Università della Calabria

Dipartimento di Elettronica, Informatica e Sistemistica

Doctor of Philosophy in Operations Research (MAT/09) – XXI edition

PH.D. DISSERTATION

Simulation-based Optimization in Port Logistics

Rina Mary Mazza

Rina Mary Mazza

Supervisor:

Prof. Lucio Grandinetti

Lucio Grandinetti

Advisor:

Prof. Pasquale Legato

Pasquale Legato

Academic Year 2007/08

To Jesus and my parents.
They're all I need to get by.

Introduction

The forecasted growth rate of containerized trade for the future years has driven competing container terminals to enhance their individual ability in fulfilling customer demand with high standard quality service, while keeping operations lean. A natural corollary to this level of commitment is found in the daily pursuit of terminal managers to obtain a nearly seamless system and maintain its operational efficiency. A similar goal calls for the use of systematic design and verification methodologies that can cope with the major sources of system complexity and return “reliable” measures for terminal performance such as container throughput, vessel/vehicle turn-a-round time and/or unproductive times.

In modern container terminals the use of Operations Research methods and models as a response to this quest is becoming a rather “large” issue. Indeed, container terminal logistics have received great interest in the scientific literature from both the theoretical and practical standpoint. In most cases, the common approach to problem design and solution is based on decomposing the original problem into several related smaller models. However, a satisfactory contribution to these complex, dynamic and random-based logistic problems often requires the combination of multiple stand-alone OR techniques to deliver overall system performance measures. This awareness leads to the introduction of an integrated methodology which can significantly aid decision-making under uncertainty.

Chapter 1 of this dissertation first introduces the three major operational concerns in a maritime container terminal (i.e. the *berth planning*, the *quay crane scheduling* and the *yard management*) and discusses how in related precedent industrial-oriented R&D activities three queuing-based network models were solved via simulation to investigate performance evaluation. It then illustrates how, for each problem, the practical need to extend simulation research efforts beyond a classical *what-if* approach naturally yields a more promising methodology, focused on systematic moves for optimum-seeking, also known as *simulation-based optimization*.

Chapter 2 focuses on the prospect of selecting the “best” among k simulated competing designs, policies or system configurations according to a pre-assigned level of probability. To this end, the major *Ranking and Selection* (R&S) approaches are illustrated for cases in which all alternative simulated system designs are known in advance. In particular, two newly proposed so-called indifference-zone based R&S procedures are presented. The first builds an “artificial” process with the same mean as the output observations of interest, but with a smaller variance. The second bases its sampling process on the corresponding variance behavior and uses a variance-weighted decisional mechanism during simulation run. Both algorithms are compared with the performances of some previously discussed classical R&S statistical techniques.

Chapter 3 deals with the case in which a combinatorial, unknown number of simulated configurations needs to be explored. A *simulated annealing* (SA) algorithm is introduced as *system generating algorithm* (SGA) to sequentially reveal k different systems configurations (with $k \geq 1$) during a simulation run. An in-depth description of the SA algorithm and its properties follows, along with a discussion on the practical limits experienced when customizing this approach to the study of a well-known problem in port logistics: the *quay crane scheduling problem*. The closing paragraph presents an integrated SA-R&S procedure and preliminary numerical experiments for the above problem.

In the final chapter, simulation-based optimization models are integrated and applied to the container terminal in Gioia Tauro, Italy. The application is required by the top manager who is currently evaluating a hypothesis of reorganization in the yard system infrastructures in conjunction with alternative operational policies and procedures pertaining to the yard area and all bordering zones. The proposed simulation-based optimization approach benefits from suitably designed advanced statistical methods for input and output data analysis.

Index

Chapter 1
Optimum-seeking by simulation issues in maritime container terminals

1.1 Introduction

Container terminals are multi-modal facilities serving the role as interface between sea and land-based container transport. Their *core business* consists in providing a wide range of integrated services across the logistic chain of container movement (e.g. handling, stacking, inspection, inter-modal dispatching, etc.).

According to the figures provided by UNCTAD (2004), containerized trade, measured in TEUs (twenty-foot equivalent units), is forecasted to grow by an average annual rate of 5.32% over the next two decades. As a result of this trend, the number of container terminals worldwide keeps increasing and competition has become both price driven and service driven. In this market struggle among container terminals, the success of individual companies mostly depends upon their ability to fulfill customer demand with high standard quality service and keep their operations lean; otherwise, they are bound to lose clients to competition.

In response to this pursuit, more and more terminal managers are drawing-up investment plans and seeking funding to improve the performance of their main logistic processes in terms of both operational efficiency and asset utilization, while saving on costs and risks. In this concern, companies are not alone: Operations Research (OR) has become a very valuable reference for those who intend to acquire and apply successful practices and support complex decisional processes based on the results of research and development studies.

The use of Operations Research methods and models in modern container terminals and operations management problems is exhaustively discussed in recent literature survey papers (Vis and de Koster 2003, Steenken et al. 2004, Stahlbock and Voß 2008). A wide variety of different approaches are presented for process optimization including exact methods, heuristic methods as well as simulation approaches. Due to the inherent uncertainties in both terminal (sub)system and input data modeling, growing attention is getting paid to simulation-based techniques, which are especially suitable for both estimating the performance measures of interest in these complex facilities and supporting decision-making processes at a strategic, tactical and operational level.

This logic is also the bottom line shared throughout the entire thesis. This chapter first introduces three of the major operational concerns in a container terminal: the *berth planning*, the *quay crane scheduling* and the *yard management*. Then, for each of these problems, an emerging awareness is addressed: whatever stand-alone OR technique is being used, especially in industry-oriented R&D activities, it needs to interact with other methodologies to give a satisfactory, practical contribution to the complex logistic problems at hand and, thus, deliver maximum results for each of them. This understanding leads to the introduction of simulation-based optimization, an integrated methodology which can significantly aid decision-making under uncertainty.

1.2 Berth planning

A berth is a properly-equipped space in a harbor beside a quay or pier where a vessel can be moored for discharging and/or loading (containerized) cargo, as shown by Figure 1.1. In a container terminal, this limited resource is usually organized in a number of segments and/or slots to be assigned to incoming vessels.

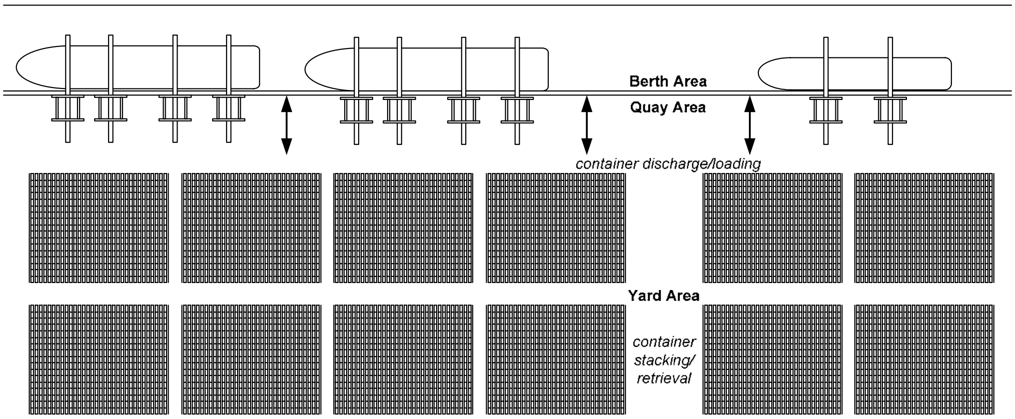


Figure 1.1 - Bird view of a berth in a container terminal and its bordering areas

The choice of berthing one vessel rather than another into a specific slot could depend on a variety of factors including:

- *contractual agreements* (e.g. in respect of formal guarantees, likely based on fixed time windows or priority mechanisms, granted to certain vessels);
- *technical feasibility* (e.g. a berth segment must be physically compliant with the requiring vessel's length and draft and, at the same time, not disregard the minimum-security distance between current or potential neighboring vessels);
- *operational efficiency* (e.g. to minimize the distance between the berth segment assigned to a candidate-vessel and the source/destination point of the vessel's container stacking area in the storage yard).

The above points help to clarify why terminal managers usually practice different berth assignment policies for different clients. "Primary" vessels are often entitled to receive reserved berth segments that are in close proximity to the yard area in which their container slots are located. On the contrary, the berthing requests of "secondary" vessels are fulfilled according to a "fill in" procedure along the free berth and on a first-in first-out basis between competing secondary vessels.

Practically, to avoid that an arriving primary vessel must wait in the roadstead until some secondary vessel releases its berth space, a priority mechanism is typically used. It can be based on a "look ahead" over the expected time of arrival (ETA) of primary vessels. In brief, the secondary vessel is forced to wait in the roadstead whenever its arrival instant falls too close to the ETA of the next primary vessel, making it impossible for the secondary vessel to leave the berth before the arrival of the next primary vessel claiming the same berth space.

With reference to this specific subsystem, in (Legato and Mazza 2001) a hierarchical model is proposed to estimate congestion effects on the sojourn time of vessels (customers) belonging to any given shipping company (class of customers), out of a fixed number visiting the terminal. The outer model in Figure 1.2 focuses on the terminal admission policy: as on may observe, the berth slots serve as "passive resources" to be held before vessel entrance until vessel exit. This feature affects the berth assignment and, thus, admission to the port facility for secondary vessels whose arrival instant is close to the next arrival instant of a primary vessel. Therefore, it prevents an analytical solution of the outer queuing network model.

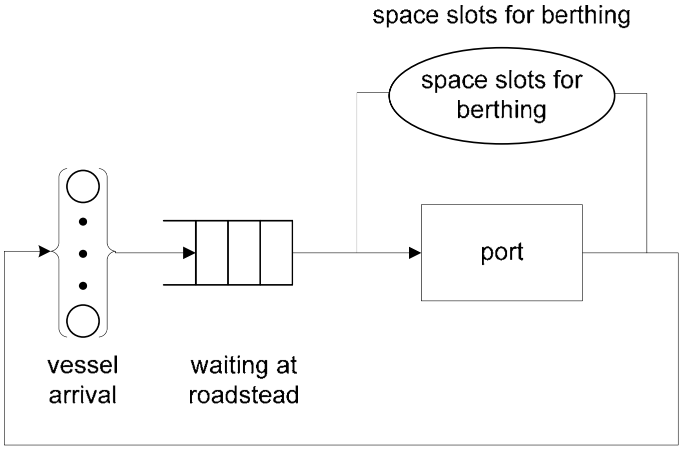


Figure 1.2 - The outer model

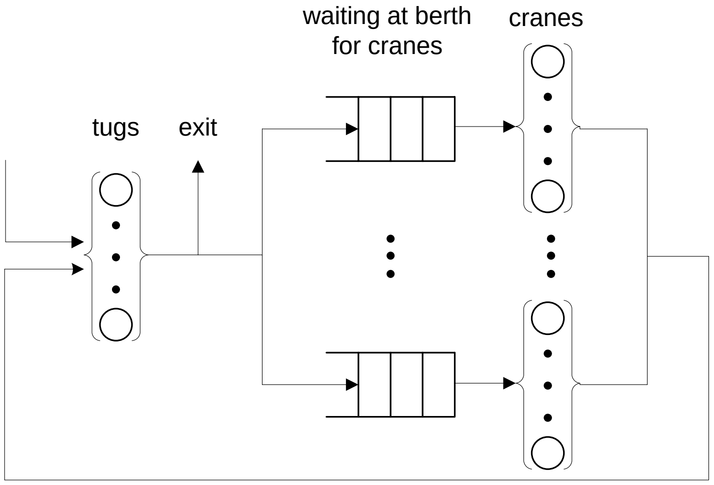


Figure 1.3 - The inner model

Resource representation in the inner model depicted by Figure 1.3 reveals that tugs are not a critical resource, while, in contrast, the limited number of quay cranes along the berth segments may generate waiting phenomena among the requiring berthed vessels. The overall closed queuing network model with multiple classes of customers is solved via simulation to represent a berthing policy with priorities, multiple crane (servers) allocation and non-exponentially distributed time between arrivals of major vessels. Some other queuing based simulation models for these core logistic processes at real container terminals can be found in (Silberholz et al. 1991; Gambardella et al. 1998; Yun and Choi 1999; Shabayek and Yeung 2002). However, all of them share the common lack of attention to the waiting phenomenon that arises when an incoming vessel stops in the roadstead, first to ask for a berthing position and then to receive one or more tugs and/or pilots that will maneuver it to the berthing position. This refinement is the subject of a more recent study (Canonaco et al. 2007) conducted on behalf of Medcenter Container Terminal S.p.A. (MCT), the company that manages the container terminal located at the port of Gioia Tauro in Southern Italy. In this work, a new queuing network model accounts for and satisfies the requirements proposed by MCT. In principle, a sort of semaphore-like device is considered to represent the holding time at the entrance channel (before berth assignment) and the cause of such delay, along with its random duration. The entire queuing network model holds in store a wide range of useful alternative configurations to be evaluated by simulation in order to support major strategic-tactical decisions. For example:

- What kind of admission policy should be adopted for vessels on wait in the roadstead?
- How many berth segments should be organized for active shipping services?
- How many quay cranes - out of the total, fixed number of these available resources - should be allocated along each of the above berth segments?
- In which segment(s) should vessels entering the port be berthed, provided that this decision may be based on some suitable attributes shared by any given subset of the active services?

To take simulation research efforts to a higher level, particular attention needs to be given to the statistical methods used to analyze simulation output. Indeed, the availability of credible simulation results can extend the use of a simulation model even beyond a classical *what-if* approach performed on the capacity planning of logistic resources or the management of logistic operations. The problem of finding a queuing system configuration that optimizes the expected value of some measures of system performance on the long-run, such as terminal *throughput* and *lead time*, is strongly demanded. More precisely, in future work the goal to pursue is twofold: on one hand, generate and simulate a sequence of system configurations (each corresponding to particular input settings) so that a configuration providing a near-optimal solution is eventually obtained; on the other, obtain such a solution with the least amount of computational expense possible (i.e. number and length of simulation runs).

1.3 Quay crane scheduling

The *quay* or ship operation area is one of the main sub-systems in a container terminal. In this area ship-to-shore cranes, such as *rail-mounted gantry cranes* (RMGCs) or *rubber-tired gantry cranes* (RTGCs), perform container discharge/loading operations from/on a vessel, while selected shuttle vehicles provide container transfer from the quay to the yard and vice versa. As in most European and North-American container terminals, the present description refers to a *direct transfer system* (DTS), which implies the use of *straddle carriers* (SCs) - special vehicles designed to pick-up/set-down and transfer one or more containers per time.

For both discharge and loading operations, a quay crane operates by moving (on wheel or rail) in horizontal directions to reach different holds within the same vessel or on different vessels. At the basis of each quay crane, a very restricted area (e.g. a 6-slot space) is naturally provided for buffering a limited number of containers. When performing discharge operations, a quay crane picks-up containers from the vessel and “feeds” them to straddle carries which provide for their direct transfer from the quay area to the assigned yard positions within the terminal storage area. As one may observe in Figure 1.4, the representation of this discharge process from the ship to the yard features a joining point in blue between the unloaded container (*TEUs waiting line under crane*) and the SC (*SC waiting line on quay*) sent for its pick-up (*set-up time*) and transfer to the yard.

As far as loading operations are concerned, a quay crane picks-up containers delivered from the terminal yard by the SCs and places them on the ship in the assigned vessel holds. Figure 1.4 accounts for this process from the yard to the ship as well: in particular, the forking point in red represents the physical separation carried out by an SC (*SC waiting line on quay*) when it first sets-down the container (*set-down time*) in the quay crane buffer area (*TEUs waiting line under crane*) and then returns empty to the yard to retrieve other containers.

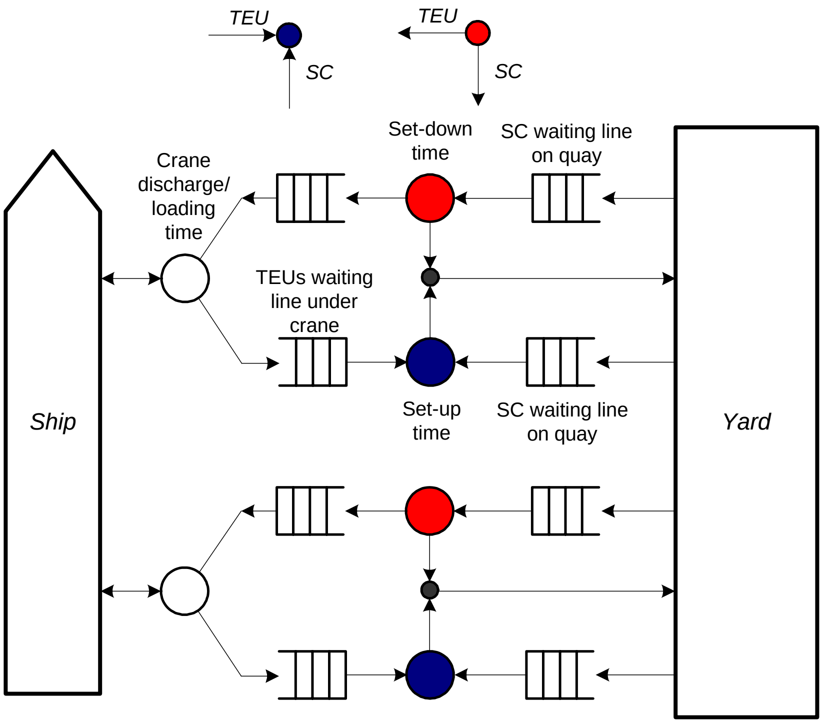


Figure 1.4 - The discharge/loading process

When a different number of cranes are called to work in parallel on the same vessel at the same instant (as required, for example, by contractual agreements), a well-known operational problem arises: the *quay crane scheduling problem* (QCSP). The objective of this problem consists in determining the so-called *crane split* or *crane schedule* or, in other words, which and in what order holds should be assigned to the single quay cranes to minimize the vessel's overall completion time (also known as *makespan*), provided that:

- a minimum distance must be left between quay cranes to avoid boom collision (*non-simultaneity constraints*);
- some vessel holds must be operated before others (*precedence constraints*);
- not every quay crane is always immediately available for requiring vessel holds (*release constraints*).

The solution of the QCSP has been successfully dealt with in literature by using both deterministic approaches (and solving the relaxation of the related

IP formulation) and metaheuristics algorithms (see for example Daganzo 1989, Kim and Park 2004, Lim et al. 2007, Sammarra et al. 2007). Nevertheless, it is worth observing that in real-life management of logistics at a maritime container terminal, the QCSP is not an isolated problem, but rather a decisional step within the entire discharge/loading process depicted in Figure 1.4. Therefore, dealing with this process as a whole, requires finding a solution to the QCSP within a wider OR-based methodology, as well as adopting cost-effective techniques in terms of results realism and quality of the solution returned.

The awareness of this methodological "lag" first occurred in a previous research by (Canonaco et al. 2008) that investigated the representational capabilities offered by consolidated modeling languages such as queuing networks (Gross and Harris 1998) and event graphs (Yücesan and Schruben 1992, Schruben 2000). Due to the complexity and the dynamic, non deterministic features of the discharge/loading process, at that time a discrete-event simulation approach was used to incorporate both the low level operational policies and work rules and the specific scheduling constraints involved in the vessel hold-quay crane assignment. During numerical experimentation, some *what-if* experiments were performed to obtain improved resource and system performance measures.

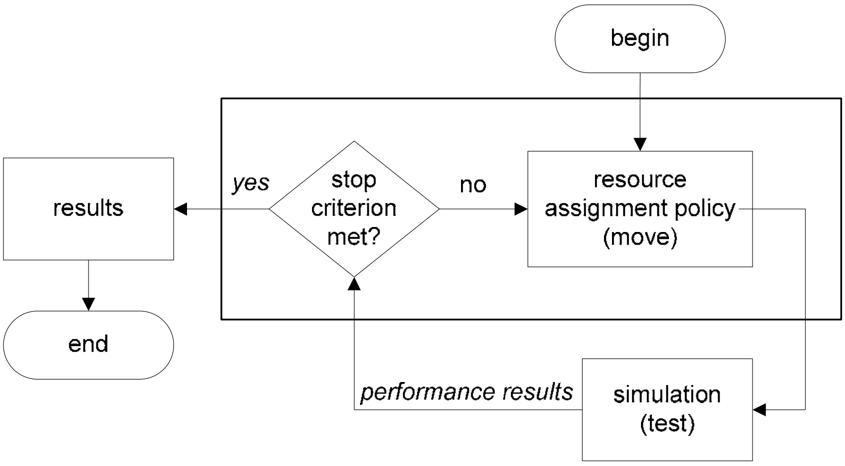


Figure 1.5 - The "test a move" approach by simulation

In particular, a “test a move” approach, based on a selective modification of either the crane-holds allocation schedule or the straddle carriers-crane assignment policy, was adopted. These *what-if* attempts were logically arranged as shown in Figure 1.5 and consisted in performing a manual local search on a few neighbor configurations of the schedule set by the end-user.

It then occurred that this schema could be envisioned as the basis to evolve from a *what-if* approach - suitable just for evaluating the goodness of an existing praxis - to a *what-to* approach for determining a (sub)optimal solution in new system design. To the latter case, the above manual local search would have to be replaced by an optimization procedure aimed to first generate an initial feasible configuration and then explore the whole feasible region until no further improvements of the *makespan* could be obtained or no further computation time remained. Once again, a similar methodology, focused on systematic moves for optimum-seeking, called for the use of both simulation and optimization.

1.4 Yard crane management

The role of a yard in a maritime container terminal is to provide storage space for containers from their import by truck to their export by vessel and vice versa, and during their (pure) transshipment from vessel to vessel.

The present description is referred to a pure transshipment terminal, where a unique storage area is shared among a certain number of shipping companies to which properly sized portions of the yard must be assigned to stack/retrieve container batches (i.e. a set of containers sharing some common properties). Such a situation actually occurs at the port of Gioia Tauro, where oceanic (mother) vessels from the Eastern Asia maritime route discharge containers mainly addressed to other ports in Northern Europe. Due to the lack of adequate rail services and road infrastructures, containers stored on the Gioia Tauro yard site are almost completely retrieved by secondary vessels (both dedicated and common feeders).

Generally speaking, from an organizational point of view, a yard may be divided into large areas called *zones*. In each zone, containers are stacked into *blocks*. As shown in Figure 1.6, a block has: a number of *lanes* or *rows* ranging from 6 to 13 (eventually plus one if transfer occurs by internal truck) placed side by side; usually 5 containers in height called *tiers* for each lane; 20 or more containers in length. A vertical section of a block (e.g. 5 tiers * 6 lanes) is normally referred to as *bay*.

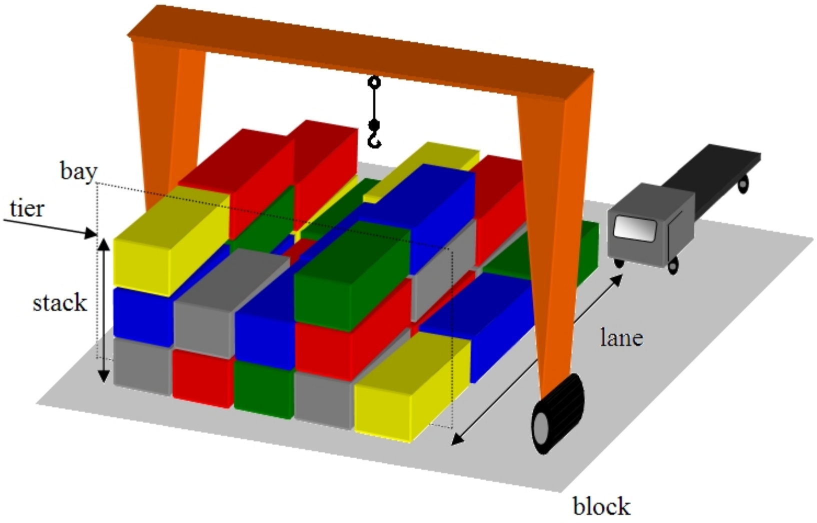


Figure 1.6 - Organization of a yard operated by cranes

If container stacking/retrieval on the yard is performed by transfer cranes, such as RMGCs or RTGCs, then a common operational issue consists in periodically deciding how many and which cranes are to be assigned to a block. This decision usually depends on the expected daily workload in each block and, therefore, on the total crane capacity (measured in time units) required to complete container stacking/retrieval operations. To do so, cranes must be transferred from one block to another. For example, Figure 1.7 illustrates how RTGCs can travel between adjacent yard blocks without any turning motion (e.g. from block 1 to block 3) or by changing lanes (e.g. from block 1 to block 4). In the former case, crane transfer can take about 10 minutes; in the latter, about 5 additional minutes are required to perform 90 degree turns. These movements are exclusively referred to inter-block (and not inter-zone) crane transfer.

