep.gr

Abstract - **This paper deals with designing and developing a Decision Support System (DSS) that will be able to manage the flow of goods and the business transactions between a port and a dry port. An integrated DSS architecture is proposed and specified and the main components are designed on the basis of simulation and optimization modules. In order to show the use and implementation of the DSS, this work tests and analyzes the case of the area of the Trieste port and manages the export flows of freights between a dry port and a seaport. An integrated approach is designed mainly at tactical and operational decision level exploiting simulation and optimization approaches and especially metaheuristic approaches.**

Keywords: Decision Support Systems, Discrete Event Simulation, Optimization, Metaheuristic algorithms, Logistics

1. Introduction

The increasing complexity of modern business environment and the vast volume of available data, that could be taken into account, make the use of advanced modeling and computerized methods a necessity. Among the others, logistics systems such as ports, dry ports and inland terminals have been widely studied during the recent years (see for instance the review in [12]). In particular, Roso et al. [17, 18] point out that the dry ports are not only terminals linked one to another but also terminals where some typical services of the seaports are moved, in order to provide more available space and to require less service time at the port area. In the related literature, there are several papers that analyse intermodal terminals [21, 7] and in particular container terminals [19]. Indeed, the management of seaport and dry port terminals has become a popular topic of academic research worldwide (see the surveys [19, 20, 22] for a comprehensive review).

However, the current concept of dry ports directly connected with the seaport opens new series of problems to be faced, since the logistics operations between the two terminals must be coordinated and synchronized [1, 9]. Moreover, the increasing availability of Information and Communication Technologies (ICTs), for the interaction among the Decision Makers (DMs) and the acquisition of information, requires the development of models and leads to the definition of novel decision making strategies with respect to the related literature.

Analogously to what is done in other application areas (e.g., production processes), it is possible to identify different hierarchical/functional levels also for logistics systems which different decisional problems are associated to [1]: a) the tactical level, related (on a middle term) to the management of logistic flows connected to the information flow and to the transportation network; and b) the operational level, including real-time decisional processes and decisions concerning the resource assignment, the vehicle routing definition, and so on. In particular, assuming a real-time availability of the information regarding the conditions of the network (like unexpected requests of transportation, variations in the availability of the transportation system, road conditions and traffic flows), operational decisions should be taken in a dynamic context.

This paper develops a Decision Support System (DSS) to be used by DMs that have to take operational as well as tactical decisions in logistics networks composed by a port and a dry port. Starting from the approach presented in [2, 7], in this paper we specify in detail the main components of the DSS: the data component, the model component, the decision component and the interface component. In particular, the structure and the activities of the DSS are described by the Unified Modeling Language (UML) diagrams. Moreover, the proposed management and planning approach is based on the specification of two main modules that are the core of the DSS: a simulation model and an optimization module. If the performance has to be improved, then the DSS determines the decision variables that should be chosen in order to optimize a specific objective function. Hence, the simulation module foresees the evolution of the system and provides to the optimization module the estimated results. On the basis of such results the optimization module can trigger other simulations with new values for the decision variables in order to optimize the chosen performances.

On the other hand, simulation is considered the standard approach for performance evaluation of logistics systems [16, 20] due to the inherent capability of dealing with the complexity and the randomness of the logistics operations. Moreover, the related literature points out that the discrete-event simulation is the most preferred modelling technique for the components of the logistics intermodal system [6, 8, 13]. Hence, also in this case, the considered logistics systems can be successfully modelled as Discrete Event Systems, whose dynamics depends on the interaction of discrete events, such as demands, departures and arrivals of means of transportation at terminals and acquisitions and releases of resources by vehicles. Here, the simulation is used as a kind of observer that allows determining the performances of the systems and evaluating how a selected set of parameters can improve the considered performance indices. To this purpose we use the Particle Swarm Optimization (PSO) combined with the Optimal Computing Budget Allocation (OCBA) schemes in order to optimally allocate the number of simulation trials and replications that achieves the "best" system performances.

In order to show the applicability and the effectiveness of the DSS in real system management, we present a case study that is constituted by the Trieste port located at the north of Italy, the traffic of trucks directed to Turkey through a roll-on/roll-off traffic (Ro-Ro) service and the dry port of Fernetti. In particular, we apply a prototype of the described DSS to take some tactical and operational decisions involving the movements of trucks between the port and the dry port area.

The paper is organized as follows. Section 2 describes the structure of the DSS and Section 3 specifies the DSS architecture and components. Section 4 describes the considerd case study and Section 5 shows how the DSS can be utilized at the tactical and operational level management. Finally, Section 6 summarizes the conclusions.

2. The Decision Support System Structure

In this section, we describe the main components of the proposed DSS. Typically, DSSs are categorized on the basis of different characteristics of the systems, e.g. whether they are for personal or decision making group oriented [15]. Based on the type of the application, DSSs can also be divided in desktop and web based applications. Despite the different categories of DSS, all of them share common characteristics; i.e., a typical DSS should include four main components: the data component, the model component, the decision component, and the interface component.

In this paper, we use the UML class diagram of Fig. 1 to describe the proposed DSS structure. More precisely, each class is represented by a rectangular box that is divided into different compartments. The first compartment holds the class name, the second holds attributes and the last holds operations. Attributes are qualities and named property values that describe the characteristics of a class. In addition, operations are features that specify the class behavior. Moreover, classes can exhibit relationships that are represented by different graphic connections: association (solid line), aggregation (solid line with a clear diamond at one end), composition (solid line with a filled diamond at one end), inheritance or generalization (solid line with a clear triangle at one end), realization (dashed line with a clear triangle at one end) and dependency (dashed line with an arrow at one end). Each component can be modeled as a different class illustrating the different types of objects that the system can have and their relationships.

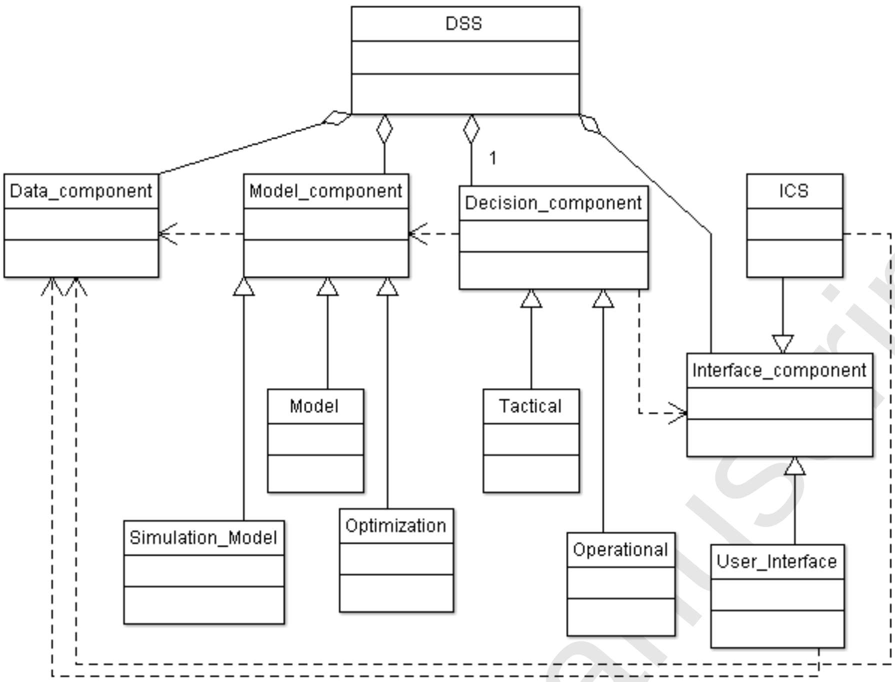

Figure 1. The UML class diagram of the DSS.