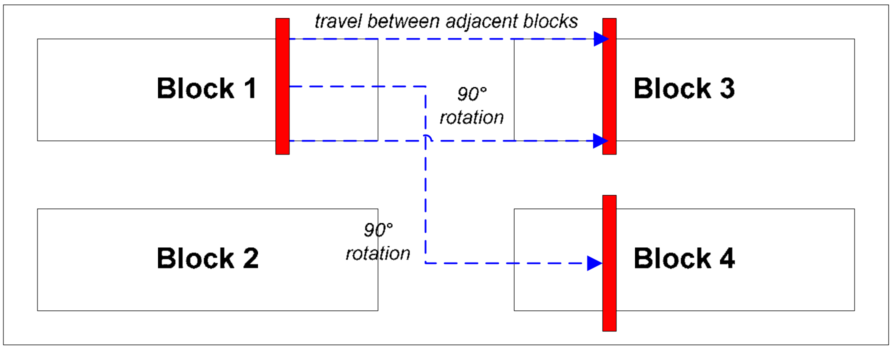


Figure 1.7 - Crane transfer between yard blocks

The point to remark is that, in many terminals the management of yard cranes has been quite experience-based and did not receive a great deal of attention until the last decade. In particular, despite the popularity of yard cranes and the importance of their role in the yard operation, there still has been a limited number of systematic studies on the yard crane deployment problem.

In particular, the RTGC deployment problem has been addressed via mathematical models (see for example Zhang et al. 2002, Cheung et al. 2002, Lim et al. 2002, Linn et al. 2003). However, in most of these formulations, some of the underlying modeling assumptions are necessary to simplify a complex analysis, yet questionable. Practically speaking, they are seldom met in the real world and can be misleading when investigating the performance of different yard crane deployment rules.

To begin with, most of the times data is assumed to be deterministic and static. Still, on-the-field experience bears out that daily workloads vary according to contingent requirements and/or circumstances, among which the calling vessels' ETA - expected time of arrival. Therefore, workloads are often known and revised in just a matter of hours. In such a case, the crane deployment problem should be solved again and a periodic update of the crane deployment schedule should follow immediately.

In the second place, converting the daily workload in terms of crane time is not as straightforward as usually assumed to be. As a matter of fact, the service time of a yard crane is not deterministic and the amount of work done in a container block per time period isn’t typically proportional to the number of cranes operating in the block during that period. As the number of cranes in a block increases, crane productivity may decrease because the space for cranes to maneuver and work becomes more limited. In addition to this, other “irregular phenomena”, such as crane starvation, blocking and/or failure, could and should be taken into account. Any of the previous may occur during the ordinary work cycle for container retrieval/stacking in the yard and cause delay upon yard operations.

In the end, the solutions obtained from stand-alone models are likely to provide only partial guidance when modeling time-evolving systems such as a storage yard in a container terminal.

On the strength of the above considerations, one may acknowledge the usefulness of reproducing via simulation the system dynamics over multiple periods, under some conditions of uncertainty due to randomly occurring events and random duration of logistic activities. In this sense, (Kim et al. 2006) use simulation to address dispatching and task sequencing rules for container yards operated by multiple dual rail-mounted gantry cranes (DRMGCs, where one crane is larger than the other, thus they can pass each other without interfering with one another) in a yard block.

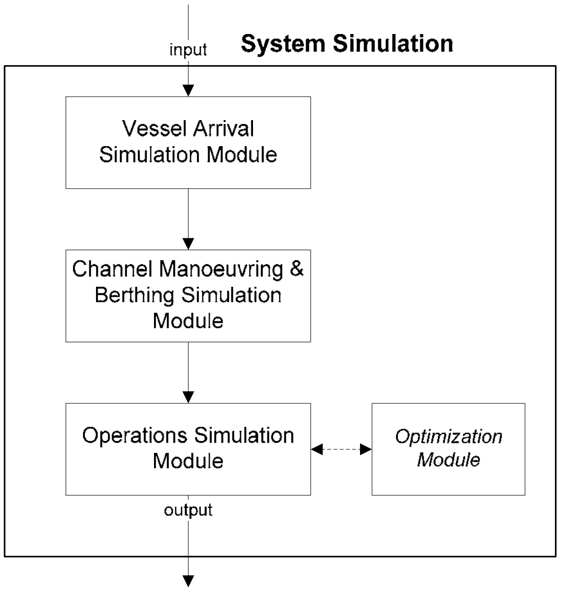

input
System Simulation
Vessel Arrival Simulation Module
Channel Manoeuvring & Berthing Simulation Module
Operations Simulation Module
Optimization Module
output

Figure 1.8 - The architecture of the integrated crane deployment framework

Instead, (Canonaco et al. 2009) extend a similar approach to the RTGC problem by introducing the integrated framework illustrated in Figure 1.8. The proposed simulation-based architecture includes an optimization model to determine the block pairs between which cranes should be transferred during the period under examination, as shown by the queuing network in Figure 1.9, in order to satisfy the crane capacity requirements and minimize the total cost for block matching and crane activation.

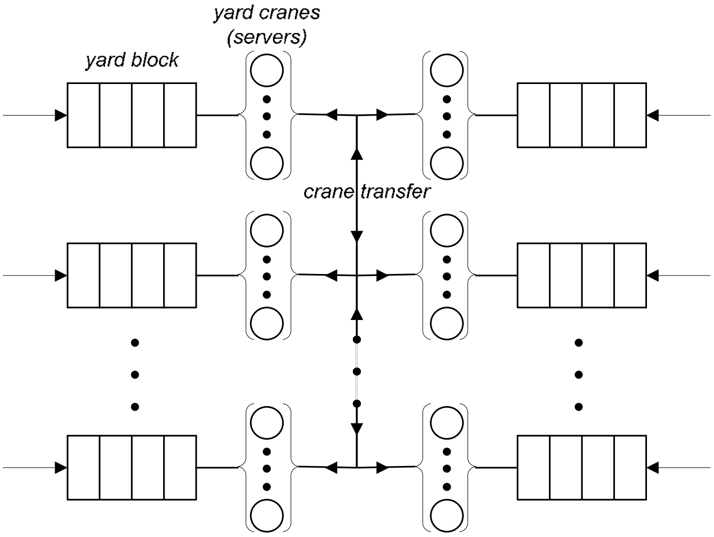


Figure 1.9 - Crane transfer between yard blocks

In conclusion, for a third time, the need for an all-inclusive simulation and optimization approach has sprung from bottom-up requirements in the operation of container terminals.

1.5 Simulation-based optimization

In the attempt to further address the problems described in the previous sections, one may start by recognizing that terminal containers, as well as many other modern day systems providing products and services in the fields of logistics, manufacturing, transportation, network-centric computing, etc., are event-driven and, thus, can be modeled as discrete-event systems (DES) with the objective of carrying-out performance analysis and optimization. As a result of this standing, the growing importance of discrete stochastic optimization is easily understood especially in the design and operation of the above systems.

A general problem of discrete stochastic optimization can be defined as follows:

$$\min_{\theta\in\Theta} E\left[f\left(\theta\right)\right] \qquad (1.1)$$

where

Θ is the solution space;

θ is the set of controllable (input) design variables;

$f\left(\theta\right)$ is the performance measure of interest;

$E\left[f\left(\theta\right)\right]$ is the mathematical expectation of the performance measure of interest.

The goal is to minimize (maximize) the objective function $E\left[f\left(\theta\right)\right]$ over all possible combinations of the controllable design variables θ. In many contexts $f\left(\theta\right)$ is a random variable subject to variance and, thus, the *actual* value of $f\left(\theta\right)$ cannot be optimized. This is straightforward if one considers the operational activities in a container terminal: vessel arrival, container movements, equipment failure, congestion phenomena, weather conditions and so on. The random nature and variety of the activities governed by uncertainty leads to the definition of a class of problems in which whatever expected performance measure cannot be determined analytically, but must be estimated on sample paths generated via discrete-event simulation.

With respect to these issues, the ultimate solution effort is known as *simulation-based optimization* (or *optimum-seeking by simulation* or *simulation optimization*). Formally, simulation-based optimization (SO) means searching for the settings of controllable decision variables that yield the minimum (maximum) expected performance of a stochastic system that is represented by a simulation model (Fu and Nelson 2003).

In a simulation-based optimization procedure, a structured iterative approach calls an optimization algorithm to decide how to change the values for the set of input parameters (i.e. configuration) and then uses the responses generated by the simulation runs to guide the selection of the next set. The logic of this approach is shown in Figure 1.10.

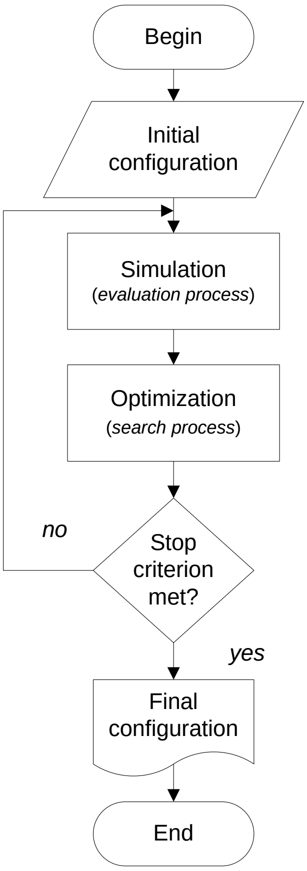


Figure 1.10 - Logic of a simulation-based optimization approach

In particular, the simulation process cannot return an estimate of the objective function by simply fitting a set of possible decision variables into a simple closed-form formula: input variables may be either quantitative or qualitative. Furthermore, the computational expense of a single replication of the simulation model of interest is likely to exceed the typical computation time required by any medium-sized (thousands of variables) linear program.

Thus, the trade-off between the amount of computational time needed to find improved configurations on the optimization side (*search process*) versus the effort in estimating via simulation the performance at a particular configuration (*evaluation process*) becomes a key issue and some practical "compromises" need to be made.

According to (Banks et al. 2001), the main simulation-based optimization approaches must:

- guarantee a pre-specified probability of correct selection since performance evaluation is based on observations that are random variates returned from a simulation process and, thus, do not guarantee the selection of the best design among competing alternatives, despite it being truly representative of the best system configuration;
- guarantee asymptotic convergence to the (global/local) optimal solution in finite time as it must be in practice;
- guarantee optimality for deterministic counterpart which, practically speaking, consists in verifying that the SO algorithm would find the optimal solution if the performance of each design could be evaluated with certainty;
- be based on robust heurisitcs, meaning that heuristics should be effective with limited or no dependence on problem structure and/or variable types, as well as tolerant to some sampling variability (e.g. simulated annealing, genetic algorithms, tabu search).

(Fu 2001) divides the types of simulation-based optimization techniques in the following main categories:

- statistical procedures (e.g. ranking & selection procedures and multiple comparison for the comparison of two or more alternative system configurations);
- metaheuristics (methods directly adopted from deterministic optimization search strategies such as simulated annealing);
- stochastic optimization (random search, stochastic optimization);
- other, including ordinal optimization and sample path optimization.

The present thesis focuses on procedures included in the first category (i.e. ranking & selection in chapter 2) to estimate the best among a set of alternatives whether they are all known in advance or actually generated during simulation run. To favor the management of the latter case, a metaheuristic approach is used (i.e. simulated annealing algorithm in chapter 3) to amend for solution generation when having to decide what alternative system configurations to simulate. As a whole, the simulation-based optimization models proposed are then integrated and applied to address yard and infrastructures organization in a real container terminal (in chapter 4) with the goal of selecting the “best” alternative overall system configuration for greater yard utilization and productivity.

Chapter 2

Selecting the best system

2.1 Introduction

A major objective of a simulation study and that which has been dominant in Management Science and Operations Research over the history is system analysis, where the intent is to mimic behavior to understand or improve system performance (Nance and Sargent 2002). When simulating competing system designs and/or alternative policies of system management, whether these alternative configurations exist or are just envisioned, the singular overriding objective of simulation becomes the detection of the solution that returns the best performance (i.e. the best value of a selected performance metric). Solution “comparison” is actually based on statistical estimates of the average performance measure of interest. These estimates are computed from a certain number of direct observations of the simulation process, that should be properly rearranged to minimize the effect of autocorrelation. Since the above statistical estimates are, in turn, random variates from unknown distributions with a non negligible variance, returned from one or multiple simulation runs, there are no guarantees of selecting the best design during the solution comparison, despite it being truly representative of the best system configuration.

To this end, in literature *homogeneity tests* (Milton and Arnold 1986) are conventionally applied to assess whether there are statistically significant differences in various populations (observation samples) with respect to some characteristics. However, they provide no information in the prospect of selecting the “best” of these populations, once the null hypothesis of the homogeneity test has been rejected. Bearing in mind this expectation, *Ranking and Selection* techniques are the next step to take when searching for a decision procedure that allows to perform a correct selection at a pre-assigned level of probability.

In this chapter the major *Ranking and Selection* approaches are presented for cases in which all alternative simulated system designs are known in advance. Two newly proposed *Ranking and Selection* procedures are also illustrated, followed by numerical experiments meant to compare their performances with those of the previously discussed classic *Ranking and Selection* techniques.

2.2 Ranking and Selection

In (Goldsman et al. 2002), *Ranking and Selection* (R&S) is defined as a natural statistical technique used to identify the best among a set of k competing designs, policies or system configurations. This method is applicable when system parameters (e.g. allocated resources, scheduling policies, etc.) are discrete and the number of competing designs is both discrete and small (e.g. $k = 2,...,20$). The method is applied once a sample mean for a measure of system performance has been constructed from simulated or real data. At the basis of the method there is the evaluation of the sample variance associated to each sample mean to be compared. The smaller the sample variance is, the more one is confident that a sample mean is better (smaller or higher) than another and, therefore, that the related system is to be preferred. As usual in classical statistics, the normality assumption of the sample mean is of great help in determining the confidence level of the selection process.

Most of the research work in R&S can be classified into the following general approaches:

- *subset selection procedures*, which aim at producing a subset of (small) random size that contains the best system, with a user-specified probability;
- *indifference-zone procedures*, where either the best or whatever solution evaluated within a fixed distance from the best can be selected, with a user-specified probability.

When operating a selection of the best system or a subset of the best among a set of simulated competing alternatives, using an R&S technique rather than another depends on which of the available procedures will most benefit a given objective or constraint set by the experimenter. An “educated” choice of an R&S procedure also requires a good knowledge of the structure of the feasible solutions space in view of the fact that the said structure impacts on the performance of the procedures that can be used for problem solving.

Everything considered, the performance level of an R&S procedure can be affected by:

- the probability of selecting the alternative which is truly representative of the best system configuration (PCS – *probability of correct selection*);
- the above probability returned within a given predetermined time budget;
- the existence of extreme configurations in which, for example, all solutions have an equal mean value (i.e. the *equal-mean configuration*) or every solution is distant exactly delta units from the best (i.e. the *least favorable configuration*) or ordered solutions are equally spaced from one another (i.e the *equally-spaced configuration*);
- the difference between solutions which is assumed to be statistically insignificant;
- the structure of the feasible solutions space.

This stated, it is quite logical that different problems require different approaches. For example, in complex systems one of the following situations might occur: *i*) all the possible alternative system configurations are known before experimentation or *ii*) system configurations are revealed (meaning generated) during experimentation. Obviously, these cases also call for the use of specific (meaning different) procedures.

In the following, the selection of the best system(s) is performed according to a user-defined probability under a pre-assigned time budget, whenever the solutions are all known *a priori* (see Bechhofer et al. 1995 for a complete summary). Both simple and combined R&S procedures that belong to the *subset-selection* and *indifference-zone* approaches are examined. For most cases, numerical experiments based on real systems and real data are conducted in the attempt to justify research efforts in searching for "intelligent" sample allocation when solving well-structured problems with significant constraints, especially within large, real-sized contexts. Practically speaking, avoiding over-sampling can affect the termination of the selection procedures and, thus, results can be obtained with a less amount of simulation (i.e. execution time).

As far as notation is concerned, let:

k the number of alternative simulated system designs ($i = 1..k$);

n the number of observations ($j = 1..n$) sampled from each system design;

$\mu_1, \mu_2, ..., \mu_k$ the unknown k expected values of the performance measure of interest;

$\mu_{[k]} \geq ... \geq \mu_{[1]}$ the ordered unknown k expected values of the performance measure of interest (i.e. the system design in position k is the greatest);

X_{ij} the j^{th} observation taken from system design i;

$\overline{X}_k, ..., \overline{X}_1$ the sample means of the performance measure of interest for each system design;

σ^2 the common (unknown) variance of the alternative system designs;

$\bar{S}_k^2,...,\bar{S}_1^2$ the sample variance of the performance measure of interest for each system design;

$1-\alpha$ the confidence level (or user-specified probability P^*)

It is worth observing that the basic underlying assumptions for all the R&S procedures considered herein, meaning independent and identically distributed normal data with common variance, usually depart from the realistic settings involved when simulating real-world systems. However, some important statistical results allow to extend the application of these simulation-based methods to problems in which simulation output data is not independent, nor normally distributed. These issues range from performing the proper process initialization (Law and Kelton 2000) to finding a consistent estimator for the sample variance (Meketon and Schmeiser 1984; Goldsman et al. 1990; Glynn and Whitt 1991; Damerdji 1994; Song and Schmeiser 1995; Steiger and Wilson 2002).

2.3 Subset-selection procedures

Rather than claiming that one population is the best, perhaps it is more convenient to claim that one is confident that the best population is contained in a subset I of the $\{1,2,...,k\}$ competing simulated systems. Subset selection procedures are based on this logic. These R&S procedures aim at producing a subset of (small) random size that contains the best system, with a user-specified probability.

This R&S approach was first introduced by (Gupta 1965) with the purpose of obtaining a subset $I \subseteq \{1,2,...,k\}$ according to which

$$P\{k \in I\} \geq 1-\alpha. \qquad (2.1)$$

Basically, Gupta’s idea was to include in I all the systems k that fall in the following interval:

$$\left[\overline{X}_k(n) - h\sigma\sqrt{\frac{2}{n}}, \quad \overline{X}_k(n)\right] \tag{2.2}$$

where σ is the common, known standard deviation and $\overline{X}_k(n)$ is the maximum among the sample means. Obviously, the most favorable case would be $|I| = 1$.

In order to guarantee (2.1), the value of h in (2.2) is determined as follows:

$$P\{k \in I\} =$$

$$= P\left\{\overline{X}_k(n) \geq \overline{X}_i(n) - h\sigma\sqrt{\frac{2}{n}}, \quad \forall i \neq k\right\} \tag{2.3}$$

$$= P\left\{\frac{\overline{X}_i(n) - \overline{X}_k(n) - (\mu_i - \mu_k)}{\sigma\sqrt{2/n}} \leq h - \frac{(\mu_i - \mu_k)}{\sigma\sqrt{2/n}}, \forall i \neq k\right\}. \tag{2.4}$$

If μ_k is the unknown performance measure of the "best" system, then $-\frac{(\mu_i - \mu_k)}{\sigma\sqrt{2/n}}$ is a positive value, thus

$$P\{k \in I\} \geq P\left\{\frac{\overline{X}_i(n) - \overline{X}_k(n) - (\mu_i - \mu_k)}{\sigma\sqrt{2/n}} \leq h, \forall i \neq k\right\} \tag{2.5}$$

and finally

$$P\{k \in I\} \geq P\{Z_i \leq h, \quad i = 1,2,...,k-1\} = 1 - \alpha \tag{2.6}$$

where $(Z_1, Z_2, ..., Z_{k-1})$ are distributed according to a multivariate normal distribution with means equal to 0, variances equal to 1 and common pair-wise correlation equal to $1/2$. In order to guarantee (2.1), h must be the $1-\alpha$ quantile of the maximum value of $(Z_1, Z_2, ..., Z_{k-1})$.

The following pseudo-code provides a high-level description of Gupta's approach for a maximization problem:

Algorithm 2.1: Gupta's subset-selection procedure

```
$k$, $1-\alpha$, $n$, $h$ ← select procedure settings
for $i = 1$ to $k$ do
    for $j = 1$ to $n$ do
        $X_{ij}$ ← take a random sample of $n$ from each of the $k$ systems
    end for
    $\bar{X}_i$ ← compute an estimate of the performance index of interest for system $i$
    if $\bar{X}_i \in \lfloor \bar{X}_k(n) - h\sigma\sqrt{2/n}, \quad \bar{X}_k(n) \rfloor$ then
        $I \subseteq \{...i...\}$ ← include system $i$ in subset $I$
end for
```

In the above procedure, the choice of $1-\alpha$ is left to the experimenter. Practically, $1-\alpha$ should be greater than or equal to 0.5, since any system could be included in I by simply tossing a fair coin. At the same time, $1-\alpha$ should also be greater than or equal to $1/k$ which is the probability of randomly selecting a system for inclusion in the subset. A pure empirical rule (Gibbons et al. 1979) recommends $1-\alpha \geq 0.5 + (0.5/k)$.

Whereas Gupta's procedure requires balanced (normal) data ($n_1 = n_2 = ... = n_k = n$) with common known variance, in (Nelson et al. 2001) a more general case is proposed that allows to deal with unknown and (perhaps) unequal variances, $\sigma_i^2 \;\; i = 1,...,k$, across all systems. To guarantee that the interval I contains a system i, this alternative approach requires that the difference between the best and the second best is at least δ. In particular, if δ is set equal to 0, this case becomes a generalization of Gupta's procedure allowing unequal variances.

Algorithm 2.2: General subset-selection procedure

```
$k$, $1-\alpha$, $n$, $t_{1-(1-\alpha)^{\frac{1}{k-1}},n-1}$ ← select procedure settings
for $i = 1$ to $k$ do
    for $j = 1$ to $n$ do
```

```
        $X_{ij}$ ← take a random sample of $n$ from each of the $k$ systems
    **end for**
    $\overline{X}_i$ ← compute the sample mean estimate of the performance index of interest for system $i$
    $S_i^2$ ← compute the sample variance estimate of the performance index of interest for system $i$
**end for**
$W_{ij} = t \cdot \left( \frac{S_i^2}{n} + \frac{S_j^2}{n} \right)^{1/2}$ ← compute this amount for all $i \neq j$
**if** $\overline{X}_i \geq \overline{X}_j(n) - W_{ij} \;\; \forall j \neq i$ **then**
    $I \subseteq \{\ldots i \ldots\}$ ← include system $i$ in subset $I$
```

As one may observe, W_{ij} decreases as the size of the random sample taken from each system (n) increases and as the $1-\left(1-\alpha\right)^{1/k-1}$ quantile of Student's t law with $n-1$ degrees of freedom decreases.

2.3.1 An application in logistics

In this example, a primitive *Gupta-like* subset-selection procedure is used to select the best yard crane assignment and transfer policy in a terminal container. In particular, once container stacking/retrieval operations are completed in a yard block, crane transfer to another block in need can occur in agreement with different suitable policies. Since the site currently under investigation does not feature at the moment similar equipment, nor related organizational solutions, to fix ideas one may consider the following five alternative options (or system configurations) derived from resource allocation policies in computer science:

- crane transfer to the yard block with shortest workload (policy 1);
- crane transfer to the yard block with greatest workload (policy 2);
- crane transfer to the closest yard block (policy 3);
- crane transfer to the yard block with greatest priority (policy 4);
- crane transfer to a random yard block (policy 5).

[Observe that results obtained when applying the “crane transfer to a random yard block” option are conceived as a lower bound reference for comparison with the results of the other policies.]

The objective of this study is to select the policy which allows to minimize the maximum average time to complete stacking/retrieval operations of suitable batches of containers (BCT) in the yard blocks (YB).

The simulation model and experiments have been verified and validated in compliance with the classes of techniques conventionally used for this purpose in (Banks et al. 2001). They have also been carefully designed and conducted within *Rockwell’s Arena* (2000), one of the major general-purpose discrete-event simulation packages available on the market. The two Arena diagram flows (Figures 2.1 and 2.2) used to model the core logistic processes at the heart of all the yard crane activity are composed of standard and user-defined blocks which allow to account for non-customary policies and procedures.

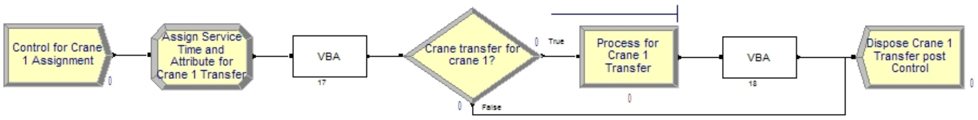

Control for Crane 1 Assignment
0
Assign Service Time and Attribute for Crane 1 Transfer
VBA
17
Crane transfer for crane 1?
0
True
0
False
Process for Crane 1 Transfer
0
VBA
18
Dispose Crane 1 Transfer post Control
0

Figure 2.1 - A diagram flow for the crane monitoring process in Arena

One periodically monitors the status of an available crane (e.g. Crane 1) and schedules its future use according to the idle, busy or in-transfer condition of the resource. The other generates multiple job requests consisting in batches of containers that require retrieval if scheduled for departure from the yard and/or stacking if planned for storage on the yard.

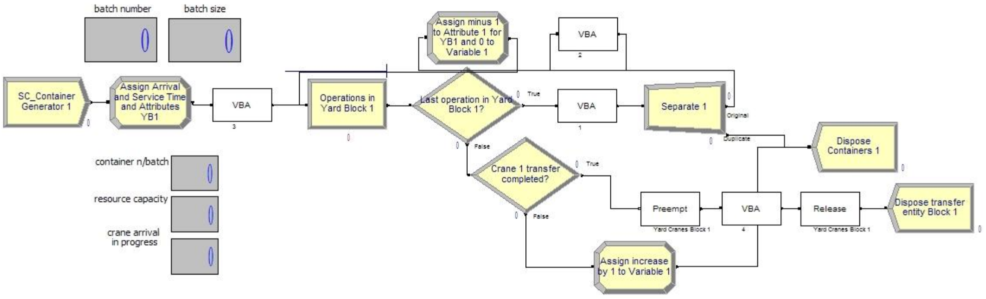
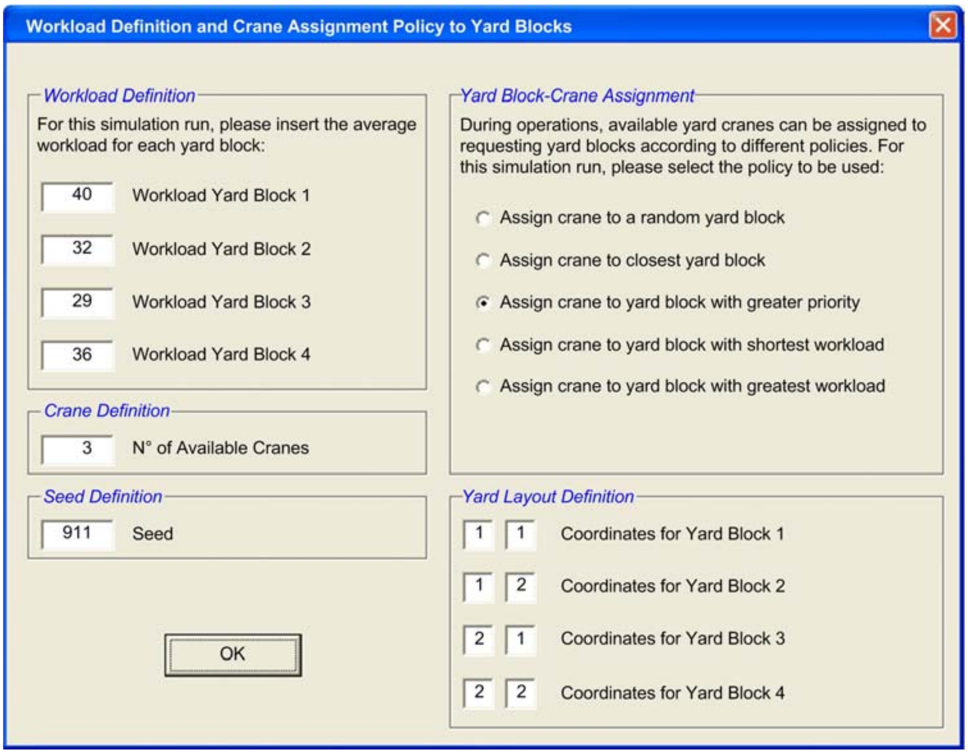

batch number
0
batch size
0
SC_Container Generator 1
0
Assign Arrival and Service Time and Attributes YB1
VBA
3
Operations in Yard Block 1
0
Assign minus 1 to Attribute 1 for YB1 and 0 to Variable 1
VBA
2
Last operation in Yard Block 1?
0
True
0
False
VBA
1
Separate 1
0
Original
0
Duplicate
Dispose Containers 1
0
container n/batch
0
resource capacity
0
crane arrival in progress
0
Crane 1 transfer completed?
0
True
0
False
Preempt
Yard Cranes Block 1
VBA
4
Release
Yard Cranes Block 1
Dispose transfer entity Block 1
0
Assign increase by 1 to Variable 1

Figure 2.2 - A diagram flow for container generation and retrieval/stacking in Arena

During numerical experimentation performed on a personal computer equipped with a 1.73 Ghz Intel Pentium M processor and 1 Gb of RAM, in order to favour an easier and less time-consuming data input and output for every scenario, separate VBA (Visual Basic® for Applications) windows have been used to obtain an integrated interaction with Arena and Microsoft® Excel 2002. For instance, a great number of experiments have been rapidly performed (approximately 2 seconds per run) by limiting data entry to the fields portrayed in Figures 2.3 and 2.4 and listed below:

- average workload per yard block;
- number of available RTGCs;
- seed (i.e. a value used to initialize the random-number generator in order to obtain the unique sequences of pseudorandom numbers behind each stochastic simulation run);
- policy for RTGC assignment to yard blocks and, if applicable, priority specification;
- yard block coordinates (to define yard layout).


Workload Definition and Crane Assignment Policy to Yard Blocks

Workload Definition

For this simulation run, please insert the average workload for each yard block:

40 Workload Yard Block 1

32 Workload Yard Block 2

29 Workload Yard Block 3

36 Workload Yard Block 4

Crane Definition

3 N° of Available Cranes

Seed Definition

911 Seed

Yard Block-Crane Assignment

During operations, available yard cranes can be assigned to requesting yard blocks according to different policies. For this simulation run, please select the policy to be used:

○ Assign crane to a random yard block

○ Assign crane to closest yard block

● Assign crane to yard block with greater priority

○ Assign crane to yard block with shortest workload

○ Assign crane to yard block with greatest workload

Yard Layout Definition

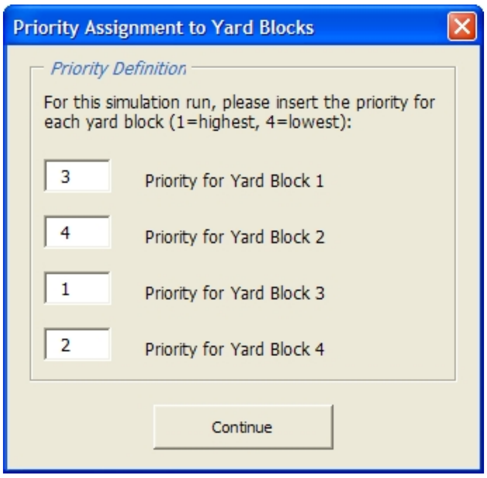
1 1 Coordinates for Yard Block 1

1 2 Coordinates for Yard Block 2

2 1 Coordinates for Yard Block 3

2 2 Coordinates for Yard Block 4

Figure 2.3 - The VB window for general data input



For this simulation run, please insert the priority for each yard block (1=highest, 4=lowest):

3 Priority for Yard Block 1

4 Priority for Yard Block 2

1 Priority for Yard Block 3

2 Priority for Yard Block 4

Figure 2.4 - The VB window for priority specification among yard blocks

No transient suppression schemes have been used during output analysis based on the following grounds. First of all, given a certain yard definition (i.e. number and position of yard blocks), terminating simulations have been performed in order to minimize the average maximum time to complete a container batch processing under different crane transfer policies.

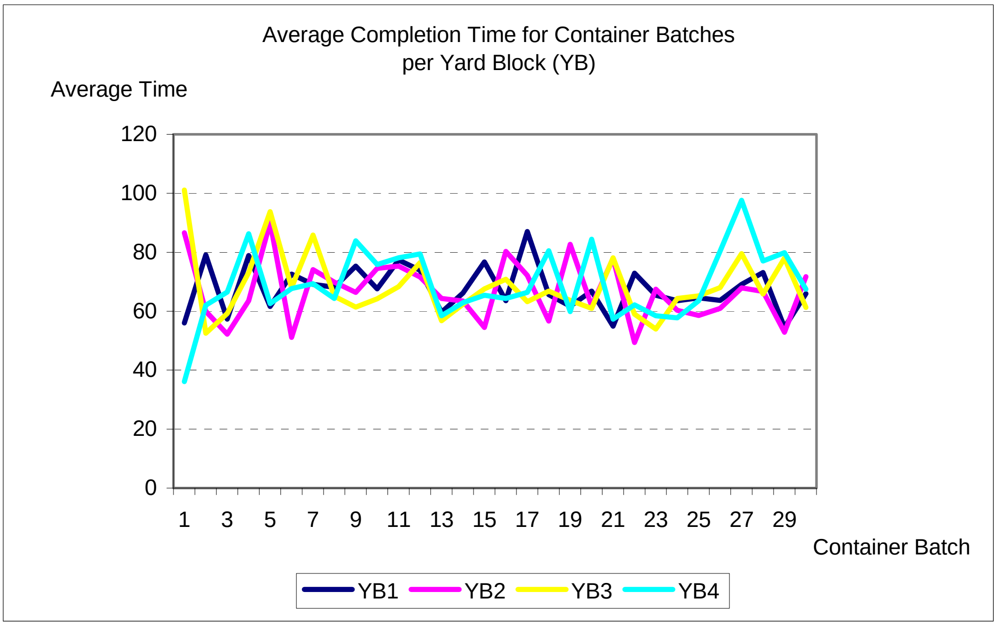


Figure 2.5 - Example of the nonexistence of the warm-up period

Therefore, there has been no need to evaluate steady-state parameters. Secondly, the “requests” to stack/retrieve a batch of containers have been assumed to occur according to a renewal process, as defined in (Heyman and Sobel 1982). The nonexistence of the warm-up period for the performance index of interest is illustrated in Figure 2.5.

To illustrate the subset-selection procedure under examination, the specific scenario proposed features medium container traffic intensity and high crane transfer time among yard blocks (thus, policy n°3 is expected to be best performer). All experiments are carried out by setting $P^* = 0.90$, $k = 5$ and $n = 10$, under the realistic assumption of unknown, but common variance for each system design. To this end, Bartlett’s test (1937) has been used to verify the common variance assumption.

The application of the subset-selection procedure (SSP) described by Algorithm 2.2 requires the definition of the proper quantile of Student’s t law $t_{1-0.90^{1/4},9} \cong 2.66$ and the quantity $W_{ij} = t \cdot \left(\frac{S_i^2 + S_j^2}{n} \right)$ for all $i \neq j$. The latter is computed on the estimates provided in Table 2.1 used to compute both the sample mean and the sample variance for each system i.

Table 2.1 – Sample means and sample variance for the five alternative yard policies

estimate/ i	1	2	3	4	5
N=1	90.71	85.91	75.98	100.01	82.17
N=2	87.79	77.19	70.73	92.16	74.42
N=3	106.38	95.52	80.08	112.71	100.34
N=4	82.47	81.59	73.90	94.93	78.40
N=5	99.79	95.29	79.64	93.78	97.92
N=6	94.73	97.60	86.34	86.08	101.38
N=7	97.95	89.52	77.63	109.96	91.35
N=8	111.83	103.78	80.38	111.15	91.83
N=9	101.41	87.56	82.92	102.60	97.05
n=10	100.64	96.47	74.17	97.13	108.57
$\overline{X}_i$	97.37	91.04	78.18	100.05	92.34
S_i^2	76.49279	66.099	21.6283	79.95434	120.3002

A matrix representation of the W_{ij} s given by the above operations is provided in the following.

$$\mathbf{W} = \left[W_{ij}\right] = \begin{bmatrix} - & 10.05 & 8.34 & 10.53 & 11.81 \\ 10.05 & - & 7.88 & 10.17 & 11.49 \\ 8.34 & 7.88 & - & 8.48 & 10.03 \\ 10.53 & 10.17 & 8.48 & - & 11.91 \\ 11.81 & 11.49 & 10.03 & 11.91 & - \end{bmatrix}$$

In this particular (minimization) problem, the system i for which $\overline{X}_i \leq \overline{X}_j + W_{ij}$ for all $j \neq i$ is only one: $I = \{3\}$.

This result has also been compared with those returned by another two R&S procedures that will be examined in the next section: Rinott's procedure (RP) (1978) and the OP procedure proposed in (Canonaco et al. 2009).

Table 2.2 - Simulation results for the five alternative yard policies

Policy *i*	**N° of Observations**			**Average BCT (minutes)**
	SSP	**RP**	**OP**	
Policy 1	10	31	10	97.369
Policy 2	10	27	17	91.043
Policy 3	**10**	**10**	**10**	**78.177**
Policy 4	10	32	26	100.052
Policy 5	10	48	17	92.343

As one may easily calculate from Table 2.2, a cumulative number (over all policies) of 148 and 80 observations are required by these two procedures, respectively, whereas the SSP accomplishes the same result with only 50 observations. The case study just presented is representative of a typical situation where system configurations are well-spaced from each other, with respect to the performance metric adopted for comparison. Here one may recognize that the SSP allows to screen suitable configurations with a very limited number of observations.

2.4 Indifference-zone approach

Similar to any other selection procedure dealing with random variates returned from a simulation process, the indifference-zone (IZ) based approach may or may not select the simulated system configuration which is truly representative of the best solution (if it does, then a correct selection (CS) is said to have been made). The novelty lies in the fact that this selection approach is statistically indifferent to which system configuration is chosen among a set of competing alternatives when these alternatives all fall within a fixed distance from the best solution.

This stated, let $P\{CS\}$ be the probability of correct selection and δ the indifference-zone chosen by the experimenter. In a maximization problem the probability of performing a correct selection with at least probability P^* is

$$P\{CS\} \hat{=} P\{\mu_k > \mu_i \forall i \neq k \mid \mu_k - \mu_i \geq \delta\} \geq P^*. \tag{2.7}$$

The probability of correct selection (2.7) was first computed in (Rinott 1978) by resorting to numerical integration under the hypothesis of normality of the statistics involved. If $P(CS)$ is the probability that $\overline{X}_k(n_k)$ is the true "best" sample mean, namely it corresponds to $\overline{X}_{[k]}(n_k)$, then

$$P(CS) =$$

$$= P\left[\overline{X}_k(n_k) = \overline{X}_{[k]}(n_k)\right] \tag{2.8}$$

$$= P\left[\overline{X}_k(n_k) > \overline{X}_{k-1}(n_{k-1})\right], \text{ for short } P\left[\overline{X}_k > \overline{X}_{k-1}\right] \tag{2.9}$$

$$= P\left[\frac{\overline{X}_k - \mu_k}{\delta/h} > \frac{\overline{X}_{k-1} - \mu_{[k-1]}}{\delta/h} + \frac{\mu_{[k-1]} - \mu_{[k]}}{\delta/h}\right]. \tag{2.10}$$

Since $\left[\dfrac{\overline{X}_k - \mu_{[k]}}{\delta/h}\right] \hat{=} T_k$ and $\left[\dfrac{\overline{X}_{k-1} - \mu_{[k-1]}}{\delta/h}\right] \hat{=} T_{k-1}$ are distributed according to Student's law with $n_k = n_{k-1} = \cdots = n_0$ degrees of freedom (Law and Kelton

2000) and since $\overline{X}_k$ and $\overline{X}_{k-1}$ are assumed to follow a Normal distribution, then

$$P(CS) =$$

$$= P\left[T_k - T_{k-1} > \frac{\mu_{[k-1]} - \mu_{[k]}}{\delta/h}\right] \qquad (2.11)$$

$$= P\left[T_{k-1} < T_k + \frac{\mu_{[k]} - \mu_{[k-1]}}{\delta/h}\right]. \qquad (2.12)$$

According to the total probability distribution conditioned on T_k,

$$P[T_{k-1} < T_k] =$$

$$= \int_{t=0}^{\infty} P[T_{k-1} \le t \mid t < T_k \le t + dt] * P[t < T_k \le t + dt] \qquad (12.3)$$

$$= \int_{t=0}^{\infty} F_{T_{k-1}|T_k}(t) f_{T_k}^{(m)}(t) dt. \qquad (2.14)$$

Because of independence between T_k and T_{k-1} then

$$= \int_{t=0}^{\infty} F_{T_{k-1}}(t) f_{T_k}(t) dt. \qquad (2.15)$$

In the particular case under examination (maximization)

$$P\left[T_{k-1} < T_k + \frac{\mu_{[k]} - \mu_{[k-1]}}{\delta/h}\right] = \int_{t=0}^{\infty} F_{T_{k-1}}\left(t + \frac{\mu_{[k]} - \mu_{[k-1]}}{\delta/h}\right) f_{T_k}(t) dt. \qquad (2.16)$$

Since $\mu_{[k]} - \mu_{[k-1]} \ge \delta$, the final result is

$$P(CS) \ge \int_{t=0}^{\infty} F_{T_{k-1}}(t + h) f_{T_k}(t) dt. \qquad (2.17)$$

Note that in (2.17) equality is verified when $\mu_{[k]} - \mu_{[k-1]} = \delta$. If the integral is set equal to P^* and solved numerically for h, for different values of n (the number of observations taken from the system to compute the sample mean), the results can be tabled and read to obtain h, which is also known as Rinott's constant. Numerical values for h are tabled in (Wilcox 1984).

2.4.1 Indifference-zone procedures

In IZ-based Ranking and Selection, making a correct selection with at least probability P^* can be successfully pursued in different ways. In (Bechhofer 1954) a single-stage procedure is proposed according to the following steps:

```
Algorithm 2.3: Bechhofer's single-stage IZ procedure
$c_{k,P^*}$, $\delta$ ← select procedure settings
$N = \left| \left( \sigma \cdot c_{k,P^*} / \delta \right)^2 \right|$ ← determine the sample size to take from each system
**for** $i = 1$ **to** $k$ **do**
    **for** $j = 1$ **to** $N$ **do**
        $X_{ij}$ ← take a random sample of $N$ from each of the $k$ systems
    **end for**
    $\overline{X}_i$ ← compute an estimate of the performance index of interest for system $i$
**end for**
$\max_i \overline{X}_i$ ← select system with greatest sample mean as best
```

where c_{k,P^*} can be taken from (Bechhofer et al. 1995).

As one may observe, deriving this result is straightforward, but, on the other hand, a single-stage procedure does not enable system feedback aimed at an adaptive control. To this end, two-stage solution algorithms have been introduced. The following pseudo-code provides a high-level description of a two-stage indifference-zone procedure:

```
Algorithm 2.4: General two-stage IZ procedure
Stage 1
```

```
1: $1-\alpha$, $\delta$, $n_0$, $h$ ← select procedure settings
2: for $i = 1$ to $k$ do
3:     for $j = 1$ to $n_0$ do
4:         $X_{ij}$ ← take a random sample of $n_0$ from each of the $k$ systems
5:     end for
6:     $\overline{X}_i$ ← compute an estimate of the sample mean of the performance
       index of interest for system $i$
7:     $S_i^2$ ← compute an estimate of the sample variance of the
       performance index of interest for system $i$
8: end for
9: $N_i = \max\left(n_0, h^2 S_i^2 / \delta^2\right)$ ← determine the sample size to take from each
   system
Stage 2
10: if $n_0 \geq \max_i N_i$ then
11:     $\max_i \overline{X}_i$ ← select system with greatest sample mean as best and
        stop
12: else
13:     for $i = 1$ to $k$ do
14:         for $j = n_0 + 1$ to $N_i$ do
15:             $X_{ij}$ ← take an additional random sample of $N_i - n_0$ from each
                of the $k$ systems
16:         end for
17:         $\overline{X}_i$ ← compute an estimate of the sample mean of the
            performance index of interest for system $i$
18:     end for
19: $\max_i \overline{X}_i$ ← select system with greatest sample mean as best
```

The number of observations required to select the best system design with $P\{CS\} \geq P^*$ is a major impact factor on procedure performance. As shown at line n°9 of the general scheme presented above, this amount mostly depends on the sample variance and, thus, on how the sample mean is computed earlier. To this end, different methods use different approaches. In pioneer two-stage R&S procedures, (Rinott 1978) uses a classic sample mean, while in (Dudewicz and Dalal 1975) a weighted sample mean is used during the second stage. In the next section, the moving-average sample mean investigated in (Canonaco et al. 2009) is presented.

2.4.1.1 OP: a new two-stage indifference-zone procedure

This approach was inspired by the graphical technique used by Welch (Law and Kelton 2000) to deal with the *problem of the initial transient* or the *start-up problem*. This procedure builds an “artificial” process with the same mean as the output observations of interest, but with a smaller variance. It then smoothes out the high-frequency random deviations by introducing a moving-average with a moving window of w values. In brief, the underlying purpose of the method is to give an unbiased estimator that has lower variance than other unbiased estimators for all possible values of the system performance measure under examination. When found, a similar estimator allows to choose the best system at a lower computational cost.

Table 2.3 - The moving-average based R&S approach with $w=1$

Simulation output of *b* *m*-sized groups of data							
1	$Y_{1,1}$	$Y_{1,2}$	$Y_{1,3}$	...	$Y_{1,m-2}$	$Y_{1,m-1}$	$Y_{1,m}$
2	$Y_{2,1}$	$Y_{2,2}$	$Y_{2,3}$	...	$Y_{2,m-2}$	$Y_{2,m-1}$	$Y_{2,m}$
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
b	$Y_{b,1}$	$Y_{b,2}$	$Y_{b,3}$	...	$Y_{b,m-2}$	$Y_{b,m-1}$	$Y_{b,m}$

$$\begin{tikzcd}
\overline{Y}_1 \arrow[d] \arrow[dr] & \overline{Y}_2 \arrow[dl] \arrow[d] \arrow[dr] & \overline{Y}_3 \arrow[dl] \arrow[d] & \cdots & \overline{Y}_{m-2} \arrow[dr] & \overline{Y}_{m-1} \arrow[d] & \overline{Y}_m \arrow[dl] \\
\overline{Y}_1(1) & \overline{Y}_2(1) & \overline{Y}_3(1) & & & \overline{Y}_{m-1}(1) &
\end{tikzcd}$$

Practically, n output observations of a performance index are organized into b groups (one per simulation run), each of size m ($m>>1$), and then used to compute an average value of the i-th observation across these groups (see Table 2.3)

$$\overline{Y}_i = \frac{1}{b}\sum_{j=1}^{b} Y_{ji}. \qquad (2.18)$$

The above values $\overline{Y}_1, \overline{Y}_2, ..., \overline{Y}_m$, are then used to define the moving-average $\overline{Y}_i(w)$ with a window length of w as follows:

$$\overline{Y}_i(w) = \frac{\sum_{s=-w}^{w} \overline{Y}_{i+s}}{2w+1} \qquad i = w+1,...,m-w. \qquad (2.19)$$

For example, to compute the moving average $\overline{Y}_2(1)$, one must first compute $\overline{Y}_2$, and then average this result out with the w values to its left and the w values to its right. In this specific case, if $w=1$, then $\overline{Y}_2(1) = (\overline{Y}_1 + \overline{Y}_2 + \overline{Y}_3)/3$ and, in general, $\overline{Y}_i(1) = (\overline{Y}_{i-1} + \overline{Y}_i + \overline{Y}_{i+1})/3$ for $i = 2,...,m-1$.

To conclude, the $\overline{Y}_i(w)$s become the observations that will be used when computing the sample mean and variance for each system, as required by both stages in Algorithm 2.4.

This stated, the OP procedure has been applied and compared with Rinott's (R) and Dudewicz and Dalal's (D&D) on the logistic problem described in §2.3.1 to verify which of the five assignment policies is likely to be the best. The comparison means to provide a measure of how the OP procedure responds in terms of the total number of observations (nobs) to be taken from each alternative configuration in order to obtain the statistics used in selecting the best system design with at least probability P^*. To this end, four different classes of problems have been defined by providing combinations of smoothly-varying container traffic intensity (low, medium, high) on the four yard blocks considered and the time (low, high) required, and therefore the distance to cover depending on the yard size, when transferring an idle crane from one yard block to another bearing insufficient crane capacity. These classes are listed in Table 2.4.

Table 2.4 - Classes of problems tested in the yard simulation

Problems	**Description**
Class 1	Low traffic, high transfer time
Class 2	Medium traffic, high transfer time
Class 3	Medium traffic, low transfer time
Class 4	High traffic, low transfer time

As demonstrated by the results reported in Table 2.5, in the worst case (i.e. for Class 1 and 3 problems) the other procedures have reached the same correct result but with an average 5% reduction in the number of total observations drawn for the two-stage sampling procedure; in the best case (i.e. for Class 2 and 4 problems), these procedures have been outperformed by the OP procedure by an average of 40%.