We briefly specify the DSS components shown in Fig. 1 as follows.

1. ***Data component***. The data component usually consists of a Database Management System (DBMS). The data used can be internal, if they come from organization's internal procedures and sources such as products and services prices, recourse and budget allocation data, payroll cost, cost-per-product etc. External data can be related with competition market share, government regulations and may come from various resources such as market research firm, government agencies, the web etc.. In some cases, the DSS can have its own database or it may use other organizational databases either by connecting directly with them or by using data available from reports.
2. ***Model component***. This component mainly includes a simulation model, a mathematical model, and a set of optimization algorithms suitable to analyze effects of choices on the system performances. The models describe the operations at different management levels and the type of functions varies with the operation that they support.
3. ***Interface component***. This module is the part of the DSS that is responsible for the communication and interaction of the system with the DMs. Such a component is very important because regardless of the quality and quantity of the available data; the accuracy of the model is based on this interface. Indeed, this component includes an Information Communication System (ICS) that is able to interact with the real system and maintains the consistency between the stored data and the real system.

4. ***Decision component***. This component consists of two-second level classes: the operational decision class and the tactical decision class. Moreover, such classes include the performance indices that have to be considered in order to take the decisions. In addition, in relation with the performance indices and the object of the decisions, the DSS has to collect the decision rules and the optimization procedures that are used by the model and the simulation component.

3. The DSS Architecture

This section describes in detail the DSS architecture and components. Figure 2 shows the DSS architecture by enlightening the basic models and the connections among them. Moreover, in the following we explain the objectives and the roles of the DSS components.

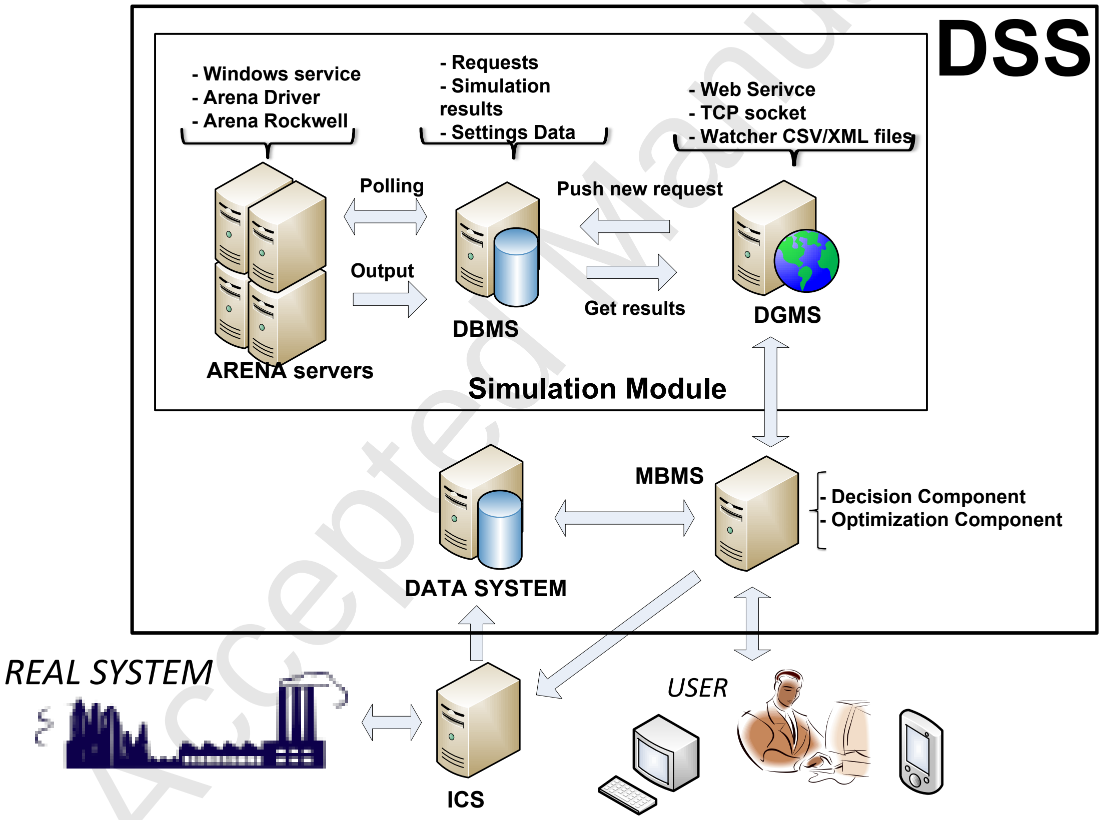

Figure 2. The DSS architecture.

3.1. *The Data Component*

The data component is specified by distinguishing three different kinds of data. The first data are managed by the DBMS that stores the internal data used by the decision and the simulation components. More precisely, the DBMS stores the requests of a new simulation with its input data, the data related to internal variables of the simulation model, and the outputs produced by simulations. Furthermore, the DBMS contains the queue of the requests to be sent to the simulation servers, the state of each request and the results of the simulation.

Moreover, it stores the description of all the simulation models that are available for the simulation runs.

The second and third types of data are stored in the Data System: the internal data and the external data. The internal data represent all data necessary to describe the internal procedures, e.g. the time required for each activity, the number of available resources, the capacity of parking areas and safety levels, etc.. On the other hand, the external data are information coming in real time from the system: the current number of vehicles, the information about the conditions of the roads, the accidents, the road maintenance works, the weather conditions, etc.

3.2. The Model Component: the simulation module

The model component is the core of the DSS: it consists of the simulation model and the optimization component. The simulation model mimes the system, applies the optimization strategies proposed by the optimization module and provides the performance measures. In the proposed solution, the simulation model is implemented by the simulation servers that consist of the following main elements [10]:

- simulation software that is a server where the simulation models are implemented and executed;
- windows service, that is automatically started. Moreover, it monitors the DBMS, executes the simulation and loads the results on the data base;
- simulation driver, that connects the system to the simulation software: it is in charge of starting and stopping simulations.

By the proposed architectural solution, a set of simulation servers can be provided and each server is independent from the others. Then, the number of operative servers can dynamically change on the basis of the computational effort that is required in real time. Moreover, such an approach allows a parallel execution of the simulation requests by reducing the total execution time.