Table 2.5 - Classes of problems tested in the yard simulation

Problems	**R**	**D&D**	**OP**
Class 1	-5%	-5%	nobs
Class 2	nobs	nobs	-40%
Class 3	-5%	-5%	nobs
Class 4	nobs	nobs	-40%

To fully complete the example, a Class 2 scenario is considered, along with the following specific settings required by the R&S algorithms: $\delta = 5$ minutes, $P^* = 0.90$, $h = 3.317$, $n_0 = 10$. All three procedures select policy n°3 as likely being the best. Observe that, when dealing with cases like these in which the workload and crane transfer time are approximately of the same order, due to the yard size and layout, this particular result is expected to emerge and, therefore, it is also used as a validation sample. However, the OP procedure arrives to this conclusion with only 80 observations, whereas Rinott and Dudewicz and Dalal need more than 150 observations for the same accomplishment as shown in Table 2.6.

Table 2.6 - Results of the R&S procedures for a Class 2 problem

Procedure	**Nobs**	**Max average container-batch completion time** (min)				
		Policy 1	**Policy 2**	**Policy 3**	**Policy 4**	**Policy 5**
R	151	97.21	93.67	**79.76**	101.57	95.52
D&D	151	97.19	93.41	**82.32**	101.01	90.95
OP	80	95.54	91.73	**79.51**	103.00	92.72

2.4.1.2 CP: a new multi-stage combined procedure

More recent and advanced indifference-zone R&S procedures are based on an n-step logic, with $n > 2$. (Kim and Nelson 2001) first take a single observation from the systems still in play and then choose whether or not to cease sampling from the systems that no longer appear to be competitive. This practice is normally referred to as a "sequential selection". A similar approach is proposed in (Chen and Kelton 2005): it takes into account both the sample variances and the differences between sample means when determining the sample sizes.

In the combined procedure (CP) proposed herein, the idea of "efficient" sampling is pursued by basing the number of output observations to be taken from each system on the corresponding variance behavior within a certain number of simulation runs. Thus, for the present enhancement, it is necessary to establish how such variance is to be estimated.

If for system i ($i = i..k$) the n elementary output observations $X_i \hat{=} \{X_{ij}, j = 1..n\}$ returned from a simulation run are independent and normally distributed, one may pursue variance estimation by simply using classical statistics and computing the sample mean

$$\overline{X}_i = \frac{1}{n}\sum_{j=1}^{n} X_{ij}, \tag{2.20}$$

followed by the sample variance which is used as variance estimator

$$VAR[X_i] = S_i^2 = \frac{1}{n-1}\sum_{j=1}^{n}\left(X_{ij} - \overline{X}_i\right)^2. \tag{2.21}$$

Should this not be the case - as customary in a simulation-based study of practically any real-life system - then one must start from the output stochastic process, organize its data and compute the process variance.

For example, for system i let $\{X_1,...,X_j,,...,X_n\}$ be a *weekly dependent* stationary output process with mean μ_X and variance σ_X^2. This process is said to be *weakly dependent* if the lag-j covariance

$$\gamma_j \mathrel{\hat{=}} Cov\lfloor X_i, X_{i+j} \rfloor, \quad j = 0, \pm 1, \pm 2, \dots \tag{2.22}$$

satisfies $\gamma_j \to 0$ as $|j| \to \infty$ (Billinglsey 1995).

If one chooses to organize this data in batches of size k, the sample mean for batch i is given by:

$$\overline{X}_i(k) \mathrel{\hat{=}} \frac{1}{k} \sum_{j=i+1}^{i+k} X_j \tag{2.23}$$

and according to the Central Limit Theorem

$$\overline{X}_i(k) \overset{D}{\to} Z(\mu_X, \sigma^2(k)/k), k >> 1, \ \forall i \tag{2.24}$$

where

$$\sigma^2(k) = \sigma_X^2 + 2\sum_{j=1}^{k-1} \left(1 - \frac{j}{k}\right) \gamma_j. \tag{2.25}$$

Furthermore,

$$\left\{ \overline{X}_1(k),..., \overline{X}_i(k),..., \overline{X}_n(k) \right\} \text{ grow independent as } k \to \infty \tag{2.26}$$

and

$$\lim_{k\to\infty} \sigma^2(k) = \lim_{k\to\infty} k\ Var\left[\overline{X}_i(k)\right] = \sigma^2, \ \forall i. \tag{2.27}$$

By (Hogg and Craig 1978)

$$\frac{S^2_{\overline{\overline{X}}}(n,k)}{\sigma^2/k} \approx \frac{\chi^2_{n-1}}{n-1}. \tag{2.28}$$

Applying the mathematical expectation to (2.28)

$$E\left[\frac{S^2_{\bar{\bar{X}}}(n,k)}{\sigma^2/k}\right] = E\left[\frac{\chi^2_{n-1}}{n-1}\right] = 1 \qquad (2.29)$$

and thus

$$E\left\lfloor k \cdot S^2_{\bar{\bar{X}}}(n,k)\right\rfloor = \sigma^2 \qquad (2.30)$$

where

$$k \cdot S^2_{\bar{\bar{X}}}(n,k) \hat{=} \frac{k}{n-1}\sum_{i=1}^{n}\left(\bar{X}_i(k) - \bar{\bar{X}}(n,k)\right)^2 \qquad (2.31)$$

is the sample variance of the estimator of the process sample mean.

This stated, the combined procedure uses a variance-weighted decisional mechanism based on the variance estimator described above to guide the sampling activity on the number of additional simulation output observations to be taken from each system. Practically speaking, when process variance decreases this multi-stage procedure is expected to terminate faster than classical two-stage R&S algorithms because of its auto-adaptive control. In every other case, the number of iterations during which the sample variance either remains constant or increases is controlled by an upper bound on the number of additional simulation runs to carry-out.

The following pseudo-code provides a high-level description of this new combined approach:

```
Algorithm 2.5: Combined procedure (CP)
$1-\alpha$, $\delta$, $n_0$, $h$, $x$, $UB$ ← select procedure settings
for $i = 1$ to $k$ do
    for $j = 1$ to $n_0$ do
        $X_{ij}$ ← take a random sample of $n_0$ from each of the $k$ systems
    end for
    $\bar{X}_i$ ← compute an estimate of the sample mean of the performance
    index of interest for system $i$
    $S_i^2$ ← compute an estimate of the sample variance of the
    performance index of interest for system $i$
```

```
end for
$N_i = \max\left(n_0, h^2 S_i^2 / \delta^2\right)$ ← determine the sample size to take from each system
if $n_0 \geq \max_i N_i$ then
    $\max_i \overline{X}_i$ ← select system with greatest sample mean as best and stop
else
    for $I = 1$ to $k$ do
        while $N_i \leq UB$ do
            $X_{ij}$ ← take one additional random sample for system $i$
            $\overline{X}_i$ ← compute an estimate of the sample mean of the performance index of interest for system $i$
            $S_i^2$ ← compute a run-weighted estimate of the sample variance of the performance index of interest for system $i$
            $N_i = \max\left(n_0, h^2 S_i^2 / \delta^2\right)$ ← determine the new sample size for system $i$
            if $N_i \leq 10$ or $S_i^2 = \text{constant}$ in last $x$ runs then
                stop sampling for system $i$
        end while
    end for
$\max_i \overline{X}_i$ ← select system with greatest sample mean as best
```

Once again, the logistic problem described in §2.3.1 will be used as test-bed for the evaluation of Algorithm 2.5 in terms of efficient sampling when comparing alternative system configurations with different variance patterns. In particular, with respect to an operational scenario in which container traffic in yard blocks is medium and crane transfer times are high (i.e. Class 2 problems), Figure 2.6 illustrates an example of the variance behavior for the five policies under examination. Observe that for the first three policies variance behavior is stable, meaning that are no significant numerical changes in variance estimation as the sampling procedure progresses. Thus the algorithm continues adding single observations (or batches or simulation runs) as required by the “stable” variance estimate until the upper bound provided by Rinott’s procedure is reached. When the variance pattern is increasing, as for policy n°4, the upper bound is still provided by Rinott’s procedure.

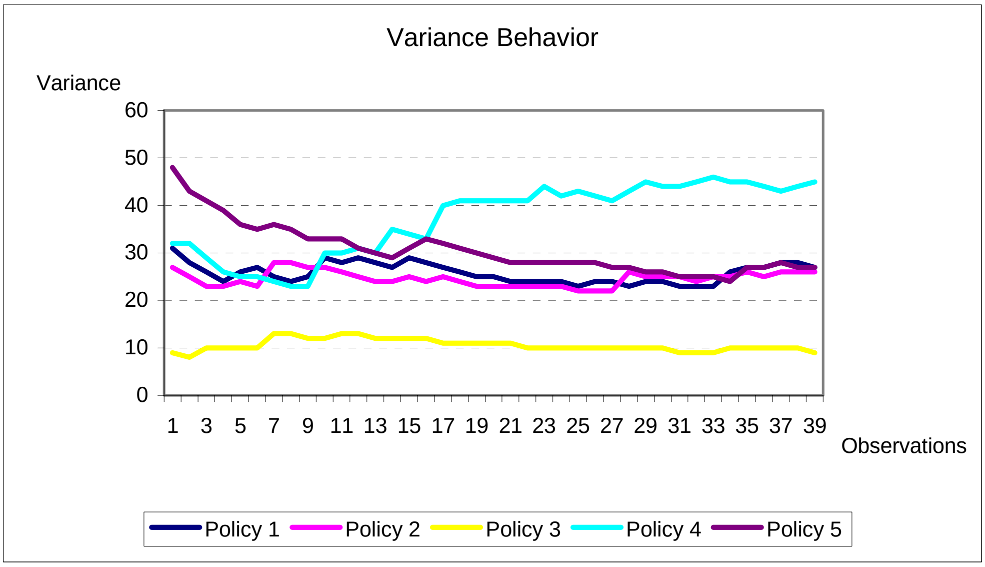


Figure 2.6 - Example of the variance behavior in a Class 2 experiment

Instead, in policy n°5 the variance estimate has a decreasing trend and, thus, the algorithm is expected to terminate faster. This expectation is justified by the auto-adaptive control of the combined procedure which can be monitored according to a step-by-step logic. In this sense, Table 2.7 provides a trace of the variance behavior for policy n°5. As one may observe, Rinott's procedure requires 38 additional observations (i.e. 48 - initial 10 runs) to guarantee the predefined probability of correct selection. Alternatively, the CP procedure after a supplementary run at step 11, requires 32 additional runs (i.e. 43 – 11 previous runs) and, thus, realizes a gain of 6 runs after a single step.

Table 2.7 - Step-by-step trace of variance behavior for policy n°5

Step	**N° of observations for policy i = 5**		**N_i**
	Sample mean	**Sample variance**	
10	92.34	120.30	48
11	92.39	108.30	43
12	92.96	102.34	41
13	92.48	96.85	39
14	92.20	90.49	36
...	...	...	...

In numerical terms, in the worst case the CP procedure returns the same results as Rinott's two-stage procedure (Δ=0%), while for decreasing variance behavior the CP procedure is more efficient by 31,25%, as illustrated in Table 2.8.

Table 2.8 - Comparison of observations required by Rinott's procedure and the combined R&S procedure

Alternatives	**N° of observations**		**CP Performance**
	RP	**CP**	**(Δ%)**
policy 1	31	31	0%
policy 2	27	27	0%
policy 3	9	9	0%
policy 4	32	32	0%
policy 5	48	33	+31.25%

At this point of the methodological framework under examination, some final considerations can be drawn when investigating the performance of indifference-zone based Ranking & Selection procedures. Basically, in this chapter two different types of approaches have been followed to "hopefully" deliver more efficient sampling than classical two-stage algorithms. In the first case, (see the OP procedure in §2.4.1.1) efforts focus on using a moving-average sample mean as a low-variance unbiased estimator, rather than a classical sample mean. In the second case, (see the CP procedure in §2.4.1.2) tracking of the variance behavior reveals the improvement trend when variance pattern is decreasing. A further possibility may lie in investigating how to use an estimate of the skewness of the sample mean distribution, given that the normality assumption is approximately verified only after a large number of simulation runs…a condition one should avoid, due to the computational burden it bears.

Chapter 3
Simulation-based optimization using simulated annealing

3.1 Introduction

The *Ranking and Selection* procedures examined in the previous chapter are based on the common assumption that a (small) number of system configurations are known *a priori*. In this particular case, the guarantee of selecting the best or near-best alternative when all solutions have already been sampled and retained appears to be both very appealing and practicable. However, at times, a combinatorial, unknown number of configurations needs to be explored. When this occurs, k different systems configurations (with $k \geq 1$) can be revealed sequentially during a simulation run by means of a so-called *system generating algorithm* (SGA) (Hong and Nelson 2007). An example of a framework, providing both system design generation and evaluation for selecting the best among a great number of competing alternatives, is summarized by the pseudo-code given below.

Algorithm 3.1: System Generating Algorithm

```
$k$, $n$, stopping condition[0] ← select procedure settings
$i^* = i$ ← set best system design = initial system design
while stopping condition[ n ]= false do
    $n = n + 1$
    $i_1(n), i_2(n), \ldots, i_k(n)$ ← at iteration $n$ generate $k$ alternative system designs
```

```
6:    $i^* = \text{best}\{ i^*, [i_1(n), i_2(n), ..., i_k(n)]\}$ ← compare the $k$ alternative
      system designs generated at iteration $n$ with current best and,
      eventually, update the best
7:    update stopping condition[ n ]
8: end do
9: $i^*$ ← return best system design
```

At this point, it should be easy to observe that the R&S procedures previously examined are suitable for system comparison and selection at line 6 in the above schema, while it is now necessary to focus on the algorithm that can be properly adopted as SGA at line 5. In particular, should an exhaustive coverage of all the possible system combinations be not reasonable, nor affordable from a computational point of view, then metaheuristic-based approaches would have to be addressed. Examples of similar new, promising methodologies use a *simulated annealing* based approach (Alrefaei and Andradóttir 1999, Ahmed and Alkhamis 2002, Prudius and Andradóttir 2005) or an *adaptive balanced explorative and exploitative search* (Prudius and Andradóttir 2004; Prudius 2007). In the next sections an in-depth description of the simulated annealing algorithm and its properties is given, followed by a discussion on the practical limits experienced when customizing this approach to the study of a well-known problem in port logistics. Numerical experiments are presented in the closing paragraph.

3.2 Simulated Annealing

3.2.1 Basic description

The original simulated annealing (SA) algorithm was introduced by developing the similarities between combinatorial optimization problems and statistical mechanics. This analogy was first pointed out by (Kirkpatrick et al. 1983) that show how the Metropolis algorithm (Metropolis et al. 1953) for approximate numerical simulation of the behavior of a many-body system at a finite temperature provides a natural tool for bringing the techniques of statistical mechanics to bear on optimization. In their work, the SA algorithm is applied to a number of problems arising in the optimal design of computers and then used on traveling salesman problems, with as many as several thousand cities, to test its power.

Practically speaking, *the annealing process is aimed to generate feasible solutions, explore them in a more or less restricted amount and, finally, stop when a satisfactory criterion is met.* To avoid getting caught in local minima, during the exploration process a *transition* (or *move*) to a worse feasible neighboring solution (higher-energy state) can occur with probability $p = e^{-\Delta/T}$, where Δ is the difference between the values of the objective function (energy) of the candidate solution (state) j and the current solution i and T is the process temperature. A prefixed value of T (e.g. $T = 0.001$) can be used to determine the stop of the entire process. As time elapses, T can decrease according to a so-called *cooling schema*.

In the following, some pseudo-code is given for the original SA algorithm with reference to a minimization problem.

Algorithm 3.2: Simulated Annealing

```
$i$ ← set initial solution = current solution
for time = 1 to time-budget do
    $T$ ← cooling-schema[time]
    if $T = 0.001$ then
        present current solution as the estimate of the optimal solution and stop
    generate a random neighbor $j$ of the current solution $i$ by performing a move.
    $\Delta = f(j) - f(i)$
    if $\Delta < 0$ then
        $i \leftarrow j$ accept $j$ as current solution
    else
        $i \leftarrow j$ accept $j$ as current solution with probability $p = e^{-\Delta/T}$
end for
```

Although this optimization technique is used in many other fields other than Operations Research, in the next sections attention remains focused on combinatorial problems. In particular, the application of a powerful tool, such as Markov chains, is exploited to present convergence results of the simulated annealing algorithm. In this analysis, the temperature (T) plays the role of a control parameter that changes as the optimization progresses, and thermodynamic equilibrium is replaced by equilibrium of a Markov process.

3.2.2 Convergence proof

In a combinatorial optimization problem let:

S	the feasible state space given by a finite number of points;
S^*	the set of global optimal solutions;
i	the system state (or configuration) $\forall i \in S$
$f(i): S \to R$	the objective function to minimize (maximize).

For each configuration i, let $N(i)$ be a subset of configurations called the *neighborhood* of i. If a transition allows to obtain a new configuration from a given one, then a neighborhood can be defined as the set of configurations that can be obtained from a given configuration in one transition or *move*. When the system is in state i, the probability of generating candidate state j as next is given by probability g_{ij}, thus $N(i) = \{j : g_{ij} > 0\}$.

As the system transitions from state to state, in order to examine the underlying *state path* in terms of a Markov chain, some other definitions and properties must be provided for future use. To begin with, given any two states i and j, if there exists a finite sequence of K states $l_1, l_2, ..., l_K$ such that $i = l_1$, $j = l_K$ and $g_{l_m l_{m+1}} > 0$ with $i_{m+1} \in N(i_m)$ for $m = 1,2,..., K-1$, then S is connected and $G = \{g_{ij}\}$ is *irreducible*.

Once a candidate state j (with $j \in N(i)$) has been generated from i, the SA algorithm will accept configuration j with a probability given by

$$a_{ij} =$$

$$= \frac{\exp(-f(j)/T)}{\exp(-f(j)/T) + \exp(-f(i)/T)}$$

$$= \frac{1}{1 + \exp((f(j) - (f(i))/T)}$$

$$\cong \exp(-(f(j) - (f(i))/T) \qquad (3.1)$$

As one may observe, the probability of choosing j decreases for increasing values of $(f(j) - f(i))$ and for decreasing values of T. Obviously, cost-decreasing transitions where $f(j) < f(i)$ are always accepted (in that case, in fact, the aforementioned probability is equal to 1).

By keeping a fixed value of T (as the iterations progress), the configurations that are consecutively visited by the SA algorithm can be seen as the states of a time-homogeneous Markov chain with transition matrix $P = P(T)$ whose elements are defined as:

$$p_{ij}(T) = \begin{cases} g_{ij} \cdot a_{ij}(T) & \text{if } j \neq i \\ 1 - \sum_{k=1}^{|S|} g_{ik} \cdot a_{ik}(T) & \text{if } j = i \end{cases}. \qquad (3.2)$$

If the neighboring states are equiprobable, the generation probabilities g_{ij} in (3.2) are given by

$$g_{ij} = \begin{cases} |N(i)|^{-1} & \text{if } j \in N(i) \\ 0 & \text{otherwise} \end{cases}, \qquad (3.3)$$

while the acceptance probabilities in (3.2) $a_{ij}(T)$ are

$$a_{ij}(T) = \min\left\{1, \exp\left(\frac{-(f(j) - f(i))}{T}\right)\right\}. \qquad (3.4)$$

Note that for $T > 0$, the irreducibility of G together with $A(T) > 0$ implies that P is irreducible, which means that in the corresponding Markov chain any two states are accessible to each other in a finite number of steps (Bhat 1984). Furthermore, every single element of P is given by the product of two independent measures, which is also a fundamental hypothesis behind the possibility of adopting Markov chains for process modeling.

The existence of the stationary distribution of the above Markov chain (i.e. the probability distribution of the configurations after an infinite number of transitions) or the selection of a proper cooling schedule for the temperature - T if is not fixed - is at the basis of the convergence results for the simulated annealing algorithm.

To prove the former, for short, let $f(i) = f_i$, $\forall i \in S$ and let π_i be defined as $\pi_i \mathrel{\hat{=}} \pi_1 |N(i)| \cdot \exp^{-(f_i - f_1)/T}$, $\forall i \in S$. In addition, let G be symmetric (i.e. if i is a neighbor of j then j is a neighbor of i).

Recalling that

$$\sum\nolimits_{k \in S} \pi_k = 1 \tag{3.5}$$

then from the following

$$\pi_1 \cdot \sum\nolimits_{k \in S} |N(k)| \cdot \exp^{-(f_k - f_1)/T} = 1, \tag{3.6}$$

π_1 can be obtained as

$$\pi_1 = \frac{1}{\sum\nolimits_{k \in S} |N(k)| \cdot \exp^{-(f_k - f_1)/T}}. \tag{3.7}$$

Replacing π_1 in π_i

$$\pi_i = \frac{1}{\sum\nolimits_{k \in S} |N(k)| \cdot \exp^{-(f_k - f_1)/T}} \cdot |N(i)| \cdot \exp^{-(f_i - f_1)/T}$$

$$\pi_i = \frac{1}{\exp^{f_1/T} \cdot \sum\nolimits_{k \in S} |N(k)| \cdot \exp^{-f_k/T}} \cdot |N(i)| \cdot \exp^{-f_i/T} \exp^{-f_1/T}$$

$$\pi_i = \frac{1}{\sum\nolimits_{k \in S} |N(k)| \cdot \exp^{-f_k/T}} \cdot |N(i)| \cdot \exp^{-f_i/T}. \tag{3.8}$$

These π_i s satisfy the detailed balance equations

$$\pi_i P_{ij} \hat{=} \pi_j P_{ji} \tag{3.9}$$

Proof

$$\frac{|N(i)| \cdot \exp^{-f_i/T}}{\sum_{k\in S} |N(k)| \cdot \exp^{-f_k/T}} \cdot \frac{1}{|N(i)|} \cdot \min\left[1, \exp^{-(f_j - f_i)/T}\right]$$

$$= \frac{|N(j)| \cdot \exp^{-f_j/T}}{\sum_{k\in S} |N(k)| \cdot \exp^{-f_k/T}} \cdot \frac{1}{|N(j)|} \cdot \min\left[1, \exp^{-(f_i - f_j)/T}\right] \tag{3.10}$$

can be written

$$\exp^{-f_i/T} \cdot \min\left[1, \exp^{-(f_j - f_i)/T}\right] = \exp^{-f_j/T} \cdot \min\left[1, \exp^{-(f_i - f_j)/T}\right] \tag{3.11}$$

If $f_j < f_i$, then the above expression becomes

$$\exp^{-f_i/T} = \exp^{-f_j/T} \cdot \exp^{-(f_i - f_j)/T} \tag{3.12}$$

thus equivalence is reached with $\exp^{-f_i/T} = \exp^{-f_i/T}$. Similarly, if $f_i < f_j$ then from (3.10) equivalence is reached with $\exp^{-f_j/T} = \exp^{-f_j/T}$.

As far as the cooling schedule is concerned, clearly, the choice of T must be based on conditions that bring the algorithm state to converge in probability to the set of global minimum solutions. In a pioneer work, (Geman and Geman 1984) prove that to obtain a global minimum of $f(i)$, it suffices to select a schema according to which T decreases no faster than $T(k) = T_0 / \ln k$, with T_0 "large enough". For the special case in which $T(k) = c/\ln(1+k)$, (Hajek 1988) proves that a simple and necessary condition is that c be greater than or equal to the depth, suitably defined, of the deepest local minimum which is not a global minimum state.

Practically speaking, one may consider the probability density for acceptance of a new state given the just previous value expressed by (3.4) in the following form:

$$a_{ij}(T) = \frac{1}{|N(i)|} \cdot \exp[-(f(j) - f(i))/T]. \tag{3.13}$$

To assure from a statistical point of view that any state $j \in S$ can be sampled infinitely often during the annealing run, one may prove that the joint probability of not doing so in the iterations successive to k_0 is equal to zero, meaning that:

$$\prod_k (1 - a_{ij}(T)) \overset{k \to \infty}{=} 0 \tag{3.14}$$

which is equivalent to

$$\sum_k a_{ij}(T) \overset{k \to \infty}{=} \infty. \tag{3.15}$$

This stated, the point becomes proving that a chosen cooling schedule satisfies (3.15). For example, considering $T(k) = T_0 / \ln k$ in (3.13) yields

$$\sum_{k=k_0}^{\infty} a_{ij}(T) \geq \sum_{k=k_0}^{\infty} \exp(-\ln k) = \sum_{k=k_0}^{\infty} 1/k = \infty. \tag{3.16}$$

Thus, (3.15) is satisfied.

3.2.3 Some variants

Many types of modifications of the Simulated Annealing algorithm have been designed to solve discrete stochastic optimization problems.

Several variants proposed use different, consistent estimates of the objective function values. For instance, (Bulgak and Sanders 1988) introduce confidence intervals to determine if in each iteration of their procedure the difference between the estimates of the objective function values obtained for the current state and the candidate state is statistically significant. In (Haddock and Mittenthal 1992) the value of the objective function in a given state is estimated with one long simulation run. Due to the computational burden required to return an accurate solution, the use of a rapidly decreasing cooling schedule is preferred here. However, this feature does not guarantee convergence in probability to the set of global optimal solutions. (Fox and Heine 1995) are the first to retain observations from previously generated solutions and use them to estimate the value of the objective function at the current iteration.

Other forms of investigations focus on a more theoretical analysis. For example, (Gelfand and Mitter 1989) prove convergence with normally distributed noise $N\left(0,\sigma_k^2\right)$ on the estimated objective function, with variance $\sigma_k = o\left(T_k\right)$ as $k \to \infty$, provided that the sequence $\{T_k\}$ is chosen properly. Convergence results for this analysis are presented by (Gutjahr and Pflug 1996) as well. They also generalize their convergence proof for other noise distributions on the estimated objective function that are symmetric around zero and more peaked than Gelfand and Mitter's $N\left(0,\sigma_k^2\right)$.

At this point, attention is drawn to a major and more recent modification of the original simulated annealing presented in (Alrefaei and Andradóttir 1999). The method proposed herein discards the basic assumption common to all of the above studies in which the positive control parameter $T_k \to 0$ as $k \to \infty$. The primary innovation consists in letting the SA algorithm work on a constant temperature $T_k = T > 0$ for all $k \in \mathrm{N}$. This stated, two different variants are proposed to estimate the optimal solution.

In the first approach, the most visited configuration (divided by a normalizer) is used for the above estimation. To do so, in iteration k L_k observations are sampled from both the current and candidate states (i.e. $X_i(r)$ and $X_j\left(r\right)$, $r = 1..L_k$), where $\{L_k\}$ is assumed to be a sequence of positive integers such that $L_k \to \infty$ as $k \to \infty$. These samples are then used to estimate the values of the corresponding objective functions (i.e. $f\left(i\right)$ and $f\left(j\right)$, respectively), as shown in Algorithm 3.3.

Algorithm 3.3: Simulated Annealing with constant T

```
$R$, $N$, $T$, $\{L_k\}$ ← select procedure settings
$k = 0$ ← set initial iteration
```

```
$i_0 \in S$ ← set initial solution
$i_k^* = i_0$ ← store most visited solution
$V_0(i_0) = 1,\ V_0(i) = 0 \quad \forall i \in S,\ i \neq i_0$ ← count initial visit
**for** $k = 0$ **to** $\infty$ **do**
    $i = i_k \quad j = j_k \in N(i)$ ← generate neighbor of $i_k$ with probability $R(i, j)$
    **for** $r = 1$ **to** $L_k$ **do**
        $X_i(r)$ ← generate $L_k$ i.i.d. unbiased observations for current solution
        $X_j(r)$ ← generate $L_k$ i.i.d. unbiased observations for candidate solution
    **end for**
    $f(s) = \frac{1}{L_k}\sum_{r=1}^{L_k} f(X_s(r)),\ s = i, j$ ← estimate competing solutions
    $U_k \sim U[0,1]$ ← generate a random number
    $g_{ij}(k) = \exp\left[\frac{-[f(j) - f(i)]^+}{T}\right]$ ← compute the acceptance probability
    **if** $U_k \leq g_{ij}(k)$ **then**
        $i_{k+1} = j$ ← accept candidate solution
    **else**
        $i_{k+1} = i$ ← accept current solution
    $k = k + 1$ ← increase iteration
    $V_k(i_k) = V_{k-1}(i_k) + 1$ ← count visit for accepted solution
    $V_k(i) = V_{k-1}(i) \quad \forall i \in S,\ i \neq i_k$ ← do not count visits for other solutions
    **if** $V_k(i_k)/D(i_k) > V_k(i_{k-1}^*)/D(i_{k-1}^*)$ **then**
        $i_k^* = i_k$ ← update best solution
    **else**
        $i_k^* = i_{k-1}$ ← keep current best solution
**end for**
$i_k^*$ ← return best solution
```

Under the proper assumptions, the sequence of states $\{i_k\}$ visited by the proposed SA algorithm is a time-inhomogeneous Markov chain.

Let $V_k(i)$ ($i \in S$) be defined as the number of times the Markov chain $\{i_k\}$ has visited state i in the first k iterations and $D(i)$ be a normalizer defined as

$$D(i) = \sum_{i' \in S} R'(i,i'). \qquad (3.17)$$

By one of the ergodic properties of Markov chains (Heyman and Sobel 1982), for which Andradóttir presents a more general case in Lemma 3.1 of (Andradóttir 1995),

$$\frac{V_k(i)}{kD(i)} = \frac{1}{D(i)} \cdot \frac{1}{k} \sum_{l=0}^{k} I_{\{i_l = i\}} \to \frac{\pi_i}{D(i)} \text{ a.s. as } k \to \infty \qquad (3.18)$$

for all $i \in S$, where $I_{\{i_l = i\}}$ is an indicator random variable defined as

$$I_{\{i_l = i\}} = \begin{cases} 1 & \text{if } i_l = i \\ 0 & \text{otherwise} \end{cases}. \qquad (3.19)$$

From the definition of π_i (the stationary distribution of P which is greater than zero for all $i \in S$) $\pi_j / D(j) \le \pi_i / D(i)$ if and only if $f(j) \ge f(i)$. This shows that

$$\arg\max_{i \in S} \frac{\pi_i}{D(i)} = S^*. \qquad (3.20)$$

Now let A be such that $P(A) = 1$ and for all $\omega \in A$ $V_k(i,\omega)/k \to \pi_i$ as $k \to \infty$ for all $i \in S$. Let $\omega \in A$. Then, since S is finite and $V_k(i,\omega)/kD(i) \to \pi_i / D(i)$ as $k \to \infty$ for all $i \in S$, it follows from (3.20) that there exists K_ω such that for all $k \ge K_\omega$, $V_k(i^*,\omega)/D(i^*) > V_k(i,\omega)/D(i)$ for all $k \ge K_\omega$, $i^* \in S^*$ and $i \in S \setminus S^*$. Hence, $i_k^*(\omega) \in S^*$ for all $k \ge K_\omega$.

It is worth observing that Algorithm 3.3 also converges with any consistent estimates of the objective function values.

The second approach in (Alrefaei and Andradóttir 1999) is based on a very similar structure, but, unlike the first variant, it selects the state with the best average estimated objective function value as optimal solution. This estimate is obtained from all the previous estimates of the objective function values sampled for that state.

In particular, $\{L_k\}$ is assumed to be a sequence of positive integers that satisfies $\lim_{k\to\infty} L_k = L \leq \infty$. In the case in which $L = \infty$, if $A_k(i)$ is defined as the cumulative value of the estimates of the objective function values over the first k iterations for all $i \in S$ and $C_k(i)$ is the corresponding number of observations, then the convergence theorem is proved by the strong law of large numbers by which $A_k(i)/C_k(i) \to f(i)$ a.s. as $k \to \infty$ for all $i \in S$. If $L < \infty$, the new transition probability matrix P' is proven to be still irreducible and aperiodic with a stationary distribution r, where $r_i > 0$ for all $i \in S$. The rest of the proof is similar to the proof with $L = \infty$.

It is worth observing that for large values of L, this variant has a greater convergence rate to the set of global optimal solutions than the previous one. However, practically speaking, it is quite difficult to determine in advance the value of L for which the search is drawn to the good states.

The constant temperature approach is then pursued by other authors. (Ahmed and Alkhamis 2002) propose a method that combines a constant temperature-based simulated annealing with a two-stage Ranking and Selection procedure and is known by the acronym “SARS”. They show that the most visited configuration during the first m iterations converges almost surely to a globally optimum solution.

In particular, they replace the classical sample mean with the more sophisticated linear combination first introduced by (Dudewicz and Dalal 1975):

$$\tilde{X}_i(n_i) = w_i \overline{X}_i(n_0) + (1 - w_i)\overline{X}_i(n_i - n_0) \qquad (3.21)$$

where the weights (w_i) are defined as

$$w_i = \frac{n_0}{n_i}\left[1 + \sqrt{1 - \frac{n_i}{n_0}\left(1 - \frac{(n_i - n_0)\cdot\delta^2}{h^2 S_i^2(n_0)}\right)}\right] \qquad (3.22)$$

and $\overline{X}_i(n_0)$ and $\overline{X}_i(n_i - n_0)$ are respectively the sample mean computed in the first and second stage of the R&S procedure. At this point, the authors demonstrate that $\lim_{k\to\infty} \tilde{X}_i(k) = \lim_{k\to\infty}\{w_i \overline{X}_i(n_0) + (1-w_i)\overline{X}_i(n_i(k)-n_0)\} = f(i)$ *with probability 1* and, thus, they propose $\tilde{X}_i(k)$ as estimate for $f(i)$ at iteration k.

Reminding that the convergence of the simulated annealing algorithm lies in the homogeneity of the underlying Markov chain, one may recognize that the core of their work relies on the following preposition: $P_k = p_{ij}^{(k-1,k)} \to P = p_{ij}$ as $k \to \infty$ where P is the transition matrix of the time homogeneous Markov chain given by equation (3.2).

Proof

Let $p_k = \exp\left(\dfrac{-\left[\tilde{X}_j(k) - \tilde{X}_i(k)\right]^+}{T}\right)$. If $i \neq j$ then

$$p_{ij}^{(k-1,k)} =$$

$$= \Pr\{i_k = j \mid i_{k-1} = i\} \qquad (3.23)$$

$$= g_{ij} \Pr\{U_k \leq p_k\} \qquad (3.24)$$

$$= g_{ij} \int_0^1 \Pr\{p_k \geq u_k \mid U_k = u_k\} f(u_k) du_k \qquad (3.25)$$

$$= g_{ij} \int_0^1 \left(1 - F_{p_k}(u_k)\right) du_k \qquad (3.26)$$

$$= g_{ij} E[p_k] \qquad (3.27)$$

$$= g_{ij} E\left[\exp\left(-\left[\tilde{X}_j(k) - \tilde{X}_i(k)\right]^+ / T\right)\right] \qquad (3.28)$$

and

$$\lim_{k\to\infty} p_{ij}^{(k-1,k)} = g_{ij} \lim_{k\to\infty} E\left[\exp\left(-\left[\tilde{X}_j(k) - \tilde{X}_i(k)\right]^+ / T\right)\right] \qquad (3.29)$$

Since $\left|\exp\left(-\left[\tilde{X}_j(k)-\tilde{X}_i(k)\right]^+ \big/ T\right)\right| \leq 1$, the bounded convergence theorem gives

$$\lim_{k\to\infty} p_{ij}^{(k-1,k)} =$$

$$= g_{ij} E\left[\lim_{k\to\infty} \exp\left(-\left[\tilde{X}_j(k)-\tilde{X}_i(k)\right]^+ \big/ T\right)\right] \qquad (3.30)$$

$$= g_{ij} E\left[\exp\left(\lim_{k\to\infty}\left(-\left[\tilde{X}_j(k)-\tilde{X}_i(k)\right]^+ \big/ T\right)\right)\right] \qquad (3.31)$$

Using $\tilde{X}_i(k)$ as estimate for $f(i)$

$$\lim_{k\to\infty} p_{ij}^{(k-1,k)} = g_{ij}\left[\exp\left(-\left[f(j)-f(i)\right]^+ \big/ T\right)\right] \qquad (3.32)$$

If $i = j$ then

$$p_{ij}^{(k-1,k)} = 1 - \sum\nolimits_{i\neq j} p_{ij}^{(k-1,k)} \qquad (3.33)$$

$$\lim_{k\to\infty} p_{ij}^{(k-1,k)} =$$

$$= \lim_{k\to\infty}\left[1 - \sum\nolimits_{i\neq j} p_{ij}^{(k-1,k)}\right] \qquad (3.34)$$

$$= 1 - \sum\nolimits_{i\neq j} p_{ij} \qquad (3.35)$$

$$= p_{ii}. \qquad (3.36)$$

Therefore

$$\lim_{k\to\infty} p_{ij}^{(k-1,k)} = p_{ij} \quad \forall i, j \in S. \qquad (3.37)$$

Once that $P(k) \to P$ as $k \to \infty$ has been demonstrated, the discussion of the convergence of the sequence of states $\left\{i_k^*\right\}$ described by the SARS algorithm follows the framework already provided by (Alrefaei and Andradóttir 1999).

The same authors, in another work (Ahmed and Alkhamis 2004) extend the constant temperature approach to deal with the sampling error stemming from the stochastic nature of the simulation output used to estimate the objective function value. To this end, in iteration k, N_k independent observations (where $N_k \to \infty$ as $k \to \infty$) are generated for $D_{ji} = X_j - X_i$, which represents the difference between the value of the objective function in state j and i.

Let $\overline{D}_{ji} = \overline{X}_j - \overline{X}_i = \frac{1}{N_k}\sum_{l=1}^{N_k} D_{ji}^l$ and $\hat{\sigma}_k = \frac{1}{\sqrt{N_k}}\sqrt{\frac{1}{N_k - 1}\sum_{l=1}^{N_k}\left(D_{ji}^l - \overline{D}_{ji}\right)^2}$ be respectively D_{ji}'s sample mean and sample standard error of the mean. In their variant of the SA algorithm, the transition matrix for the k-th step is given by

$$p_{ij}(k) = P\{i_{k+1} = j \mid i_k = i\} =$$

$$= \begin{cases} g_{ij} P\left\{U_k \le \exp\left[\frac{-\left[\overline{X}_j - \overline{X}_i - t_k\hat{\sigma}_k\right]^+}{T}\right]\right\} & \text{if } j \in N(i) \\ 1 - \sum_{l \in N(i)} P_{il}(k) & \text{if } j = i \end{cases} \tag{3.38}$$

where t_k denotes a selected upper critical value of Student's distribution with N_k–1 degrees of freedom and U_k is a uniform random variable defined on the interval $[0,1]$. To verify that their approach is guaranteed to converge almost surely to the set of global solutions, again, they concentrate on proving that $P(k) \to P$ as $k \to \infty$. Specifically:

$$\lim_{k\to\infty} p_{ij}(k) = g_{ij} \lim_{k\to\infty} P\left\{U_k \le \exp\left[\frac{-\left[\overline{X}_j - \overline{X}_i - t_k\hat{\sigma}_k\right]^+}{T}\right]\right\} \tag{3.39}$$

$$= g_{ij} \lim_{k\to\infty} E\left[\exp\left[\frac{-\left[\overline{X}_j - \overline{X}_i - t_k\hat{\sigma}_k\right]^+}{T}\right]\right] \tag{3.40}$$

$$= g_{ij} E\left[\exp\left[\frac{-[f(j) - f(i)]^+}{T}\right]\right] \tag{3.41}$$

$$= p_{ij}. \qquad (3.42)$$

Therefore, since $\hat{\sigma}_k \xrightarrow{p} 0$ as $N_k \to \infty$, which is an assumption at the basis of this work, the rest is very similar to (Alrefaei and Andradóttir 1999), including the use of the state that is most visited by the algorithm as the estimated optimal solution.

3.3 Practical limits

In the previous section, the convergence results given for the constant temperature-based simulated annealing variants are obtained as k, the number of iterations required by the SA algorithm, goes to infinity. Bearing this in mind, the numerical applications proposed in these modifications involve very simple test cases such as an $M/M/1$ queuing system (Alrefaei and Andradóttir 1999) or a system with at the most 20 configurations (Ahmed and Alkhamis 2002). Therefore, from a computational point of view, it is possible to carry out multiple visits to every state of the system and estimate the best performing configuration. In contrast, larger problems are likely to be unsolvable in a reasonable amount of time and this restriction brings similar approaches to be non-applicable to many practical situations. For example, the SA algorithm proposed in (Roenko 1990) is marked as “non-realistic” by different authors since it calls for storing all the feasible solutions encountered during the execution of the algorithm in order to perform comparison with each newly generated solution. This is also the case of the *quay crane scheduling problem* (QCSP) described in §1.3.

Let n be the number of fixed holds to be operated for discharge/loading operations and $n_i \geq 0$ the number of holds assigned to crane i, $i = 1..m$ ($\sum_{i=1}^{m} n_i = n$ is a necessary imposition). Although the state space of this particular problem is finite, the number of states is very large and equal to the number of unordered partitions of n holds among m cranes (Liu 1968):

$$\text{number of states} = \begin{pmatrix} n+(m-1) \\ (m-1) \end{pmatrix} * n! = \frac{(n+(m-1))!}{(m-1)!}. \qquad (3.43)$$

As one can observe, even a very limited number of cranes and holds may generate difficult-to-solve combinatorial problem instances. For example, for a medium-size vessel with 8 holds waiting to be operated by 3 cranes, the total number of possible combinations is 1.814.400. Therefore, a frequent exploration of every alternative hold-crane schedule goes beyond practical possibilities.

In response to this concrete request for performance, a non-prohibited option currently under investigation consists in introducing a guided-search refinement in the SA algorithm based on a different choice of the candidate solution. In particular, at iteration k, let i be the current solution in a minimization problem and m, with $m > 1$, the number of candidate neighboring configurations to be generated from the current solution. At this point, the refinement consists in defining j as $j = \arg \min\limits_{\substack{l=1..m \\ j_l \in N(i)}} f(j_l)$, meaning that, among the m neighboring solutions $j_1, j_2, ..., j_m$ of configuration i, j will be chosen as the candidate solution. This statement does not affect the mechanism of solution acceptance, according to which the corresponding probabilities are still given by (3.4). If G is proven to remain irreducible, then once again, the configurations that are consecutively visited by the SA algorithm can be seen as the states of a time-homogeneous Markov chain with transition matrix $P = P(T)$ defined in (3.2) and with stationary distribution defined by (3.8).

Clearly, on one hand, the object of the above approach is to reach a globally optimum solution within a finite number of iterations by evaluating a sufficient, but not exhaustive, number of configurations. On the other, the approach requires accurate estimates for $f(j_1), ..., f(j_m)$ and $f(i)$, while also guaranteeing the selection of the best configuration according to a pre-defined level of confidence. Both of these prerequisites can be met by using an n-

stage Ranking and Selection procedure within the SA algorithm, as illustrated by the following pseudo-code, in which $n = 2$.

```
Algorithm 3.4: Modified Simulated Annealing
$G$, $N$, $T_k$, $T_f$, $\alpha$, $m$, $n_0$, $\delta$, $h$ ← select input parameters
$k = 0$ ← set initial iteration
$i_0 \in S$ ← set initial solution
$i_k^* = i_0$ ← store best solution
while $T_k > T_f$ do
  $i = i_k$ ← set current solution
  for $l$ = 1 to $m$ do
    $j_l \in N(i)$ ← generate neighbor of $i$ with probability $G(i, j_l)$
    for $r$ = 1 to $n_0$ do
      $X_{j_l}(r)$ ← generate $n_0$ i.i.d. unbiased observations for candidate solution $j_l$
    end for
    $f(j_l(n_0)) = \frac{1}{n_0}\sum_{r=1}^{n_0} f(X_{j_l}(r))$,
    $S_{j_l}^2(j_l(n_0)) = \frac{1}{n_0 - 1}\sum_{r=1}^{n_0}\left(f(X_{j_l}(r)) - f(j_l(n_0))\right)^2$ ← compute first-stage sample mean and variance of candidate solution $j_l$
    $N_{j_l} = \max\left(n_0, \frac{h \cdot S_{j_l}}{\delta}\right)^2 \quad \forall\ j_l \in N(i)$
  end for
  if $n_0 < N_{j_l} \quad \forall\ j_l \in N(i)$ then
    for $r$ = $n_0$ + 1 to $N_{j_l}$ do
      $X_{j_l}(r)$ ← generate $N_{j_l} - n_0$ i.i.d. unbiased observations $\forall\ j_l \in N(i)$
    end for
    $f_k(X_{j_l})$ ← generate $N_{j_l} - n_0$ i.i.d. unbiased observations for candidate solution $j_l$
    $f(j_l(N_{j_l})) = \frac{1}{N_{j_l}}\sum_{r=1}^{N_{j_l}} f(X_{j_l}(r))$ ← compute second-stage sample
```

```
         mean of candidate solution $j_l$
21:  $j = \arg\min_{\substack{l=1..m \\ j_l \in N(i)}} f(j_l)$ ← set $i$'s best neighbor as final candidate solution
22:  $U_k \sim U[0,1]$ ← generation of a random number
23:  $a_{ij}(k) = \exp\left[\frac{-\left[f(j(N_j)) - f(i(N_i))\right]^+}{T_k}\right]$ ← compute acceptance probability
24:  **if** $U_k \le a_{ij}(k)$ **then**
25:      $i_{k+1} = j$ ← accept the candidate solution
26:  **else**
27:      $i_{k+1} = i$ ← accept the current solution
28:  $k = k + 1$ ← increase iteration
29:  **if** $f(i_k) < f(i_k^*)$ **then**
30:      $i_k^* = i_k$ ← update best solution
31:  $T_k = \alpha \cdot T_{k-1}$ ← decrease temperature
32: **end while**
33: $i_k^*$ ← return best solution
```

Clearly, a further enhancement on Algorithm 3.4 could be achieved by introducing (at line 23) either an interval estimate as presented in (Alkhamis and Ahmed 2004) or a hypothesis test on the “differences” between candidate and current solutions. Both modifications would represent a supplementary aid to reduce statistical errors that affect the acceptance probability of a given solution.

3.4 Application of the SA algorithm

As previously stated, the simulated annealing algorithm is widely used in many disciplines other than Operations Research. As a matter of fact, the SA list of applications includes combinatorial optimization problems related to diverse scientific and technical fields among which very-large-scale integration (VLSI) design, image processing, neural networks and so on (Aarts and van Laarhoven 1987).

Whatever the discipline or project, if a decreasing temperature schema is chosen, then the *initial temperature* (T_0), *cooling schedule* and *final temperature* (if the latter is eventually adopted as a *stopping criteria*) must all undergo a setting and calibration process in order to provide results that are consistent with the goal of the application. With respect to this issue, it is worth recalling that:

- if some (weak) hypotheses hold on the *cooling schedule*, it is possible to demonstrate that the algorithm converges in probability to the set of global optimal solutions;
- if the temperature is high and a minimum-state energy is found, then the algorithm continues running to eventually escape from the above state should it just be a local minimum;
- if the temperature decreases too rapidly, then some thermal fluctuations are frozen: in the OR language, this means that the algorithm may stay trapped in a local minimum;
- if the temperature decreases too slowly, then the search process may take too long and, thus, the system may not reach the steady state (i.e. a global minimum).

Table 3.1 - Examples of cooling schedule for SA

Discipline	Cooling Schedule	Properties
mathematics/combinatorics	$T_{i+1} = \alpha T_i$	$0{,}70 \le \alpha \le 0{,}99$
data analysis	$T_i = FT_{i-1}$	$F = \left(T_{\min} / T_{\max}\right)^{1/N_{cycles}}$
Biology	$T_{i+1} = \gamma T_i$	$\gamma = \left(T_{final} / T_{initial}\right)^{1/\left(N_{cycles}-1\right)}$
Finance	$T_{i+1} = r_T T_i$	$0 \le r_T \le 1$ *ad hoc* selection

Unfortunately, in the literature there is no algorithm that can determine “correct” values for the initial (final) temperature and cooling schema, but, as suggested by empirical knowledge, simple cooling schemas seem to work well. With respect to specific disciplines, some examples are given in Table 3.1 (Ingber 1993).

To remark the importance of temperature parameters and how their correct setting impacts on the search process, one may start by considering the simple cooling scheme adopted in mathematics/combinatorics $T_{i+1} = \alpha T_i$. With the cooling rate $\alpha = 0.9$, the initial temperature T_0 set equal to 100 and the final temperature $T_f = 1$, in terms of *cycle length* or number of competing configurations evaluated, the SA algorithm generates and explores 44 alternative solutions, as depicted by the blue trend-line in Figure 3.1. However, in a highly combinatorial optimization problem, this number may be far from satisfactory. The inadequacy of α's previous setting is even more pronounced for low values of the initial temperature: for example, if $T_0 = 10$, then the corresponding cycle length drops to 22 as illustrated by the red trend-line in Figure 3.1.