3.3. The Model Component: the optimization module

The second basic module of the model component is the optimization module (see Fig. 2) that combines a variant of PSO [11] with an OCBA scheme [4]. In particular, the main concept of the PSO includes a population, called a swarm, of potential solutions of the problem at hand, called the particles, probing the search space. The particles iteratively move in the search space with an adaptable velocity, retaining in a memory the best positions they have ever visited, i.e., the positions with the lowest function values (considering only minimization problems). The exploration capability of PSO is promoted by information exchange among particles. In the global PSO variant, the neighborhood of each particle is the whole swarm and the overall best position is the main information provider for all particles. On the other hand, in the local PSO variant, the neighborhoods are strictly smaller, usually consisting of a few particles. In such cases, each particle may have its own leader that influences its velocity update. The use of the global best simplifies the comparisons that are involved in simulation optimization procedures, as it will be described in the following paragraphs. Therefore the two fundamental equations governing global best PSO version are presented for the noise free case.

We consider the following generic minimization problem:

$$\min_{x \in S \subset R^d} J(x) \tag{1}$$

where given k particles, the vector $x_i \in S$ for $i=1,2,\ldots, k$ of dimension d represents the state of the i-th particle and S is the set of all possible states. The i-th particle has a velocity (position shift) v_i and keeps in his personal memory the best position, $p_i \in S$, that it has ever visited. Moreover, the swarm keeps in its collective memory the best position ever visited by a member of the swarm p_g. The constriction coefficient version of PSO is described by the following set of equations [14, 5], which govern the update of the positions and the velocities of the particles/candidate solution:

$$v_{ij}^{(t+1)} = \chi\left[v_{ij}^{(t)} + \varphi_1\left(p_{ij}^{(t)} - x_{ij}^{(t)}\right) + \varphi_2\left(p_{gij}^{(t)} - x_{ij}^{(t)}\right)\right] \tag{2}$$

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)} \tag{3}$$

where $i=1,2,\ldots, k$ and $j=1,2,\ldots, d$. The parameter χ is the constriction coefficient and it is used as a means to control the magnitude of the velocities. The other two parameters are defined as $\varphi_1 = c_1 r_1$ and $\varphi_2 = c_2 r_2$, where c_1 and c_2 are positive constants, also called the cognitive and the social parameter, respectively, and r_1, r_2, are random variables uniformly distributed in [0,1], different for each i, j and t.

In real life problems we usually accept good enough solutions instead of the globally optimal solution. Therefore, in case of "noisy" functions, "following" the "best" particle becomes difficult since the actual value of a particle is obscured by noise and repeated function evaluations are required in order to accurately estimate the true value. Especially in situations where a function evaluation is a costly process, a compromise should be reached between the need for an accurate estimation of the true value and the need of having as small as possible number of function evaluations. In PSO, there are two main procedures that need to be accomplished at each iteration. One concerns the determination of the personal best position for each particle, and the other deals with the selection of the neighborhood's best particle. In simulation optimization, the challenge lies at the main task of identifying the actual best particle among k candidates, with respect to the smallest mean objective function value, while minimizing the total number of replications needed for a precise and safe selection. To this aim, OCBA can be a useful tool.

Indeed, the OCBA is a procedure to optimally allocate a predefined number of trials/replications in order to maximize the probability of selecting the best system/design: allocate replications not only based on the variance of the different designs but also taking into account the respective means.

According to OCBA, if we have a total budget of T replications, $T = \sum_{i=1}^{k} N_i$ then we try to (asymptotically) maximize the probability of Correct Selection P{CS} (the probability of actually selecting the best b among k designs) [3] (for the case of a minimization problems):

$$P\{CS\} = P\left\{ \operatorname*{I}_{i=1, i\neq b}^{k} \left(J_b < J_i\right)\right\}. \tag{4}$$

Within the OCBA framework, instead of estimating the P{CS} one resorts to a much easier to compute lower bound which is called the Approximate Probability of Correct Selection (APCS) that can be in two forms [3]: APCS-B and APCS-B.

APCS-B is in a summation form derived using the Bonferroni inequality:

$$APCS - B \equiv 1 - \sum_{i=1, i\neq b}^{k} P\left\{J_b > J_i\right\} \leq P\{CS\}. \tag{5}$$

APCS-P is expressed as a product form:

$$APCS - B \equiv \prod_{i=1, i\neq b}^{k} P\left\{J_b < J_i\right\} \leq P\{CS\}. \tag{6}$$

In [4] it is proved that in order to asymptotically maximize APCS (as the simulation budget approaches infinity) and as a result P{CS} or to minimize the total number of replications for a given confidence interval, the following relationship between two non-best designs (N_i, N_j) should hold:

$$\frac{N_i}{N_j} = \left(\frac{\sigma_i / \delta_{b,i}}{\sigma_j / \delta_{b,j}}\right)^2, \text{ for all } i \neq j \neq b, \tag{7}$$

with the number of simulation replications for the best design given as

$$N_b = \sigma_b \sqrt{\sum_{i=1, i\neq b}^{k} \frac{N_i^2}{\sigma_i^2}}, \tag{8}$$

where, μ_i, σ_i, are the mean and standard deviation of the i-th design, $\delta_{b,i} = \mu_i - \mu_b$, and b is the best design. More specifically, the noisier the simulation output (larger variance), the more replications are allocated while more replications are also given to the design that its mean is closer to that of the best design. The integration of OCBA with PSO is performed in order to satisfy the aforementioned procedures. A schematic description of the followed procedure for the minimization problem is given in Fig. 3.

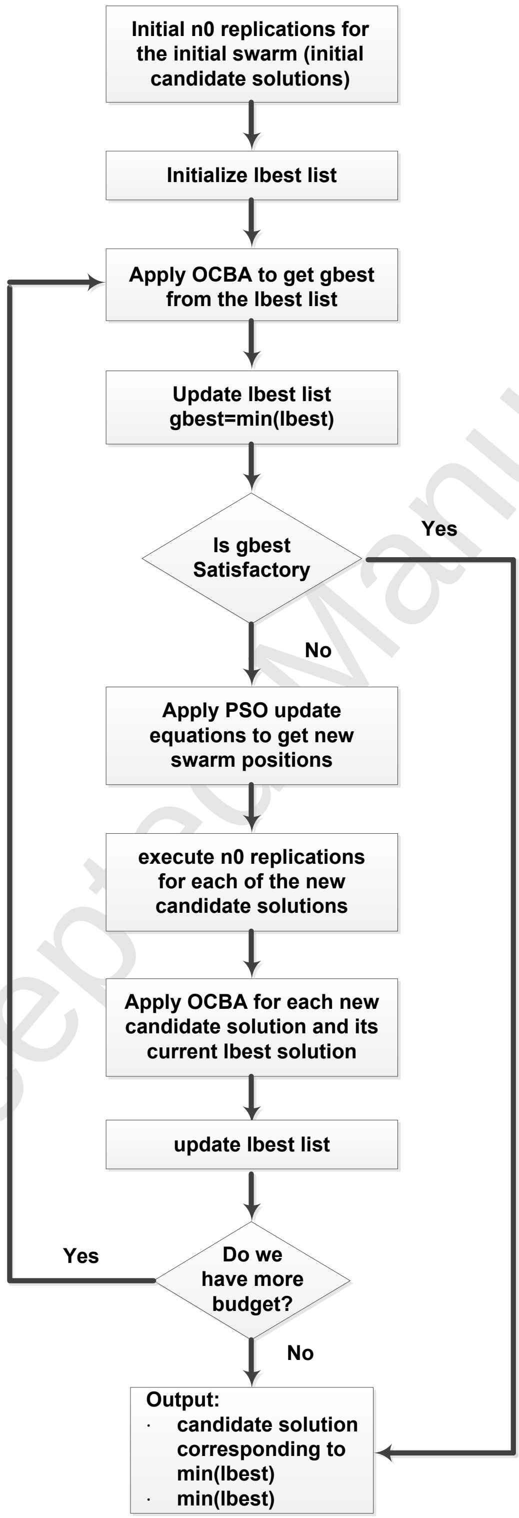

Figure 3. The outline of the PSO+OCBA procedure (for a minimization problem).

At the beginning of the algorithm the particles are initialized randomly, taken values within the feasible range of solutions (note: this part can be slightly modified to include candidate solution dictated by expert knowledge), and for each particle a number of n_0 replications is executed. These initial candidate solutions also constitute the initial list of the *lbest* candidates (holding the p_i for each candidate) since no other solutions exist at this stage. From that *lbest* list the overall best solution *gbest* (p_g) is retrieved after using the OCBA procedure. At this point, we must note that each time the OCBA procedure is invoked it assigns extra replications, one at a time (Δ=1), till either the available budget is exhausted or the predefined APCS is reached.

After the initialization phase the main PSO-OCBA loop is executed until the available computational budget is consumed. More specifically once a new position is x_i is produced by the PSO algorithm, n_0 replications are assigned to it. After these initial n_0 replications have been executed of each one of the new positions of the swarm, the OCBA procedure is used to update the *lbest* list: each new candidate x_i is compared with its personal best position p_i using the OCBA procedure. At the end of this pairwise comparison cycle, the *lbest* list is updated. The updated *lbest* list is used for the update of the p_g value and the determination of the global best position. This procedure by itself can alter the *lbest* list (by the extra replications executed) and as a result the *lbest* list is also updated at the end of this procedure. If p_g meets the performance criterion then it is returned along with the corresponding solution. Otherwise, having the updated global best p_g and the updated local best values p_i (i=1,2,…,k) PSO is ready to produce new candidate solutions. Once the available budget is exhausted the best found solution found so far is returned to the user along with its value, even if it does not meet the performance criterion.

Note: For each candidate solution, as well as for *lbest* list, we store not only the mean values but also the corresponding standard deviations and the total number of replications, since they are integral parts of the OCBA procedure. These lists are updated after each extra replication

3.4. The Interface Component

In the presented architecture we consider two interfaces (see Fig. 2): the first interface connects the simulation module with the Model Base Management Server (MBMS) by means of the Dialog Generation/Management Server (DGMS) module; the second one connects the DSS and the real system through the MBMS.

In particular, the ICS module represents the information system of the whole infrastructure and is the interface between the real system and the information system. It updates the system status stored in the Data System in real time.

The DGMS allows the communication between the decision component and the simulation component. It receives requests from the decision component such as running a new simulation or retrieving the results of a finished simulation. All requests coming from the MBMS are loaded into the DBMS by the DGMS. In order to guarantee the modularity of the architecture and the possibility to execute the DSS components under different platforms, we implement the following three different ways to communicate with the DGMS:

- *Web Service (WS)*. The WS can trigger a new simulation, provides information about the state of the request and provides the simulation results. A web service is a process running on a server and allowing a client to execute a process on the server. This kind of application communicates by the Simple Object Access Protocol (SOAP).
- *TCP Socket*. This application is a process running on the server that creates a listener on a preconfigured communication gate. The MBMS sends messages to DGMS by the TCP (Transmission Control Protocol) channel. These messages are strings codified by a prefixed communication protocol. By this application the MBMS can request a new simulation or access to stored data.
- *File Watcher*. This process monitors the files on a shared folder. As soon as a new file is loaded by an FTP client, it analyses the contents of the file and executes the corresponding operations. When a simulation goes to an end, the file watcher creates a report file in the same folder so that all data are available to the client.

Furthermore, the MBMS shows data to the DM: if the current performance of the system is not satisfactory, then the DM can decide to evaluate the impact of some decision using a "what if" approach. The MBMS allows the DM to run a simulation directly without the decision module control. This proposed solution is very useful for the DM and contributes to generalize all the features offered by the DSS.

3.5. The Decision Component

The MBMS server implements the decision component. In particular, it monitors the system state stored in the Data System component and decides if it is necessary to run a new simulation or not. Indeed, if the performances of the system decrease, then the decision component starts the optimization and the decision procedure. Figure 4 shows how the used decision approach is based on the optimization module and the validation by the simulation. More precisely, the optimization algorithm proposes some solutions that are sent to the simulation module. The ICS provides the current state of the system by returning the values of the variables that cannot be determined by the DSS that invokes the simulation module: the simulation starts and applies the proposed management strategies. The obtained performance indices allow evaluating the impact of the proposed solution on the system. Then a new set of candidate variables is passed to the simulation model and the process continues till the algorithm leads to a satisfactory decision described by a set of candidate variables. In this model, the candidate variables are the controllable inputs that may change in the tactical or operational decision making process. The proposed approach is general and it not strictly correlated to a specific case study.

In the proposed DSS, the hybrid optimization module (PSO+OCBA) is connected by a web service to the simulation module. The optimization module sends inputs and the number of replications for each candidate solution/design and it records the outputs, again through the web service. On the basis of the current state of the system and the design variables, the DSS can be used to take decision both at the tactical and operational levels.

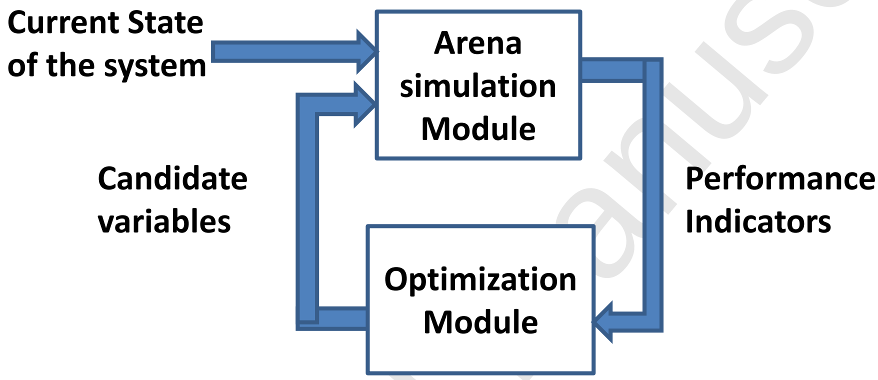

Figure 4. The decision making approach.