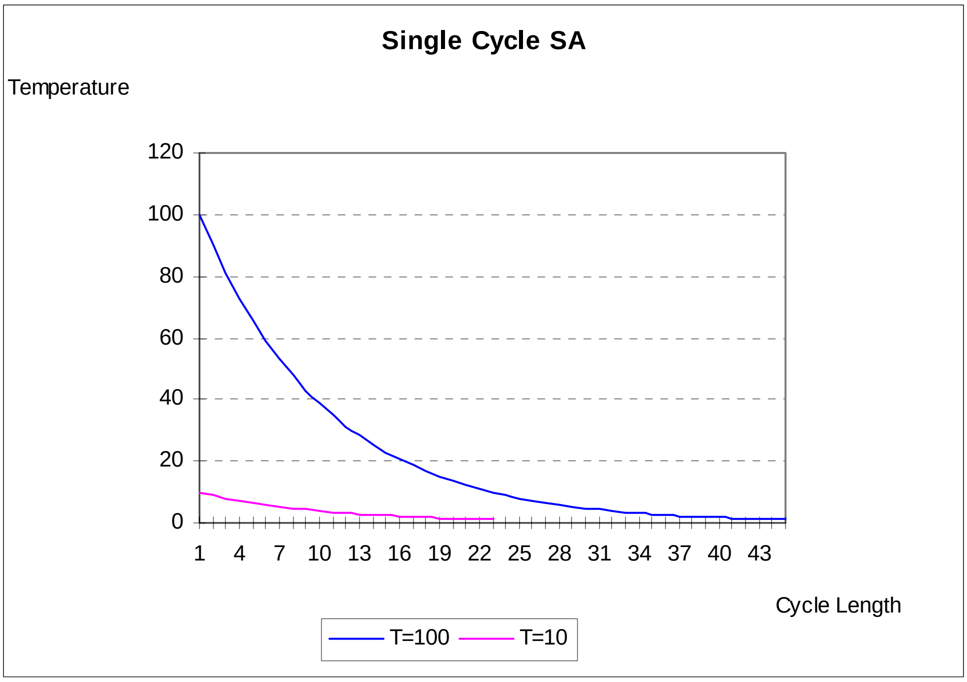


Figure 3.1 - Configurations explored by a single cycle SA

The introduction of nested cycles in the SA procedure can prevent a premature termination of the search process. In logistics, this type of adjustment can be found in (Kim and Moon 2003) where the SA approach involving a pair of nested loops is applied to the berth scheduling problem; in a preliminary study by (Legato and Mazza 2007) a three-level nested SA algorithm is at the basis of an optimization by simulation procedure for the quay crane scheduling problem. As for a clarifying example, in the mathematics/combinatorics schema one may consider maintaining the same cooling schedule, cooling rate and final temperature, as well as the values for the initial temperatures (i.e. $T_0 = 100$ and $T_0 = 10$). With these settings, the number of solutions evaluated by a three-level nested algorithm, rise to 132 and 66, respectively as shown by Figure 3.2.

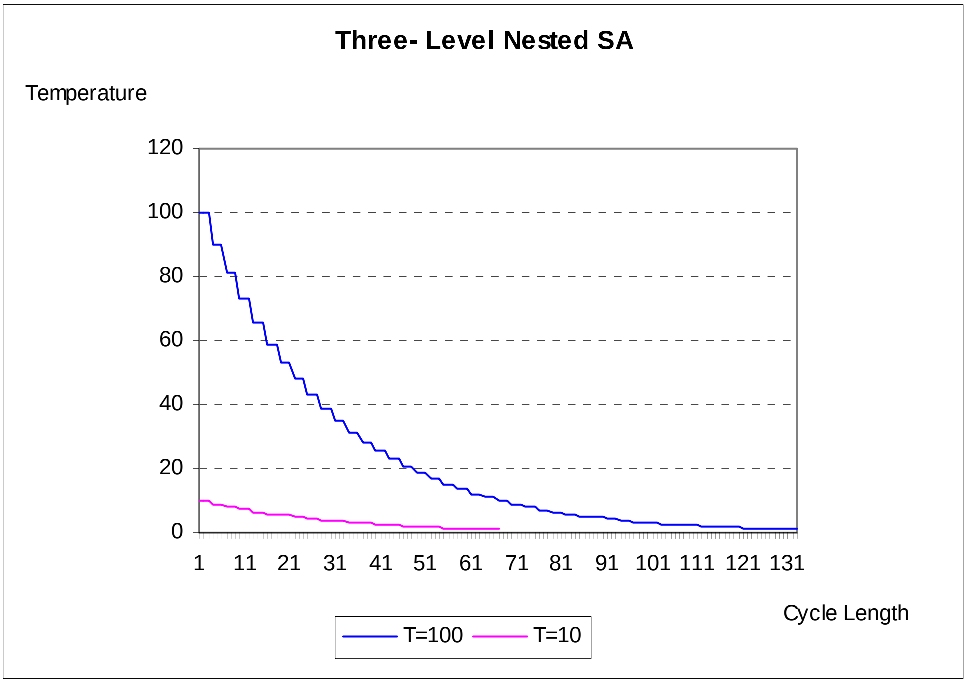


Figure 3.2 - Configurations explored by a 3-level nested SA

In the next sections, these and other issues will be further discussed and tailored to the quay crane scheduling problem, followed by numerical experiments on real data.

3.4.1 Customization to the QCSP

The application of the simulated annealing approach to determine the *makespan* in the QCSP is set-off by performing some choices required by the customization process for the problem at hand.

To begin with, as previously stated, choosing the proper cooling schema impacts on reaching a global minimum. In this particular problem, it affects the number of hold-quay crane assignment schedules (solutions) evaluated by running the SA algorithm. To this end, testing continues on the so-called simple “mathematics” cooling schema $T_{i+1} = \alpha \cdot T_i$ according to which the best results have been returned for an initial temperature $T_0 = 100$ and a decreasing rate $\alpha = 0.995$.

The “move” definition for neighborhood generation is also very context-sensitive. For the QCSP, with respect to (eventual) release, precedence and non-simultaneity constraints that determine the feasibility (or lack thereof) of a container discharge/loading schedule, some examples of moves are:

- move hold l assigned to crane i from position r to position s ($r \neq s$) within the same crane i;
- move hold l from position r on crane i to position s on crane j ($i \neq j$);
- swap the positions of holds l and k ($l \neq k$) on crane i;
- swap the positions of holds l and k ($l \neq k$), originally assigned to cranes i and j ($i \neq j$), respectively.

In the SA implementation currently under examination, the second option for the move definition has been adopted.

As far as the stopping criteria is concerned, QCSP designers can chose among the following possibilities:

- stop when the algorithm has reached a fixed number of iterations n or an upper bound on the available time-budget;
- stop when the current solution has not been updated in the last m iterations;

- stop when the cooling schema has reached a fixed *lower bound* on the temperature value (T_f).

For this setting, the lower bound temperature value $T_f = 10^{-5}$ has been chosen.

3.4.2 Numerical experiments

Numerical experiments discussed in this section use a simplified simulation model referred to the queuing network for the discharge/loading process depicted by Figure 1.4 and describing the operations around the QCSP. Specifically, both the “SC waiting line on quay” and the “TEUs waiting line under crane” have been short-circuited with the purpose of isolating and highlighting the random effects of process discharge/loading times upon the schedules and, therefore, on the *makespan*.

The object of the analyses reported in the following is twofold. On one hand, experiments on the QCSP mean to investigate and compare the performance of the SA algorithm when system dynamics are affected by one major source of uncertainty: the discharge/loading service times operated by the quay cranes (measured in container moves per hour). The results returned are also examined in relation to the optimal value found by the commercial LP software CPLEX for the stand-alone optimization model proposed in section 3 of (Legato et al. 2008b), which provides a lower bound on the value of the *makespan* when data is deterministic. On the other hand, the same tests intend to show how a simulation-based optimization algorithm is often the only practical solution method available when dealing with difficult-to-solve combinatorial problem instances, embedded in realistic, dynamic environments characterized by several elements of randomness.

This matter is even more evident as soon as one considers the real medium-size vessel illustrated in Figure 3.3 (courtesy of the container terminal in Gioia Tauro) for which a limited number of holds $n = 14$ must be operated by a small number of cranes $m = 3$.

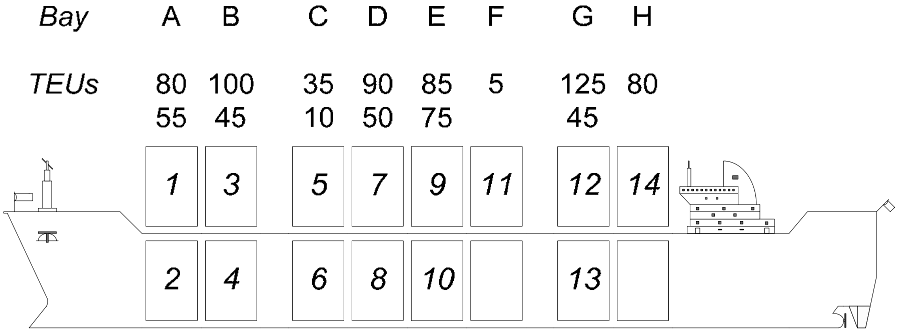


Figure 3.3 - Map with discharge/loading info per vessel bay

Although the state space of this particular problem is finite, the number of states is very large: as a matter of fact, as many as $1.04614 \cdot 10^{13}$ possible combinations may occur. Therefore, the exploration of every alternative schedule could go beyond practical possibilities. However, in this case, the number of feasible schedules that can be generated and, thus, evaluated is smaller due to the precedence and non-simultaneity constraints between vessel holds (i.e. *task pairs*) summarized in Table 3.2.

Numerical experiments are carried out on three different scenarios according to which the quay crane discharge/loading times can either be deterministic or follow an exponential or hyper-exponential distribution law. Computational efforts focus on these two particular laws because of their aptitude to represent a growing process variance related to the discharge/loading times.

Table 3.2 - Problem constraints

Constraints	*Task pairs*		
precedence	(1,2)	(3,4)	(5,6)
	(7,8)	(9,10)	(12,13)
non-simultaneity	(1,3)	(5,7)	(7,9)
	(12,14)	(2,4)	(6,8)
	(8,10)	(1,4)	(2,3)
	(5,8)	(6,7)	(7,10)
	(8,9)	(10,11)	(13,14)

While the specific settings for the simulation-based optimization procedures have already been discussed and reported in paragraph 3.4.1, here a special mention is deserved by Rinott's two-stage indifference-zone based Ranking and Selection procedure (1978), which has been introduced in the over-all framework to perform a correct selection between competing schedules with at least probability P^*. The common parameters of the R&S procedure are: the initial number of simulation runs $n_0 = 10$, the confidence level $1-\alpha$ with $\alpha = 0.1$ and the indifference zone $\delta = 0.25\,\text{h}$ on the *makespan* value. Additional input specifications concern both the quay crane discharge/loading rate (i.e. 28 container moves per hour) and the initial hold - quay crane assignment schedule for the QCSP which is selected randomly.

In general, once parameters are set, the estimate of the objective function produced by the algorithm converges to the value of the optimal solution, as the number of iterations grows. In the long run, if compared to algorithms characterized by particular global-local search paradigms (see Legato et al. 2008b), the SA algorithm slightly outperforms these other classes in terms of average execution time and quality of the *makespan* estimate. This is due to the algorithm's specific capability of jumping out of local minima by accepting candidate solutions that are worse than the current solution. In contrast, the SA convergence begins at a later stage (see Figure 3.4), as a result of procedure set-up where a random generation of the initial hold-crane assignment takes place. Recalling that in the context of simulation-based optimization evaluating the objective function entails running the simulation model, being able to find high quality solutions early in the search is of critical importance. A logical answer to this fault can consist in providing an "educated" starting solution intended for (eventual) improvement. In this case, the SA search process can appear to be more effective since the starting point is set on a solution that is "close" to high quality ones which can be reached by a single move according to the predetermined mechanism for neighborhood definition.

As one may observe in Figure 3.4, under deterministic quay crane service times, the average makespan value determined by the SA algorithm for the problem at hand converges to the lower bound of *10.536 hours* returned by CPLEX for the IP formulation (1)-(8) proposed in (op. cit.). Despite that an exhaustive coverage of all the possible combinations in the quay crane scheduling problem is not performed by the above algorithm, nor is any sort of control running on which part of the feasible set is being explored, the schedule returned as final output (in a large number of numerical tests carried out within these experiments) is already situated within the indifference-zone of the optimal solution (i.e. 15 minutes) after 2000 iterations. Results are provided in just a few seconds, while CPLEX returns the optimal solution after several minutes ($\cong 25$).

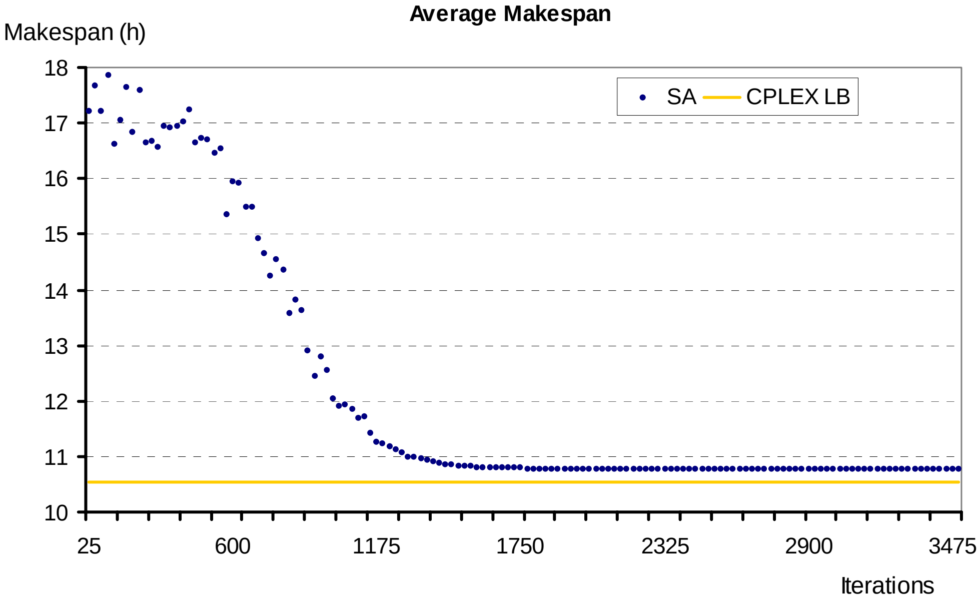


Figure 3.4 - Makespan for deterministic service times

When quay crane service times are non-deterministic the optimal solution of the IP formulation is no longer truly representative of the discharge/loading operations since the corresponding mathematical model does not account for uncertainty. This becomes more pronounced as the process variance increases due to greater randomness in the quay crane operational cycle (e.g. delays, blockages, failures). As shown in Figure 3.5, after more or less 2500 iterations, the SA algorithm returns a *makespan* value of 10.7 hours. This value is still close to the optimal value previously returned by CPLEX (deterministic case) because the randomness introduced by the exponential law does not produce significant effects on the non-simultaneity and precedence constraints.

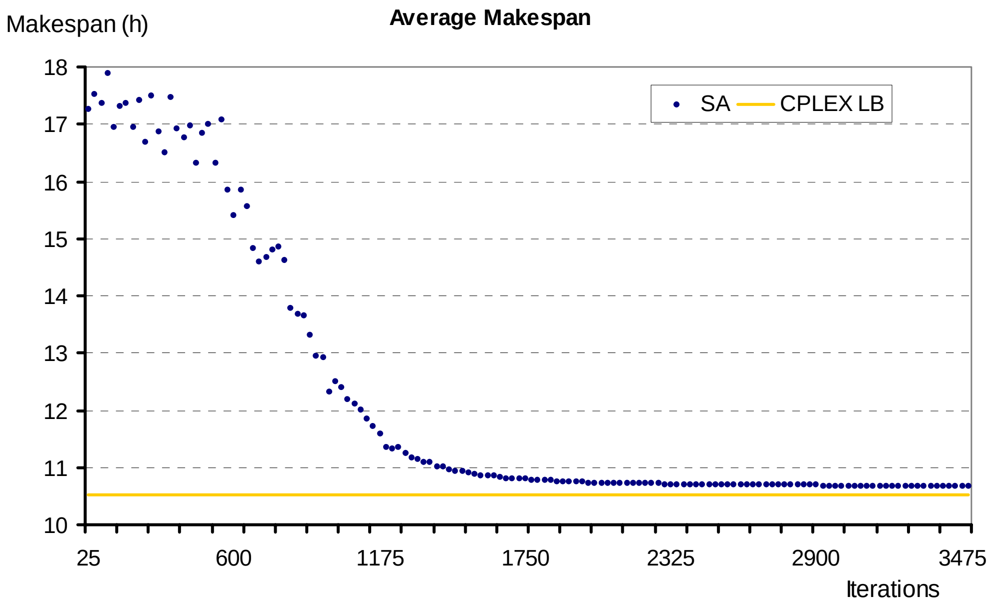


Figure 3.5 - Makespan for exponential service times

It is worth observing that algorithm performance does not deteriorate when dealing with exponential service times (see Figure 3.6).

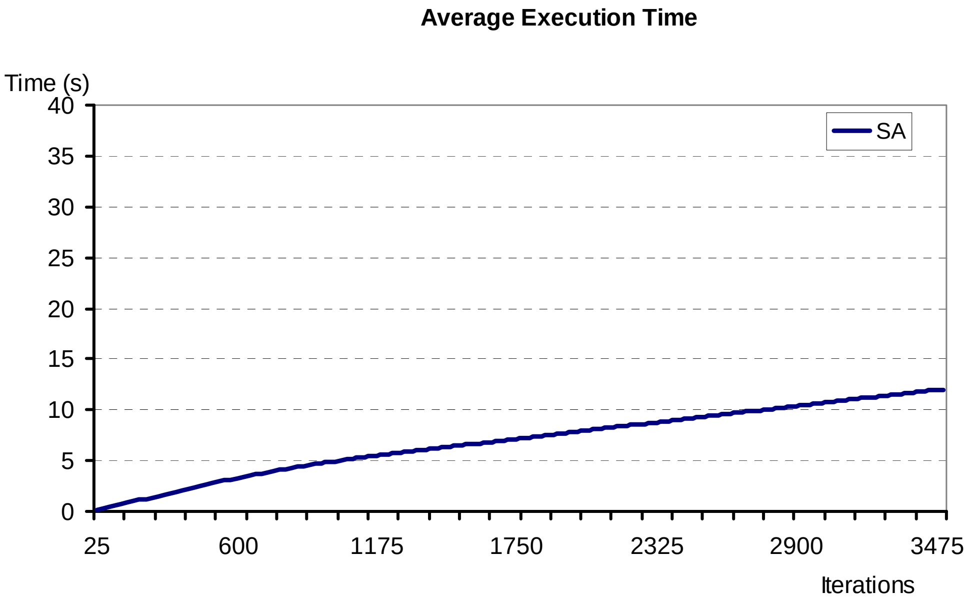


Figure 3.6 - Algorithm performance for exponential service times

Conclusions differ a great deal when in the last scenario discharge/loading operations are modeled with a hyper-exponential distribution (according to which quay crane service occurs with probability 0.95 at a rate of 28 container moves per hour and with probability 0.05 at a rate of 2 container moves per hour). As mentioned previously, a similar set-up is particularly suitable for modeling quay crane stoppage events during operations. Figure 3.7 shows how the SA algorithm achieves an average *makespan* estimate which departs from the value of the objective function of the solution found with CPLEX by more than 80%.

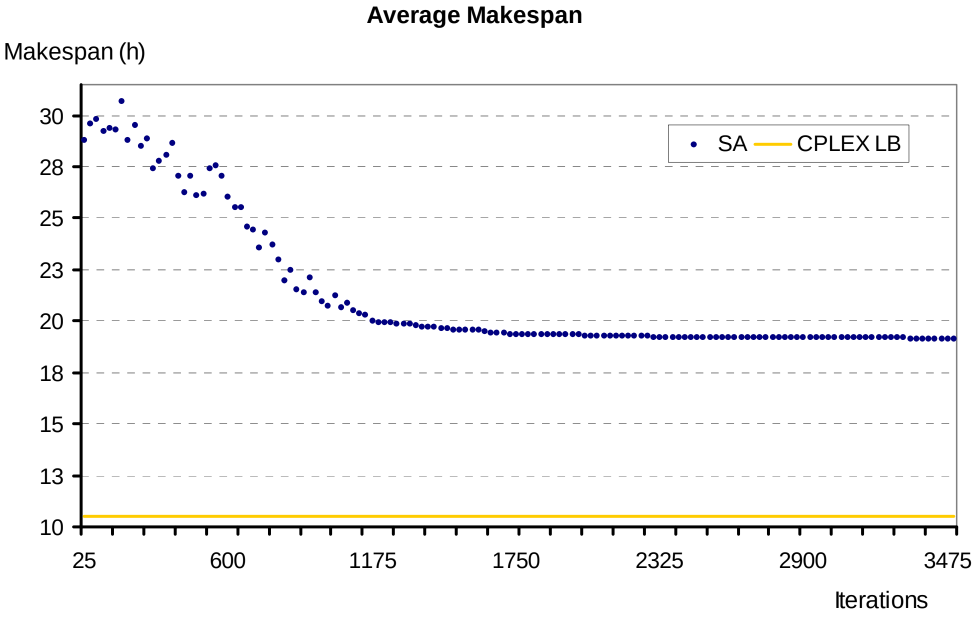


Figure 3.7 - Makespan for hyper-exponential service times

Thus, as the uncertainty of the logistic process grows, the simulation-based optimization procedure becomes the more suitable and challenging solution for representing system dynamics.

Chapter 4
Integration and application of simulation-based optimization models in container terminals

4.1 Introduction

The need to increase the productivity and efficiency of the yard subsystem at the container terminal in Gioia Tauro has, once again, paved the way to seek for OR methods and models yielding more effective, affordable and safer solutions than stand-alone technology and experienced-based advice.

The present case study is another cornerstone on the simulation side of the above paradigm. It follows a previous research and technology transfer project reported in (Canonaco et al. 2007) and based on an enhanced modeling of the integrated logistic process for vessel entrance and berthing at the Gioia Tauro terminal. At that time, the object was to verify whether the entrance channel shared by container vessels and “other traffic” could have become a bottleneck in view of a future increase in containerized traffic. Today, similar concern is attributed to the yard and to providing a model which may hold in store a wide range of useful alternative configurations to be evaluated in order to support major decisions related, but not limited to:

- the size and TEU capacity of yard blocks;
- the types and fleet size of the shuttle vehicles serving each block;
- the purchase of yard cranes for container handling and, eventually,

- the location of the yard crane buffer area.

In this chapter, these and other matters of investigation that affect yard organization and infrastructures are addressed by using a simulation-based optimization approach to pursue the goal of selecting the “best” alternative overall system configuration for greater yard utilization and productivity. In particular, the proposed sequence of problem definition, input modeling, output analysis and final comparison of competing scenarios will be reviewed and discussed in light of the key system features to be modeled in component-like integrated modules and embedded in a custom-made operations simulator.

4.2 The modeling problem

The Medcenter Container Terminal SpA (MCT) in Gioia Tauro is considering for future development a hypothetical change of yard system infrastructures in conjunction with alternative organizational and operational policies and procedures pertaining to this area and all bordering zones.

The container yard is currently operated according to a *direct transfer system* (DTS) based on the use of straddle carriers as equipment for both on-the-yard handling and transportation between the quay and the yard subsystems (for a complete classification based on yard equipment see Steenken et al. 2004).

As for forthcoming projects, the top management intends to prepare and evaluate alternative transfer systems, whether *direct*, *indirect* or *combined*, as well as yard layouts to improve the performance of existing facilities by stages. To this end, site set-ups (e.g. number, general dimensions and locations for all major facilities) and internal connection areas required for organizational and operational purposes will also be taken into account.

A joint academic-industrial modeling effort has been undertaken to fully disclose the advanced knowledge required to provide a “correct” representation of the complex nature of the real yard subsystem. In this sense, if attention is drawn to discharging operations and, thus, container transfer to the yard area for stacking purposes, the corresponding *work cycle* can be represented by the model in Figure 4.1 and described in the following.