Moreover, in order to give a description of the DSS activities and actions and in particular how the decision component interacts with the simulation and optimization module, we use the UML activity diagram shown in Fig. 5 that gives a description of the action sequence that the components of DSS have to perform. The use of swim lanes in the activity diagrams allows easily showing, which part of the system is responsible in each phase of the activities. Figure 5 shows that DSS activities start when the DM sends a new request by the user interface and specifies the objectives. Then, the decision component retrieves data about the system from the data component, analyzes them and compares the system performances with the objectives. Successively, the decision component decides if it necessary to start an optimization process and a run a new simulation. In this case, the model component simulates the system and estimates the performances. If the results are satisfying, then the decision is submitted to the DM by the user interface, otherwise a new decision is evaluated.

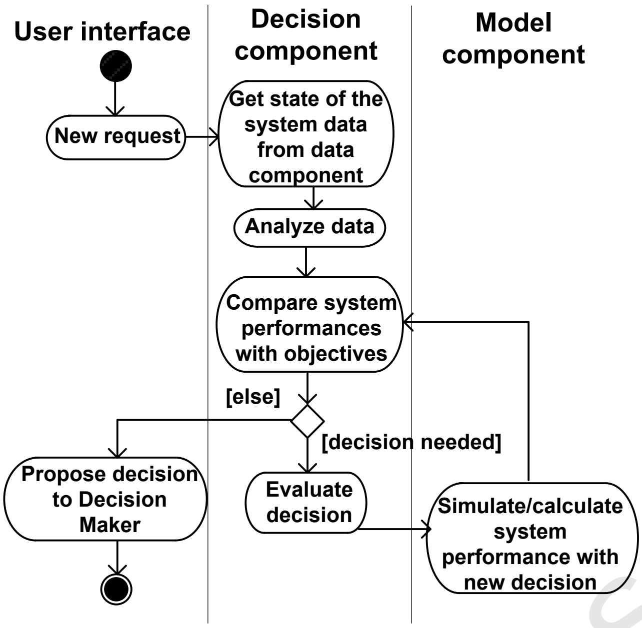

Figure 5. The activity diagram of the DSS.

4. Case study

4.1. System description

This section specifies the DSS with the proposed structure in a specific real case study that includes the logistics network of the Port of Trieste, the dry port area of "Fernetti" and the ground connection.

The logistics system in the Friuli Venezia Giulia region (Italy) is particularly significant both for its geographical location, as the meeting point of the trans-European Corridor V and the Adriatic Corridor, and for its concentration of ports and land, sea and railway transport networks. A requirement analysis identifies two different configurations for the specific test case: one for the containers and one for the trucks. In the port of Trieste, the traffic of trucks directed to Turkey through a Ro-Ro service represents a consolidated traffic. The containers have large areas to be warehoused in the port. On the contrary, a limited space is dedicated to the truck parking area. Hence, the study of the optimization of the truck paths between the port area and the dry port area is crucial and needs the application of suitable management strategies.

The intermodal terminal of Fernetti includes two different areas: an area of 120 parking places for the trucks that have to perform customs clearance operations. For each ship about 15-20 trucks perform the customs clearance in this area. A second area is devoted to the trucks that are waiting to be called in the port: it has 252 places and it represents the zone that has a function of the real dry port area.

Of the total number of trucks arriving at the port of Trieste, the 30% of them pass through Fernetti. In particular, there is a local regulation imposing that all the complete trucks have to stop at Fernetti. Therefore, the remaining truck trailers that have to be loaded on the Ro-Ro ship arrive directly at the port by motorway or by railway.

Normally a Ro-Ro ship transports 238 units, and one third of them are complete trucks. The volume of the traffic in 2010 was of about 105.000 loaded units (37.000 complete trucks and 68.000 truck trailers), divided in 15 ships per week. In normal conditions the waiting time before loading is about 25/30 hours but in case of congestion it can even reach 100 hours.

4.2. Flow of goods description by UML diagrams

The port of Trieste is a free port and the Port Authority has the role of controlling, coordinating and managing the port operations. In the analysis of the case study, we consider the flow of goods that are managed by the following actors:

- final customer: there are several customers involved and the flow of goods always begins with an order of the customer;
- the driver: it is in charge to move freight and trucks during the transportation phase;
- shipping agent: it operates as intermediary, taking care of authorization and booking procedures. The shipping agent also has a key role in the organization of the flow of goods and information;
- terminal operator: it provides a full range of additional services including container freight station, warehousing and storage, survey, container repair and maintenance and dedicated areas. Goods transported in containers are unloaded in the terminal area;
- customs: the Custom Agency of Fernetti and Trieste is the authority responsible for collecting and safeguarding customs duties and for checking the flow of goods. The customs clearance procedures can be performed at the origin of the flow, in Fernetti or in the Port of Trieste.

The current export flow of goods that is considered for this case study is divided in the following phases:

1. *New order from the customer*: if the proprietary of the goods decides to perform all the customs clearance procedures in the plant, the domiciled procedure authorized by the Customs Authority is carried out. Otherwise, the customs clearance will be performed in another point of the transportation flow.
2. *Choice of transportation mode*: the goods are ready to be sent. There are two different possibilities: through Fernetti or directly to the port.
3. *Arrival of trucks*: the flow is organized in two different ways depending on the type of means of transport to be boarded. More precisely, a complete truck (with the trailer and the cab) has to stop in the dry port area before entering the port; a trailer has the possibility to choose either to go directly to the port or to pass through the dry port

area (typically these represent the 30% of the total flow of trailers). When a truck arrives at the Fernetti intermodal terminal, its arrival is registered.

4. *Inside the Fernetti terminal*: if goods are already cleared, then there is a truck parking area, where truck is waiting to be called for the boarding in the port of Trieste. Otherwise, there is a dedicated area where the truck will be moved and the customs clearance procedures will take place.
5. *Customs clearance procedures*: the bill of loading and the "cargo manifest" are transferred to the customs in order to transfer information about all the transported goods. Successively, the customs duties are paid.
6. *Booking*: when a truck arrives at Fernetti, the truck driver books a place on a Ro-Ro vessels and gives to the shipping agents all the needed documents. When the ship arrives at the port and it is ready to be loaded, the truck driver receives a communication through a variable message panel. At this point, the truck driver receives back a certificate that enables him to go in the port.
7. *Transportation phase to the port*: the truck driver leaves the Fernetti terminal and goes to the port of Trieste. In order to avoid too many delays, each truck driver has an hour and a half to reach the port.
8. *Security checks*: frights arriving at the port may have to be checked by the customs. If the freight has to be checked, the truck driver has to move the truck to a special area for the security check operations, made by the Customs Agency and by a Customs Anti-Fraud Service.
9. *Boardin*g: at this point of the flow the trucks or the trailers are ready to be loaded on the ship.

The described phases of the export flow of goods are described by the UML activity diagram shown in Fig. 6 that points out the actions and the actors of the flow. In particular, the diagram reports the phases from phase 2 (*arrival trucks*) to phase 9 (*Boarding*) and in the considered application the shipping agent operations are performed b the terminal operator.

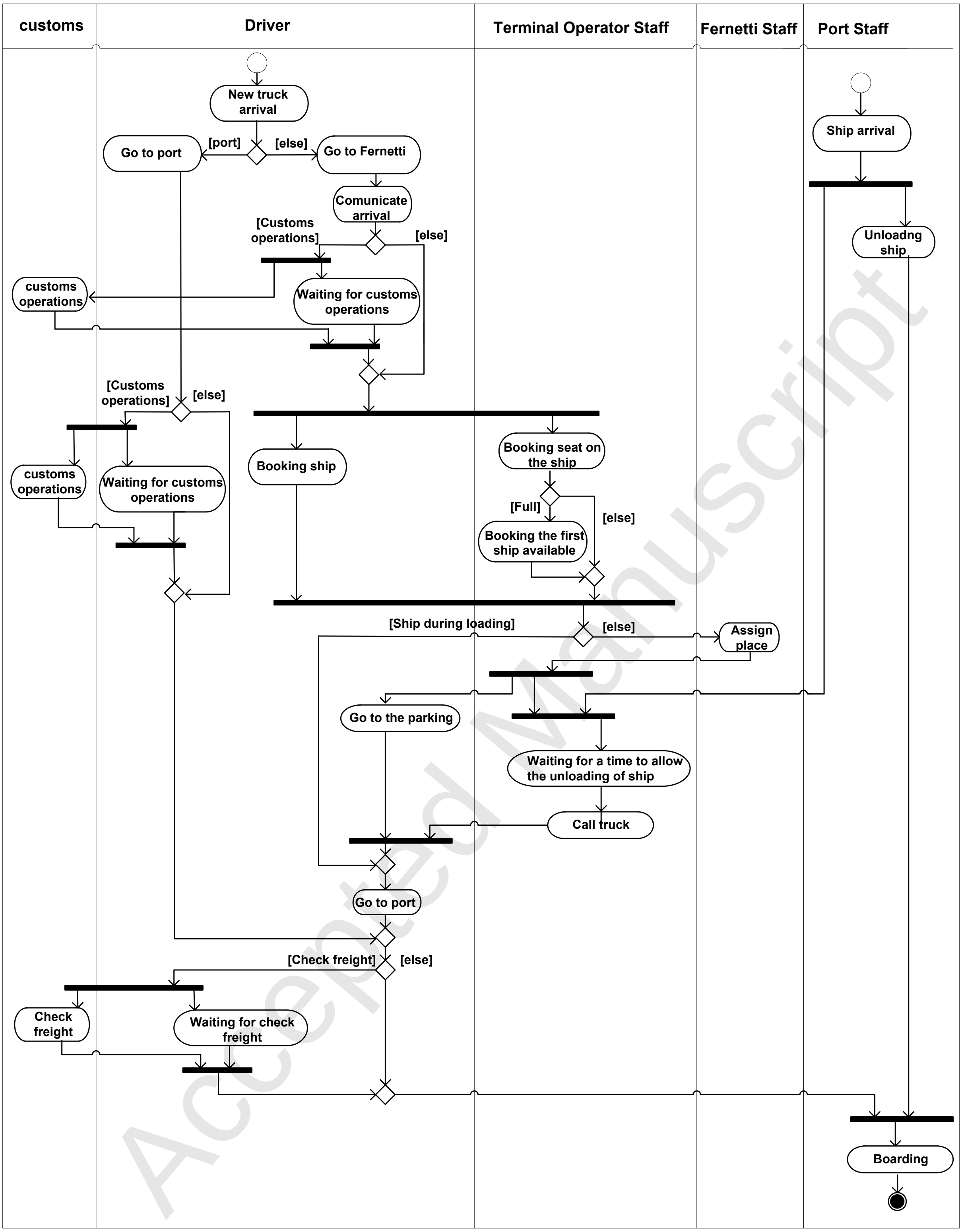

Figure 6. The activity diagram of the export phases of the case study.

5. The DSS Model Component for the Case Study

In this section we present the model component for the considered case study. In particular we focus on the simulation and the optimization modules specification. Moreover, in order to show the effectiveness of the proposed DSS, we present two different applications: the first one considers decision at the tactical level and the second one at the operative level.

5.1. Simulation module description

The presented model that is described in the UML framework is implemented in the Arena environment [15] that is a discrete-event simulation software particularly suited for dealing with large-scale and modular systems. The general-purpose feature of the Arena environment is suitable for logistics and heterogeneous systems because it can fit all different kind of activities performed during the whole delivery process.

The simulation approach used by Arena is process-oriented, since the overall behavior of the considered system is described by the interaction of different processes and represented by a flow chart with different shapes for different functions.

The activity diagrams can be easily used to generate the Arena simulation model that can be straightforwardly implemented by the following three steps [7]:

- the UML activity diagrams are translated in the Arena simulation environment. To this aim, the Arena modules are associated to the UML activity diagram elements, by establishing a kind of mapping between each Arena module and the UML graphical element of the activity diagrams;
- the simulation parameters are included in the Arena environment: i.e., the activity times, the process probabilities, the resource capacities, the average input rates are assigned. These specifications can be modified in every simulation and enable the choice of the scenarios in the case study actual implementation and management;
- the simulation run of the experiments is singled out. In this step the performance indices are determined and evaluated with suitable statistics.

Figure 7 shows a snapshot of the model implemented in Arena environment depicting the main components of the system:

- the Fernetti inland terminal is described by two areas: Area 2 for units that have to do the Customs operation and Area 1;
- the transport system is the stretch of highway connecting the Fernetti area to the Port;
- the port area, including Railway, the Customs Authority office, the checking area, the boarding zone and the bay.

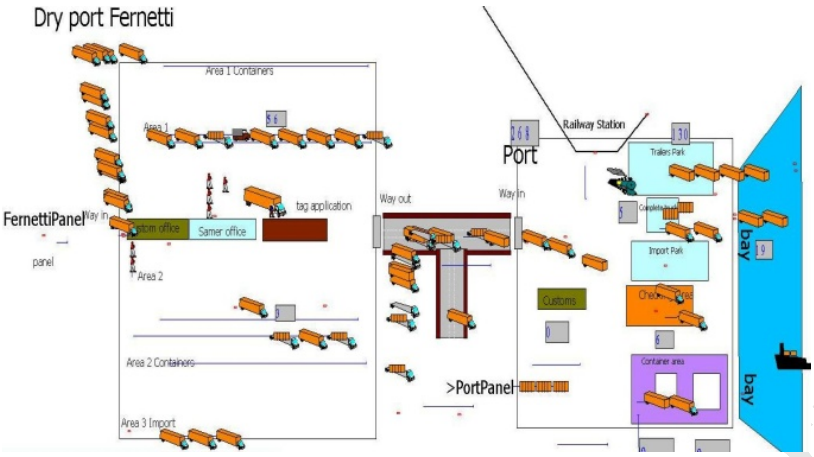

Figure 7. The snapshot of the Arena Model.