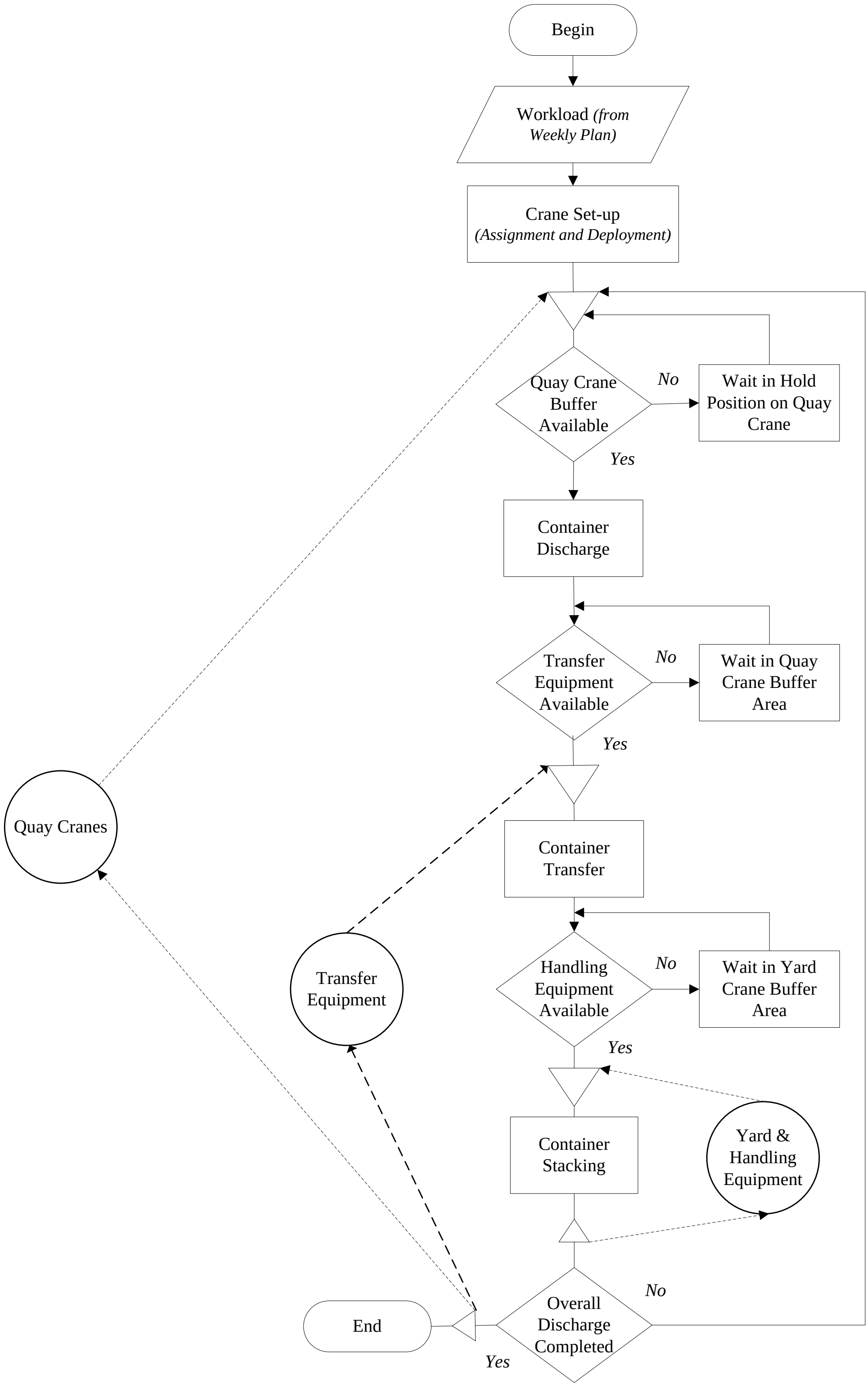

Begin
Workload *(from Weekly Plan)*
Crane Set-up
(Assignment and Deployment)
Quay Crane Buffer Available
No
Wait in Hold Position on Quay Crane
Yes
Container Discharge
Transfer Equipment Available
No
Wait in Quay Crane Buffer Area
Yes
Quay Cranes
Container Transfer
Handling Equipment Available
No
Wait in Yard Crane Buffer Area
Transfer Equipment
Yes
Container Stacking
Yard & Handling Equipment
Overall Discharge Completed
No
End
Yes

Figure 4.1 - Work cycle for container stacking in yard

[Clearly, when considering container loading operations and, thus, retrieval from a yard block and transfer to the quay area, the order of the resource request and acquisition is reversed.]

The *work cycle* under examination is triggered by the container *workload* occurring from discharge (loading) operations on the vessels scheduled for berthing. All parties involved in the modeling stage agree to consider the berth *weekly plan* generated by the CALEMA simulator in (Canonaco et al. 2007) as the natural, initial event source for the yard problem at hand.

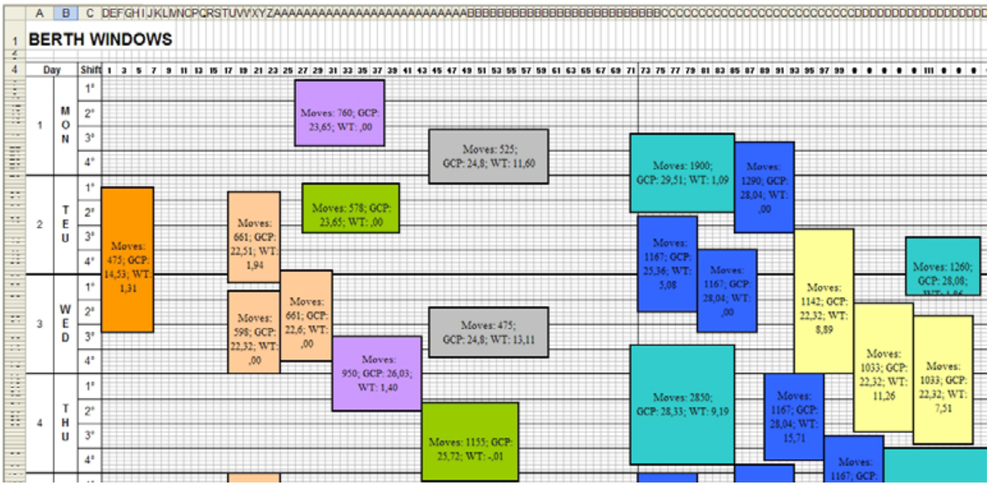

BERTH WINDOWS
Day
Shift
1 MON
2 TEU
3 WED
4 THU
1° 2° 3° 4°
Moves: 760; GCP: 23,65; WT: ,00
Moves: 525; GCP: 24,8; WT: 11,60
Moves: 1900; GCP: 29,51; WT: 1,09
Moves: 1290; GCP: 28,04; WT: ,00
Moves: 475; GCP: 14,53; WT: 1,31
Moves: 661; GCP: 22,51; WT: 1,94
Moves: 578; GCP: 23,65; WT: ,00
Moves: 1167; GCP: 25,36; WT: 5,08
Moves: 1167; GCP: 28,04; WT: ,00
Moves: 1142; GCP: 22,32; WT: 8,89
Moves: 1260; GCP: 28,08;
Moves: 598; GCP: 22,32; WT: ,00
Moves: 661; GCP: 22,6; WT: ,00
Moves: 475; GCP: 24,8; WT: 13,11
Moves: 950; GCP: 26,03; WT: 1,40
Moves: 2850; GCP: 28,33; WT: 9,19
Moves: 1167; GCP: 28,04; WT: 15,71
Moves: 1033; GCP: 22,32; WT: 11,26
Moves: 1033; GCP: 22,32; WT: 7,51
Moves: 1155; GCP: 25,72; WT: -,01
Moves: 1167; GCP:

Figure 4.2 – Example of a weekly plan returned by Calema

Figure 4.2 illustrates an example of this *weekly plan* in a graphical version which, given a fixed time unit (e.g. hour or shift), aids a horizontal reading across the entire berth subsystem. This practical feature provides useful insight for resource planning and management. As a matter of fact, quay cranes in this area are known to operate under a significant degree of parallelism which strongly depends on the intensity of incoming vessels. However, the number of available cranes is not always sufficient to guarantee the completion of discharge/loading operations, for each vessel, within the related expected time of un-berthing. Therefore, when modeling the container yard operations a simulated *weekly plan* can be seen as a feasible starting point and, with respect to the above limitations, it should contain for every vessel-occurrence:

- a *lower bound* on the berthing time: this results from relaxing the constraints on quay crane availability for assignment and deployment. As previously mentioned, in real-life situations a vessel's actual time of berthing and un-berthing (or *berth window*) is heavily constrained by the number and position of the available cranes assigned and deployed for its operations. Thus, a complete or partial lack of or delay in crane availability will lead to postponing the operations for the vessels already berthed and, consequently, for those waiting to be berthed along the same segments;
- the berth position given by the *from-to* bollards that delimit the berth segment of interest;
- the basic physical characteristics such as *length*, meaning the measurement of the vessel extent, and *draft* which quantifies the minimum requirement for vessel berthing in terms of water depth;
- the *service name* which identifies the sequence of ports (AKA *port rotation*) visited by a vessel belonging to that service;
- the number and composition of each cluster of containers "on hold" for discharge (loading) operations, depending on the service of reference.

It is important to remark that, since this *weekly plan* provides only a lower bound on the berthing time, then it should also be coupled with optimization models providing for crane assignment to the berthed vessels and crane deployment along the berth. Examples of integer programming models that well-support these critical decisions are given in (Cordeau et al. 2005, Legato et al. 2008a).

This stated, as soon as a vessel is berthed and properly equipped with human and mechanical resources, container discharge is carried-out by one or more quay cranes that leave the containers in the limited buffer areas (e.g. six single-space slots) located at their feet. Once discharged, a container is picked-up by a shuttle vehicle (e.g. carriers or internal trucks) that provides for transportation to the yard area. If a shuttle vehicle is not immediately available then the container must attend in the buffer until this occurs. Since this area has a limited capacity, should it be full, then the quay crane will have to holdup its discharging activities, causing a congestion phenomenon to rise in the transfer system where functionality is governed by the relationship between equipment speed and container flow.

After the container reaches the yard via shuttle vehicle, it is handled by the yard equipment that serves this area by:

- identifying the slot position in the block (i.e. lane, column and tier);
- performing moving tasks when the slot position is at a ground or intermediate level;
- setting-down the container in the designated position.

To do so, whatever the type of container mover employed (e.g. yard cranes, straddle carriers, reach stackers), it must be available for container stacking (retrieval) and have available operation space as well (e.g. the transfer lane along the side of a yard block used by rail mounted gantry cranes as handover point).

The entire *work cycle* described above is based on a sequence of seize-delay-release actions stemming from container flow between bordering terminal areas: further details and options referring to the processes herein involved (i.e. container discharge/loading, container transfer and container stacking/retrieval) are given later in the operational features section. In any case, the cycle continues looping until all containers have been stacked in the yard (loaded on the vessel).

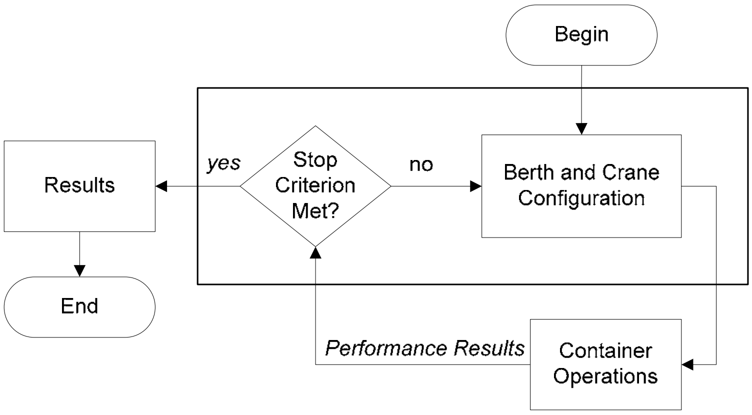


Figure 4.3 - The simulation-based optimization scheme

From a methodological point of view, at the most outer level the simulation-based optimization approach adopted in this problem is fed by the berth *weekly plan* along with the crane set-up. This means that, given an initial berth-crane configuration, the S&O model runs a precise scenario in which yard layout and transfer and stacking systems are already fixed, as depicted by the scheme in Figure 4.3. However, the simulation-based optimization can also be extended to inner levels. For instance:

- with preset container transfer and handling equipment, various yard layouts can be evaluated by varying the number of yard blocks, their position (e.g. vertical, horizontal) or their capacity;
- for a given yard layout and container handling solution, alternative transfer systems can be tested (e.g. direct, indirect and combined);
- once the transfer system and the yard layout are defined, different container handling equipment and services can be explored (e.g. rail mounted grantry cranes with side lane rather than front buffer).

Clearly, in the above examples, the outcome and output from inner-level configuration set-ups will in turn affect each other (e.g. in the first case, different yard layouts impact on the service times of transfer and handling equipment) and, ultimately, the “outer” berth-crane configuration (e.g. berth windows could require tuning adjustments). Thus, the problem solution is not straightforward and an iterative structural simulation-based optimization should be exploited.

4.3 Input modeling

Input modeling is a major task from the standpoint of time and resource requirement common to many different disciplines. Since the input provides the driving force for running the model, one must always bear in mind that if the input data is inaccurately collected, inappropriately analyzed, or not representative of the environment, the output data could be misleading and possibly damaging or costly when used for policy or decision making (Banks et al. 2001).

In the approach considered herein, the data model must be developed from two complementary directions: on one hand, in a much broader sense, data characterization is required to model communication, coordination and cooperation rules, regulations and practices adopted for business organization and/or resource management; on the other, in terms of a more strict interpretation, numerical data is collected and used to quantify resource utilization according to the workload generated by applications/users trying to carry-out a sequence of requests throughout the system. In the former case, to both enhance the quality of the resulting model and increase the confidence of using the model in scenario analysis, data should be obtained in close cooperation with company managers, analysts and end-users. In the latter, data is always obtained by means of statistical analysis on the past and/or expected system behavior.

4.3.1 Operational features

Many different company-based rules, regulations and practices are widely used across the subsystems of the container terminal under examination and, thus, call for representation. In particular, attention is drawn to organizational and operational issues required to define the yard layout and manage the yard activity with respect to policies and equipment employed for container stacking/retrieval, respectively.

As far as yard layout is concerned, block location, size and TEU capacity are the most common three degrees of freedom provided to the user to model this limited resource in a variety of ways. An example of two alternative yard layouts is given in Figure 4.4.

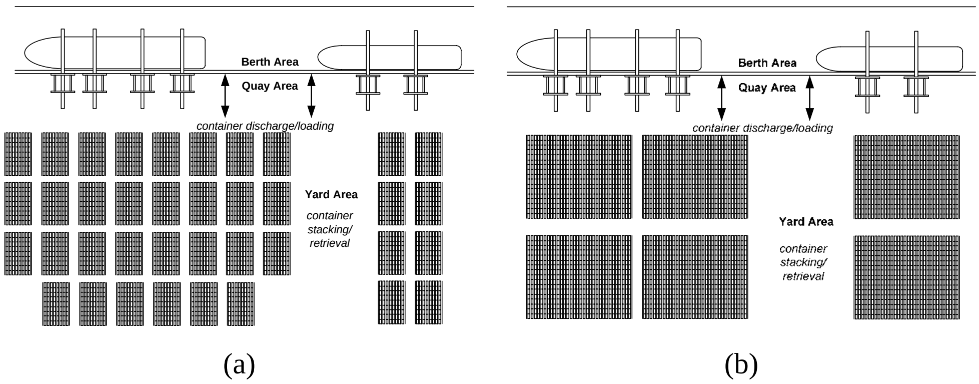

Berth Area
Quay Area
container discharge/loading
Yard Area
container stacking/ retrieval
(a)
Berth Area
Quay Area
container discharge/loading
Yard Area
container stacking/ retrieval
(b)

Figure 4.4 - Two alternative yard layouts

It is worth observing that both the number of blocks and the TEU capacity of each block in a given yard layout affect the average travel time of shuttle vehicles cycling between the quay and the yard areas, as well as the container handling time on the yard. For instance, in the yard configuration depicted by Figure 4.4.(a), the average distance to be covered in order to reach a container is greater than the average distance deriving from the solution portrayed in Figure 4.4.(b). On the other hand, more container handling equipment can be concentrated in a specific area in the former case, thus returning a smaller service time, whereas this possibility is prevented in the latter case due to potential interference between container movers meant to operate on adjacent yard bays.

As for container stacking (retrieval) policies, storage strategies are extremely critical for the management of containers on many levels. For example, the higher the tier, the greater the saving on ground space; then again, as stack height grows, the number of reshuffles/rehandles required to reach a specific container grow as well. Only after the appropriate stack layout has been determined, consideration is granted to another major question concerning the identification and assignment of a container storage location. An experience-based strategy usually stores containers in groups or stacks according to some basic attributes such as:

- length (e.g. 20’, 40’ and 45’);
- height (e.g. standard, high-cube);
- weight class (e.g. heavy and light);

- type (e.g. reefer, IMO-class-x);
- out-of-gauge (e.g. top-OOG, side-OOG and front-OOG);
- loading vessel;
- port of destination.

The benefits of this approach are evident when considering, for example, a group of containers scheduled for loading on the same vessel. If they are in the same stack, then the order in which the containers are transferred and loaded is irrelevant and, most of all, the performance of on-the-yard handling equipment is improved since this *homogenous stacking policy* guarantees that such a container is always located at the top position of a stack and, thus, nonproductive movements of the above equipment can be reduced. In addition, if these stacks are also dispersed throughout the yard blocks, then during vessel loading, each quay crane working is sure to be supported by several container movers on the yard and, thus, quay crane starvation is likely not to occur. Container dispersion will also contribute in balancing each yard block’s workload at any given point in time.

In most cases, these and other practical matters are left to the providers of advanced ICT solutions which support the yard planning process by implementing user-defined planning parameters and taking into account the already existing procedures on the terminal.

The ultimate decision affecting yard organization is related to the type of equipment involved in the container cycle. In particular, within the constraints of the available budget, different solutions can be built around the choice of the transfer system and the on-the-yard handling equipment, along with the terms of employment and management for both categories. Some of the possible options that can be explored are:

- single modality systems (e.g. sole employment of straddle carriers for container transportation and slot positioning, set-down and pick-up);
- mixed modality systems (e.g. combined use of straddle carriers and rail mounted gantry cranes or a mixed combination of straddle carriers, trailers and rail mounted gantry cranes);

- buffer areas for container movement on the yard blocks (e.g. definition and localization of front or lateral areas including width for truck and/or rail-car access).

Clearly, once the above settings and the stack height are defined, the TEU capacity of each yard block will be defined as well.

4.3.2 Statistical data issues

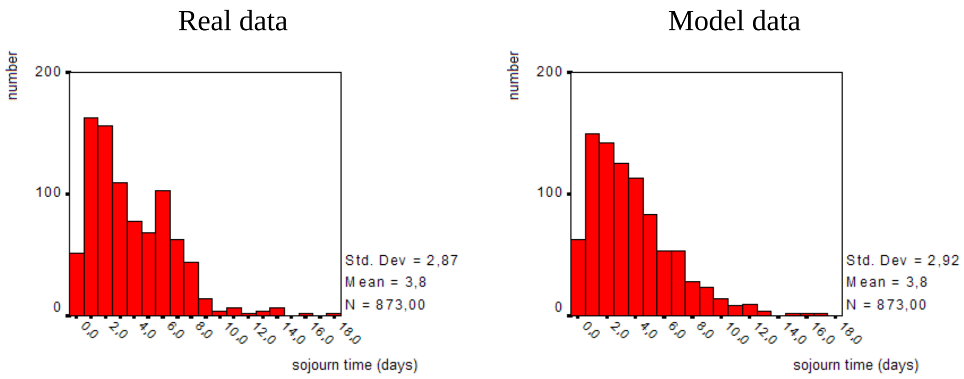
When dealing with numerical values, specific statistical models are used to collect, summarize and, thus, represent the data referring to the input process of interest.

If data is available, it is used to identify a probability distribution (or distribution family) that best describes the probabilistic behavior of the input process. With reference to the objectives of the present simulation study, the *work cycle* illustrated in Figure 4.1 calls for the following data collection:

- composition and routing of container clusters flowing between “discharge/loading points”, “transshipment points” and “service points in the corresponding yard area”;
- service time of the container transfer vehicle(s);
- service time of the on-the-yard container handling equipment;
- other operational features (e.g. failures, stops, maintenance, lead times, etc.).

When data is not available, then system knowledge, experience and “educated guesses” usually step forward to fill-in the related gap. By example, one may think of simulating a scenario based on the employment of specific equipment which is currently not on the premises, but placed under evaluation for future purchase as for rail mounted gantry cranes meant for yard operations at the port of Gioia Tauro. Clearly, in this case no quantitative, nor qualitative data is present in the company records. As a consequence, configuration and operation settings must occur according to the individual know-how of experts working across the company or by outsourcing to partner terminals already using similar technologies.

Whatever the case, once information on data variance and data skewness is available, then a tuning phase of the corresponding input models will follow. In this sense, the flexibility of the probabilistic models can be better guaranteed by using a *mixture distribution*-based methodology (Titterington et al. 1985) in which the density of a single mixture distribution, also called compound distribution, can be expressed as a weighted sum of the component distributions. Previous experience related to modeling yard operations in a container terminal (Legato et al. 2000) has proved the above methodology to be successful (see Figure 4.5).



(a) (b)

Figure 4.5 - Real data compared to model data for on-the-yard container sojourn time

In particular, the following mixture distribution has already been used to represent the container sojourn time in a specific yard area of the terminal managed by MCT S.p.A.:

$$F(x) \hat{=} \alpha F_1(x) + (1-\alpha)F_2(x), \qquad 0 \le \alpha \le 1 \tag{4.1}$$

where

$$F_1(x) = 1 - \exp(-\mu x) \tag{4.2}$$

is an exponential distribution with rate $\mu = 1/3.8$ and

$$F_2(x) = 1 - \exp(-n\lambda\, x)\sum_{i=1}^{n-1}\frac{(n\lambda\, x)^i}{i!} \qquad (4.3)$$

is a two-stage erlang distribution ($n = 2$) with rate $\lambda = 1/3.8$.

The *goodness-of-fit* of distribution (4.1) and the associated parameters has been formally evaluated via the Kruskal-Wallis statistical test. This method has been applied to test the null hypothesis that the two different samples under comparison - the real data in Figure 4.5.(a) and the model data in Figure 4.5.(b) generated via Monte Carlo simulation - have been drawn from the same distribution.

4.4 Development and use of the simulation model

The design and implementation of the simulation model is bound to be carried-out in compliance with the conventional steps used to guide a thorough and sound simulation study. Figure 4.6 provides a high-level view of the modeling process, but other sources and more detailed discussions can be found in (Banks et al. 2001, Law and Kelton 2000).

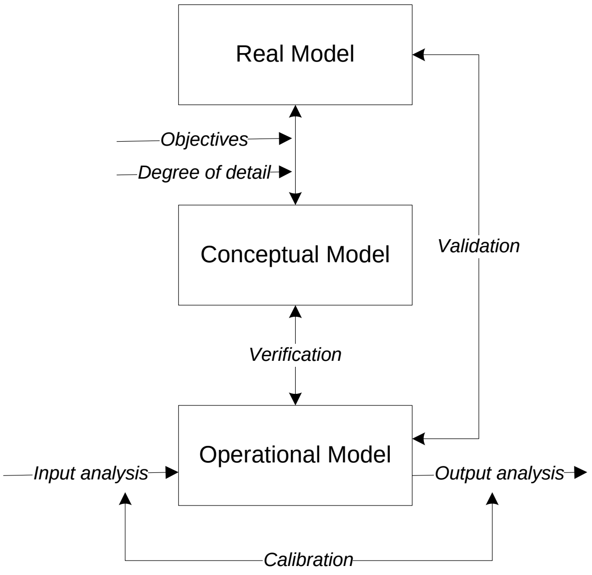


Figure 4.6 - Modeling logic in a simulation study

As one may observe, the primary function of the *conceptual model* is to bridge the troublesome gap between the *real model* and the *operational model*.

The conceptual model to develop depends on the objectives of the study, the complexity of the issues being analyzed and the necessary degree of detail required in system representation, but, in general, it can be structured in a network model form in which every single object has a more or less complex nature. Starting from the graphs used in Operations Research to model network flows and simple queuing systems with one or more servers and one or more buffers (whether dedicated or not), more complex, perhaps hierarchically-organized and powerful data structure-based objects can be envisioned and attained. In such a case, the model complexity can be better managed by making use of a special programming language. This may appear to be even more likely if one thinks of having to model the container stacking area in a container terminal.