With the aim to analyze and improve the system behavior, a set of performance indices are selected:

- *system throughput*, i.e., the average number of transportation units delivered per time unit by the inland terminal;
- *lead time (LT)*, i.e., the average time elapsed from the entrance of a units in the system to the ending of boarding operation;
- *average utilization of the resources,* i.e., the ratio between the average time in which the resource is occupied and the total simulation time.

5.2. Optimization module description

Following the schema described in Fig. 4 the PSO algorithm proposes different candidate solution and passes them to the Arena model that estimates the performance indicators. The performance indices are determined by a simulation run starting from the beginning of phases 1 (new order from the customer) and end with phase 9 (boarding).
In particular we can distinguish between two different inputs

- Candidate variables: these variables are proposed by the PSO algorithm and represent the control variables such as the size of parking and the number of operators available for each different actors involved in the process.
- Current state of the system: these variables represent the unpredictable events that cannot be fixed by the PSO algorithm, e.g. the inter arrival time of a new order or the number of place actually occupied for each parking area.

The arrival time instants of transportation units are randomly generated by an exponential distribution of mean of 137 units/day. In addition, the processing times of the phases described in this section are shown in Table 1 and have a triangular distribution. In particular, the second column of Table 1 reports the modal values δ of such distribution, the third and fourth columns show the maximum and minimum values of the range in which the processing time varies, denoted respectively by D_δ and d_δ. Moreover, the last column of Table 1 reports the number of infrastructure operators, denoted by *Op*, that are necessary to perform the corresponding operation.

Operation	δ (t.u.)	D_δ	d_δ	*Op*
New order from the customer	30	180	25	1
Checks at Fernetti entrance	5	6	4	1
Payment	15	30	12	1
Transport authorization	15	120	12	1
Inside the Fernetti terminal	15	30	12	1
Customs clearance procedures	15	30	12	1
Booking	30	180	25	1
Transportation phase to the port	30	120	25	1
Security checks	30	40	24	1
Boarding	120	144	96	1

Table 1. The triangular distribution of processing times and number of necessary operators

All the performance indices are evaluated by a long simulation run of 540000 t.u. (equal to 12 months and 15 days, if we associate one minute to one t.u.) with a transient period of 21600 t.u. In particular, the number of replications is estimated by the OCBA approach in order to guarantee a 95% confidence interval. Table 2 reports the parameters used in the PSO algorithm.

Parameter	Value
c1	2
c2	2
χ	0.729
Max Iteration	100

Table 2.Value of parameter used in the PSO algorithm.

In order to show the effectiveness the proposed approach, we present two different applications: the first one tries to choose the best size of the parking areas in order to maximize the system throughput. The second one chooses the best number of operators for each actors involved in the process in order to minimize the system lead time.

5.3. A DSS Application at the Tactical Level

In this case, we consider the problem connected with a forecasted increase (double number of transportation units) of the truck flow. Then, the DSS has to deal with the following decision: determining the optimal number of locations in the parking areas 1 and 2 of Fernetti terminal in order to maximize the function value expressed by the system throughput. In this application, the number of operators does not change such that only the number of available places in each parking area can change by means of PSO optimization procedure. Under these assumptions the particle position is represented by the number of available places in Area 1 and Area 2 at Fernetti.

This kind of decision is referred to a long term investment due to the costs that have to be faced. For this reason we consider it a tactical level decision.

Table 3 reports the actual number of available places, the maximum and the minimum allowed values. Moreover, Table 4 reports the values proposed by the DSS as a result of the optimization and simulation procedures.

At the tactical level, the solution offered by the DSS shows an increment in Area 1 as well as in Area 2 to obtain a throughput of 229,5 units/day. More in detail, the DSS suggests a higher percentage increment of Area 1 (+57%) with respect to Area 2 (+23%): this result was expected because the custom operations are performed in Area 1.

Resources	number	Max	Min
Area 1 location	120	240	120
Area 2 location	252	500	252

Table 3. Number of location available

	Area1 location	Area 2 location	Throughput Units/day
Solution	189	310	229.5

Table 4. Values suggested by the DSS

Resources	number	Max	Min
Forwarders in Fernetti	6	10	1
Customs staff in port	2	10	1
Port area staff	8	20	1
Customs staff in Fernetti	4	10	1
Area staff in Fernetti	6	20	1
Forwarders in port	6	6	6

Table 5. Number of available operators.

The first column of Table 5 reports the number of operators for each resource.

We remark that in the presented case study, the PSO+OCBA algorithm determines the system throughput (function value) for a single particle position by 14 iterations replications in the worst case. Finally, the total decision process requires 33 minutes.

5.4. A DSS Application at the Operational Level

At the operation level the DM needs to evaluate the optimal number of operators that should daily work in the port and in the Fernetti terminal, in order to minimize the function value expressed by the average lead time. This decision is based on the expected flows of trucks for the successive day: we assume an input flow of 137 trucks/day.

In this application the number of places in the Fernetti areas does not change and the number of operators can change by means of PSO+OCBA optimization procedure. Table 3 shows in the third and forth column the maximum and minimum possible numbers of the operators, respectively. Hence, the particle position is represented by the number of available operators in the Fernetti terminal.

The actual average lead time is about of 750 minutes (12.5h). On the other hand, Table 6 shows the "optimal" number of the operators proposed by the DSS: a total number of 33 operators (only one more) and a suitable re-assignment of the workers to the teams. Considering such number of operators the average lead time results reduced to 534 minutes (about 9h).

We remark that in the presented case study, the PSO-OCBA algorithm determines the system throughput (function value) for a single particle position by 8 iterations in the worst case and the total decision process requires 32 minutes.

Resources	Number
Forwarders in Fernetti	10
Customs staff in port	3
Port area staff	6
Customs staff in Fernetti	2
Area staff in Fernetti	6
Forwarders in port	6

Table 6. Results for the operational level

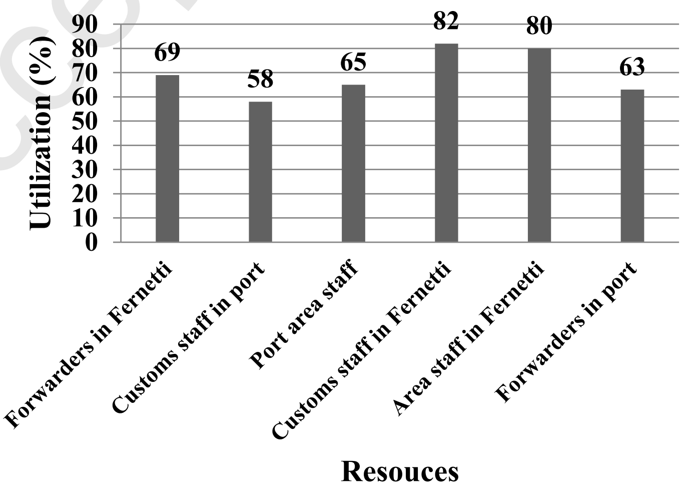

Figure 8. The percentage utilization of resources.

Moreover, Fig. 8 shows the average percentage utilizations of the resources. We can notice that it is sufficiently high: indeed, the lead time does not decrease if we try to increase the number of resources. This means that to further reduce the lead time it is necessary to modify the system dimension at the tactical level, for instance increasing the parking area or reducing the loading/unloading operation times.

6. Conclusions

This paper presents the architecture of a DSS to be used by DMs that have to take operational as well as tactical decisions in logistics systems. In particular, the main modules of the proposed DSS are specified: the simulation model and the optimization module. The simulation module is realized by a discrete event simulation and the optimization module uses the PSO metaheuristic algorithm combined with the OCBA scheme. The integrated employment of the two modules allows the DMs to take decisions by optimizing suitable performance indices.

A prototype of the designed DSS is applied to a case study that is constituted by the Trieste port and the dry port of Fernetti, in the north of Italy. The results of the decision procedure show how the management strategies are of basic importance to improve the system performances. Moreover, two different use cases have been presented for the tactical and the operation levels. We highlight that, thanks to DSS general architecture, it is very easy to pass from the tactical level use case to operational level one.

The presented discrete event simulation study shows that the suitable application of the modern ICT based solutions has a huge potential for efficient real time management of transport systems, reducing the lead times in the port and dry-port areas. Moreover, the simulation results allow proposing a reorganization of the workflow in order to suitably utilize human resources.

Hence, the proposed DSS can be efficiently applied in order to improve the logistics services by employing new ICT management tools.

Future research will focus on the DSS specification for new complex decisions such as multimodal services with environmental and sustainable objectives.

ACKNOWLEDGEMENT

This work was supported by the E.U. FP7–PEOPLE–IAPP–2009, Grant Agreement No. 251589 (SAIL) and the E.U. CIP-ICT-PSP-2013-7, Grant Agreement No. 621112, COoperative loGISTICS for sustainable mobility of goods (CO-GISTICS).

REFERENCES

[1] V. Boschian, M. Dotoli, M.P. Fanti, G. Iacobellis, and W. Ukovich, A Metamodeling Approach to the Management of Intermodal Transportation Networks, IEEE

Transactions on Automation Science and Engineering. 8(3) (2011) 457-469, ISSN 1545-5955.

[2] V. Boschian, M.P. Fanti, G. Iacobellis, G. Georgoulas, C. Stylios, W, Ukovich, A decision support system for intermodal transportation networks management, Proceedings of the 25th European Modeling & Simulation Symposium (Simulation in Industry), 25 - 27 September 2013, Athens, Greece, 2013.

[3] C.H. Chen, L.H. Lee, Stochastic Simulation Optimization: An Optimal Computing Budget Allocation, World Scientific Publishing Co., 2010.

[4] C. H. Chen, J. Lin, E. Yücesan, and S.E. Chick, Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization, Journal of Discrete Event Dynamic Systems: Theory and Applications. 10 (2000) 251-270.

[5] M. Clerc, J. Kennedy, The particle swarm–explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput.. 6(1) (2002) 58–73.

[6] M. Dotoli, M.P. Fanti, A.M. Mangini, G. Stecco, W. Ukovich, The Impact of ICT on Intermodal Transportation Systems: a Modelling Approach by Petri Nets, Control Engineering Practice. 18 (8) (2010) 893- 903 .

[7] M.P. Fanti, G. Iacobellis, G. Georgoulas, C. Stylios, and W. Ukovich, A Decision Support System For Intermodal Transportation Networks Management, Proceedings 24th European Modeling & Simulation Symposium, September, 19-21, 2010, Vienna, Austria, 2010.

[8] J.-L. Gallego, J.-L. Farges, J.-J. Henry, Design by Petri nets of an intersection signal controller, Transportation Research Part C. 4(4) (1996) 231-248.

[9] G. Giani, G. Laporte, R. Musmanno, Introduction to Logistics Systems Planning and Control,Willey (2004).

[10] W.D. Kelton, R.P. Sadowski, N.B. R. Swets, Simulation With Arena, 5th ed. MA: McGraw-Hill, Boston (2009).

[11] J. Kennedy, R.C. Eberhart, Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks, Perth (Australia). (1995) 1942-1948,

[12] L. Lättilä, V. Henttu, O.-P. Hilmola, Hinterland operations of sea ports do matter: Dry port usage effects on transportation costs and CO_2 emissions, Transportation Research Part E. 55 (2013) 23-42.

[13] F. Parola, A. Sciomachen, Intermodal container flows in a port system network: analysis of possible growths via simulation models, Int. Journal Prod. Econ. 97 (2005) 75–88.

[14] K. E. Parsopoulos, M. N. Vrahatis, Particle Swarm Optimization and Intelligence: Advances and Applications, Information Science Publishing, IGI Global, 2010.

[15] D. Power, Decision Support Systems: concepts and resources, Quorum Books. Westport, USA (2002).

[16] L. Ramstedt, J. Woxenius, Modelling Approaches to Operational Decision-Making in Freight Transport Chains, Proc. 18th NOFOMA Conference, Oslo, 7-8 June 2006, Norway.

[17] V. Roso, Factors influencing implementation of a dry port, International Journal of Physical Distribution and Logistics Management. 38 (10) (2008) 782–798.

[18] V. Roso, J. Woxenius, K. Lumsden, The dry port concept: connecting container sea ports with the hinterland, Journal of Transport Geography. 17 (5) (2009) 338–345.

[19] R. Stahlbock, S. Voß, Operations research at container terminals: a literature update, OR Spectrum. 30 (2008) 1–52.

[20] D. Steenken, S. Voß, R. Stahlbock, Container terminal operation and Operations Research—a classification and literature review, OR Spectrum. 26 (2004) 3–49.

[21] S. Vasilakos, G. Iacobellis, C. D. Stylios and M.P. Fanti, : Decision Support Systems Based on a UML Description Approach, In Proceedings of 6th IEEE International Conference on Intelligent Systems IS'12, 6-8, 2012, September Sofia, Bulgaria, pp.41-46.

[22] I.F.A. Vis, R. de Koster, Transhipment of containers at a container terminal: an overview, Eur J Oper Res. 147 (2003) 1–16.

- The paper presents a Decision Support System (DSS) to manage logistic systems.
- The DSS architecture is specified and the main components are designed.
- An integrated approach is proposed using simulation and optimization modules.
- The case study constituted by the export flows of freights between the dry port and the port Trieste (Italy) is analyzed.