Once the conceptual model is addressed and implemented, the operational model can be simulated to generate the trajectory followed by the state variables and, therefore, carry-out simulation experiments with the objective of performing output analysis by means of statistical models. The first experiments will always serve the purpose of establishing the overall credibility of the simulator! In this sense, two most direct and intuitively accessible measures of simulation credibility are given by *verification* and *validation*. Both are based on the replication of experiments followed by output analysis, but, on a conceptual level, they are to remain distinguished from each other: verification is concerned with the correspondence between the input parameters and logical structure of the conceptual model and the operational model, while validation refers to determining if the real system is accurately represented by the operational model (in terms of system operation rules and data). Although verification and validation are placed in a specific pattern and order in Figure 4.6, they are to be considered both iterative and continuously repeated processes as the design and use of the model progresses.

Verification is made possible by the application of a wide range of techniques than can be grouped in the following classes (Banks et al. 2001, Carson 2002):

- *common-sense techniques*, used in developing any sort of software project, suggest to closely examine the model output for reasonableness under a variety of settings of the input parameters in order to aid the detection of mistakes in model logic and data misspecifications. For example, model reasonableness can be evaluated by observing the values returned for certain output indices (e.g. waiting time) as certain input parameters expected to have correlation with the former vary (e.g. service time);
- *thorough documentation* ranges from providing brief comments and definitions for all variables and parameters to describing each major section to clarify the logic of a model and allow (the modeler or someone else) to verify its completeness;
- *tracing* produces a detailed printout of the state of the simulation model as it changes over time; it usually covers rare events, specific locations and particular conditions.

As far as validation is concerned, a three-step approach suggested by (Naylor and Finger 1967) consists in first building a model that has high *face validity*. This requires user involvement for major insight on both system structure and reliable data, followed by sensitivity analysis in which efforts can be addressed to monitor whether the model behaves as expected when input changes.

The second step is reserved to test major assumptions in terms of model structure and data. In the former case, observation and discussion with key figures of the system are critical activities; in the latter, random samples are used to identify the appropriate probability distribution, estimate the related parameters and validate the assumed statistical model by a goodness-of-fit test.

In the end, model input-output transformations are compared to the corresponding input-output transformations for the real system with reference to the entire system (for existing systems) or subsystems (for nonexistent systems).

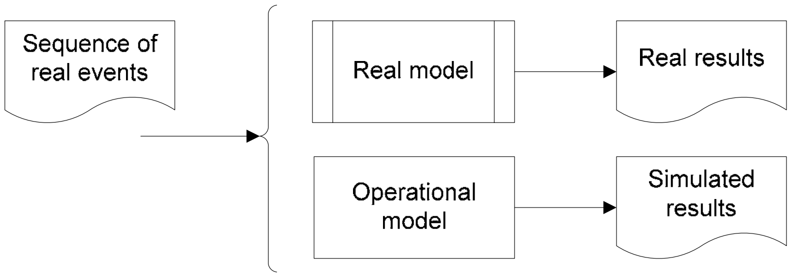


Figure 4.7 - Modeling logic in a simulation study

This final stage of the study is depicted in Figure 4.7. For this purpose, once again, statistical tests can be applied for making quantitative decisions about a given process or processes: for example, a typical point of inquiry aims at investigating whether there is enough evidence to “reject” the hypothesis that the mean value of the simulated results is equal to the mean value obtained from real data. The iterative process according to which real system data is compared with model data in order to reduce the estimated differences is called *calibration* of the simulation model.

4.5 Output analysis

The design and simulation of a set of highly relevant, yet quite dissimilar scenarios has a sole purpose: obtaining estimates of given performance measures under a variety of conditions. In the study case at hand, plausible scenario instances are generated by coupling alternative transfer systems, whether *direct*, *indirect* or *combined*, with different yard layouts to improve the average value of the following performance measures:

- yard occupation;
- waiting time per equipment type;

- productivity (i.e. throughput) per equipment type with particular reference to the *GCP - gross crane productivity* (AKA *GCR - gross crane rate*), one of the most important performance measures in a container terminal.

Economically speaking, the “predictions” returned by the simulation study must favor cost-effective decision making, meaning that the total cost required to pursue and obtain facility improvements, in terms of the above indices, must be less than savings resulting from the greater service quality achieved over the period of time considered. On the other hand, the optimal economic solution must lie within the boundaries of technical feasibility in order to prevent terminal subsystems from making little or no progress due to critical resources that become exhausted or too limited to perform needed operations.

To make this clearer, one may wish to investigate via simulation the extent to which cost-effective operational set-ups serve as a powerful strategy for improving *GCP*. In (Petering and Murty 2008), a first set of preliminary experiments is designed around a similar objective: two different policies for yard crane deployment are proposed within both a large and small terminal to show how *GCP* depends on the yard block length. In their empirical tests across a certain number of artificially generated scenarios, the concavity of *GCP* seems to hold with respect to block length as shown, for example, in Figure 4.8.

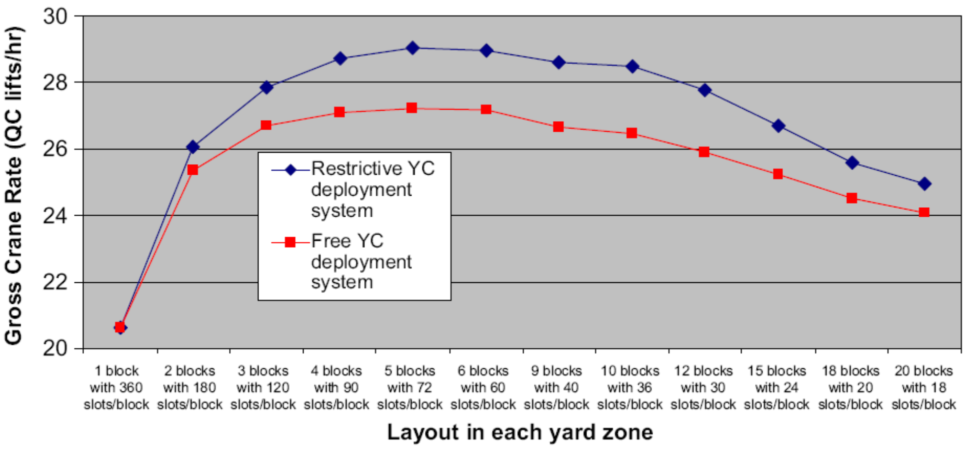


Figure 4.8 - GCP point estimation for different block length

Without loss of generality, the same *GCP* investigation can be conducted with reference to

- volume of container traffic;
- type of transfer system;
- number of active units in the transfer system;
- type of on-the-yard handling system;
- number of active units in the on-the-yard handling system;
- location and area size devoted to internal connections and buffers

in order to detect threshold values beyond which the growth of crane marginal productivity is low or most likely to crash (e.g. the *thrashing phenomenon* that arises when a great number of active cranes interfere with one another's operations).

In the attempt to provide a more "robust" answer to these and other specific *GCP* evaluations, an effective method for interval estimates of average *GCP* or average point estimates followed by variance estimation is strongly demanded. In particular, the choice of what method to apply must be made considering that the most usual form of policy evaluation is based on an extensive empirical investigation of "simulation outcomes" with the aim of obtaining a representative reading of the model behavior. Consequently, different alternative configurations of the system of interest need to be compared on the basis of the long-run average values of *GCP* under the goal of detecting significant differences among the configurations and selecting the best one, or a near-best, with a user-specified probability of correct selection.

Unfortunately, as discussed in section 2.2, any available statistical technique has proven performance for the asymptotic case in which the simulation output process is stationary and the observations are independent and identically distributed data from a normal distribution. Thus, the problem to deal with lies on the way of batching output observations and on the empirical evaluation of the properties of any batch-based estimator of the asymptotic variance constant of the simulation output processes. Clearly, this is at the basis of pursuing the true probability of correct selection of the adopted R&S procedure.

In this sense, personal experience (see Canonaco et al. 2007 for details) in generating confidence intervals and, thus, estimating the sample mean and sample variance of long-term based performance indices in container terminal simulations, confirms the substantial robustness of the coverage properties with respect to the different ways of arranging observations into batches and distributing them into one or multiple replications. In particular, experimented options have included:

- 30 batches taken from a unique replication (classical batch means method) and used to compute both the sample mean and the sample variance;
- 15 batches taken from the same replication to estimate the sample mean and 15 additional batches taken from another 15 replications to obtain an independent estimate of the sample variance;
- 1 batch per replication (classical independent replication method).

The 30-batch case study revealed that correlation among batches is not expected to generate significant errors when producing interval estimates for an average performance measure, due to the ergodic property of the underlying stochastic process. This property guarantees a non-biased estimate of the sample mean accompanied by a sufficiently small sample variance estimate bearing an error (due to the unavoidable correlation among batches) which becomes irrelevant.

Vice versa, the other border case in which a single batch is taken from a single replication seems to be jeopardized by the opposite possibility of incurring in a significant error on the estimation of the sample mean, while obtaining a good estimate of the sample variance, due to the absence of correlation among batches.

Finally, the intermediate case of taking all 15 batches from the same replication was designed to combine the benefit of ergodicity on the sample mean estimate with the need to avoid a significant correlation error on the sample variance thanks to the additional 15 independent replications.

Indeed, the strength of such considerations leads to choosing the first option which, however, could result in a computationally intensive activity if the batch normality and non-correlation assumptions are to be granted. In view of the fact this effort has to be repeated at each iteration of a simulation-based optimization procedure, the overall computational burden is likely to be unaffordable. To this end, a practical approach consists in working with small batch sizes as long as their associated batch means belong to a distribution shape with a sufficiently limited skewness. The rationality of this recommendation is explained by the following.

Let $\overline{X}_i(k)$ be the sample mean of the k observations in batch i (not normally distributed if k is relatively small) and consider the biased sample variance estimator:

$$\hat{\sigma}^2(n) = \frac{1}{n}\sum_{i=1}^{n}\left(\overline{X}_i(k) - \overline{\overline{X}}(n)\right)^2. \qquad (4.4)$$

According to the so-called Edgeworth expansion,

$$P\left\{\sqrt{n}\cdot\left(\frac{\overline{\overline{X}}(n)-\mu}{\hat{\sigma}(n)}\right) \le x\right\} =$$

$$= P\{N(0,1) \le x\} + \frac{E\left[\left(\overline{X}(k)-\mu\right)^3\right]}{6\sigma^3\sqrt{n}}\left(2x^2+1\right)\frac{e^{-x^2/2}}{\sqrt{2\pi}} + O(1/n) \qquad (4.5)$$

where $O(1/n)$ is a term that basically looks like a constant C divided by n (Hall 1987). Observe that the error in the normality assumption is basically described by the second term of the right-side member and, in particular, by the skewness $\gamma = \frac{E\left[\left(\overline{X}(k)-\mu\right)^3\right]}{\sigma^3}$ of the unknown distribution of $\overline{X}(k)$, due to the fact that the number of batches (n) appears under square root. Thus, k becomes the parameter by means of which one can control both the error upon the normality assumption and the computational burden in output analysis and, to this purpose, it should be dimensioned just large enough to size-down the skewness.

In conclusion, the more symmetric the shape of the batch mean is, the better the CLT approximation holds and the more accurate the sample variance approximation to the process variance is.

Conclusions

Queuing network models have been developed and applied in the planning and management of logistic resources and processes at a pure transshipment container terminal located in Southern Italy throughout the entire duration of this thesis. Solution of these models by discrete-event simulation has been discussed and experienced both within a practical "what-if" approach and a theoretical "what-to" approach to the optimization of port logistics by simulation. An example of the former approach is given by the simulator for managing vessel entrance and berth assignment; an example of the latter approach is provided by the simulator for assigning holds to cranes and scheduling the discharge/loading operations. A further example arises with the queuing network model of the yard organization and yard-crane deployment, where a relatively small number of configurations and policies should be simulated and compared. Hence, this thesis has focused on statistical techniques for ranking and selection of the best in all of the above examples, followed by the integration of these techniques with an algorithm for best solution search which behaves as a homogeneous Markov chain.

The computational burden of the search process has been considered with respect to the problem of establishing how many simulation experiments should be carried out to guarantee the fixed probability of correct selection, provided that the output of simulation is, usually, a sequence of correlated measures. Numerical results seem to confirm the goodness of the proposed estimator for the sample mean of the simulation output process, as well as the goodness of the idea of modulating the number of simulation runs in dependence on the behavior of the sample variance.

A second methodological contribution comes from the analysis of the convergence proofs of the simulated annealing-based search process experienced on the simulator for container discharge/loading operations. In particular, two modifications have been proposed and remain under current investigation: one regards the step of selecting the candidate solution to be compared to the current solution; the other regards the step of comparing the estimate for the objective function of the candidate solution against the estimate of the current solution. The former modification is based on selecting as candidate solution the best out of a suitable subset of solutions in the neighborhood of the current one; the latter modification uses an interval estimate for the difference between the sample mean related to the objective function evaluation of the candidate solution and the sample mean related to the current solution.

Acknowledgements

It is time to thank everyone deserving of such praise for supporting this journey through ideas, suggestions and comments. The list is long and I am likely to omit someone. Thus, I prefer just to mention the one person who has been sharing with me, during this three-year period of my "renewal", his profound disciplinary basis, clear theoretical orientation, time and experience. I am grateful for receiving more than I gave during this educational and professional opportunity in the flagship area of Operations Research.
Thank you Pasquale Legato.

References

Aarts, E.H.L. and P.J.M. van Laarhoven. 1987. *Simulated Annealing: Theory and Applications (Mathematics and Its Applications)*. Kluwer Academic Publishers, The Netherlands.

Ahmed, M.A. and T.M. Alkhamis. 2002. Simulation-based optimization using simulated annealing with ranking and selection. *Computers & Operations Research* 29: 387-402.

Ahmed, M.A. and T.M. Alkhamis. 2004. Simulation-based optimization using simulated annealing with confidence interval. In *Proceedings of the 2004 Winter Simulation Conference*, R.G. Ingalls, M.D. Rossetti, J.S. Smith, and B.A. Peters, eds., 514-519.

Alrefaei, M.H. and S. Andradóttir. 1999. A simulated annealing algorithm with constant temperature for discrete stochastic optimization. *Management Science* 45(5): 748-764.

Andradottir, S. 1995. A method for discrete stochastic optimization. *Management Science* 41(12): 1946-1961.

Banks, J., J.S. Carson, B.L. Nelson and D.M. Nicol. 2001. *Discrete-event system simulation*. Third edition. Prentice-Hall, Upper Saddle River, NJ.

Bartlett, M.S.. 1937. Properties of sufficiency and statistical tests. *Proceedings of the Royal Statistical Society Series* A 160: 268-282.

Bechhofer, R.E.. 1954. A single-sample multiple decision procedure for ranking means of normal populations with known variances. *Annals of Mathematical Statistics* 25(1): 16-39.

Bechhofer, R.E., T.J. Santner and D.M. Goldsman. 1995. *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*. John Wiley, New York.

Bhat, U.N. 1984. *Elements of applied stochastic processes*. Second Edition. John Wiley & Sons, New York.

Billingsley, P.. 1995. *Probability and Measure*. Third Edition. John Wiley & Sons, Inc..

Bulgak, A.A. and J. L. Sanders. 1988. Integrating a modified simulated annealing algorithm with the simulation of a manufacturing system to optimize buffer sizes in automatic assembly systems. In *Proceedings of the 1988 Winter Simulation Conference,* M. Abrams, P. Haigh and J. Comfort, eds., 684-690.

Canonaco, P., P. Legato and R.M. Mazza. 2007. An Integrated Simulation Model for Channel Contention and Berth Management at a Maritime Container Terminal. In *Proceedings 21st European Conference on Modelling and Simulation,* I. Zelinka, Z. Oplatková and A. Orsoni, eds., 353-362.

Canonaco, P. P. Legato and R.M. Mazza. 2009. Yard crane management by simulation and optimization. *Maritime Economics and Logistics* special issue on *OR Methods in Maritime Transport and Freight Logistics*, doi:10.1057/mel.2008.

Canonaco, P., P. Legato, R.M. Mazza and R. Musmanno. 2008. A queuing network model for the management of berth crane operations. *Computers and Operations Research* 35, 2432–2446.

Carson, J.S. II. 2002. Model Verification and Validation. *In Proceedings of the 2002 Winter Simulation Conference,* E. Yücesan, C.-H. Chen, J.L. Snowdon, and J.M. Charnes, eds., 52-58.

Chen, E.J. and W.D. Kelton. 2005. Sequential selection procedures: using sample means to improve efficiency. *European Journal of Operational Research* 166: 133-153.

Cheung, R.K., C.-L. Li and W. Lin. 2002. Interblock crane deployment in container terminals. *Transportation Science* 36, 1(21), 79-93.

Cordeau, J.F., G. Laporte, P. Legato and L. Moccia. 2005. Models and tabu search heuristics for the berth-allocation problem. *Transportation Science* 39(4): 526-538.

Daganzo, C.F.. 1989. The crane scheduling problem. *Transport Research, Part B* 23(3), 159–175.

Damerdji, H.. 1994. Strong consistency of the variance estimator in steady-state simulation output analysis. *Mathematics of Operations Research* 19: 494-512.

Dudewicz, E.J. and S.R. Dalal. 1975. Allocation of observations in ranking and selection with unequal variances. *Sankhya* B7: 28-78.

Fox, B.L. and G.W. Heine. 1995. Probabilistic search with overrides. *The Annals of Applied Probability* 5(4) 1087-1094.

Fu, M.C. 2001. Simulation optimization. In *Proceedings of the 2001 Winter Simulation Conference*, B.A. Peters, J.S. Smith, D.J. Medeiros and M.W. Rohrer, eds., 53-61.

Fu, M. and B. Nelson. 2003. Guest Editorial. *ACM Transactions on Modeling and Computer Simulation* 13(2), 105–107.

Gambardella, L.M., A.E. Rizzoli and M. Zaffalon. 1998. Simulation and planning of an intermodal container terminal. *Simulation* 71(2), 107-116.

Gelfand, S.B. and S.K. Mitter. 1989. Simulated annealing with noisy or imprecise energy measurements. *Journal of Optimization Theory and Applications* 62: 49-62.

Geman, S. and D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6(6) 721-741.

Gibbons, J.D., I. Olkin and M. Sobel. 1979. An introduction to ranking and selection. *The American Statistician* 33(4): 185-195.

Glynn, P.W. and W. Whitt. 1991. Estimating the asymptotic variance with batch means. *Operations Research Letters* 10: 431-435.

Goldsman, D.M., M.S. Meketon and L.W. Schruben. 1990. Properties of standardized time series weighted area variance estimators. *Management Science* 36: 602-612.

Goldsman, D.M., S.-H. Kim, W.S. Marshall and B.L. Nelson. 2002. Ranking and selection for steady-state simulation: procedures and perspectives. *INFORMS Journal on Computing* 14(1): 2-19.

Gross, D. and C.M. Harris. 1998. *Fundamentals of queueing theory*. Third Edition. John Wiley & Sons, Inc..

Gupta, S.S.. 1965. On some multiple decision (ranking and selection) rules. *Technometrics* 7: 225-245.

Gutjahr, W.J. and G.Ch. Pflug. 1996. Simulated annealing for noisy cost functions. *Journal of Global Optimization* 8: 1-13.

Haddock, J. and J. Mittenthal. 1992. Simulation optimization using simulated annealing. *Computers and Industrial Engineering* 22 387-395.

Hajek, B. 1988. Cooling schedules for optimal annealing. *Mathematics of Operations Research* 13(2), 311-329.

Hall, P. 1987. Edgeworth expansion for Student's t statistic under minimal moment conditions. *The Annals of Probability*, 15(3): 920-931.

Heyman, D.P. and M.J. Sobel. 1982. *Stochastic models in operations research*. Volume 1. McGraw-Hill, New York.

Hogg, R.V. and A.T. Craig. 1978. *Introduction to mathematical statistics*. Fourth Edition. Macmillan Publishing Co., Inc., New York.

Hong, L.J. and B.L. Nelson. 2007. Selecting the best system when systems are revealed sequentially. *IIE Transactions* 39.

Ingber, L.. 1993. Simulated annealing: practice versus theory. *Mathematical Computer Modelling* 18 (11): 29-57.

Kim, K.H. and K.C. Moon. 2003. Berth scheduling by simulated annealing. *Transportation Research Part B* 37 541–560.

Kim, K.H. and Y.M. Park. 2004. A crane scheduling method for port container terminals. *European Journal of Operations Research* 156(3), 752–768.

Kim, S.-H. and B.L. Nelson. 2001. A fully sequential procedure for indifference-zone selection in simulation. *ACM TOMACS* 11: 251-273.

Kirkpatrick, S., C.D. Gelatt Jr and M.P. Vecchi. 1983. Optimization by simulated annealing. *Science* 221: 671-80.

Law, A.M. and W.D. Kelton. 2000. *Simulation modeling and analysis*. Third Edition. McGraw-Hill, New York.

Legato, P., D. Gullì and R. Trunfio. 2008a. Assignment and deployment of quay cranes at a maritime container terminal. In *Proceedings of the 11th International Workshop on Harbor, Maritime & Multimodal Logistics Modeling and Simulation,* A. Bruzzone, F. Longo, Y. Merkuriev, G. Mirabelli and M.A. Piera, eds.. Campora S. Giovanni (CS), Italy, 214-220.

Legato, P., M. Fortino and F. Reitano. 2000. Simulazione dei processi di stoccaggio e movimentazione dei container presso il terminal di Gioia Tauro. In Proceedings of *Metodi e Tecnologie dell'Ingegneria dei Trasporti - Seminario 2000,* G. Cantarella and F. Russo, eds.. Franco Angeli, Milano, 85-100.

Legato, P. and R.M. Mazza. 2001. Berth planning and resources optimisation at a container terminal via discrete event simulation. *European Journal of Operational Research* 133, 537-547.

Legato, P. and R.M. Mazza. 2007. Progettazione di procedure di ottimizzazione mediante simulazione per il problema del crane scheduling. *Technical Report RT-OR1-IMPR-CS1/5*, Consorzio R&D.LOG – Logistica Ricerca e Sviluppo.

Legato, P., R.M. Mazza and R. Trunfio. 2008b. Simulation-based optimization for the quay crane scheduling problem. In *Proceedings of the 2008 Winter Simulation Conference*, S.J. Mason, R. Hill, L. Moench, and O. Rose, eds., 2717-2725.

Lim, A., B. Rodrigues and Y. Zhu. 2002. Crane scheduling using squeaky wheel optimization with local search. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning*, Singapore.

Lim, A., B. Rodrigues and Z. Xu. 2007. A m-parallel crane scheduling problem with a non-crossing constraint. *Naval Research Logistics* 54(2), 115–235.

Linn, R., J. Liu, Y. Wan, C. Zhang and K.G. Murty. 2003. Rubber tired gantry crane deployment for container yard operation. *Computers & Industrial Engineering* 45, 429-442.

Liu, C.L. 1968. *Introduction to combinatorial mathematics*. McGraw-Hill, New York.

Kim, K.H., S.-J. Lee, Y.-M. Park, C.H. Yang and J.-W. Bae. 2006. Dispatching yard cranes in port container terminals. *TRB – Transportation Research Board Annual Meeting*.

Meketon, M.S. and B. Schmeiser. 1984. Overlapping batch means: something for nothing? In *Proceedings of the 1984 Winter Simulation Conference*, 227-230.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller. 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics* 21(6): 1087-1092.

Microsoft® Excel 2002, Copyright© Microsoft Corporation 1985-2001.

Microsoft Visual Basic 6.0 Professional Version, Copyright© 1991-1998 Microsoft Corporation.

Milton, J.S. and J.C. Arnold. 1986. *Probability and statistics in the engineering and computing sciences*. McGraw-Hill, New York.

Nance, R.E. and R.G. Sargent. 2002. Perspectives on the evolution of simulation. *Operations Research* 50(1): 161-172.

Naylor, T.H. and J.M. Finger. 1967. Verification of Computer Simulation Models. *Management Science* 2: B92-B101.

Nelson, B.L., J. Swann, D. Goldsman and W.-M. Song. 2001. Simple procedures for selecting the best system when the number of alternatives is large. *Operations Research* 49: 950-963.

Petering, M.E.H. and K.G. Murty. 2008. Effect of block length and yard crane deployment systems on overall performance at a seaport container transshipment terminal. *Computers and Operations Research* doi: 10.1016/j.cor.2008.04.007.

Prudius, A.A.. 2007. *Adaptive random search methods for simulation optimization*. Ph.D. thesis, H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA. Available via <http://etd.gatech.edu/theses/available/etd-06252007-161154> [accessed April 10, 2008].

Prudius, A.A. and S. Andradóttir. 2004. Simulation optimization using balanced explorative and exploitative search. In *Proceedings of the 2004 Winter Simulation Conference*, R.G. Ingalls, M.D. Rossetti, J.S. Smith, and B.A. Peters eds., 545–549.

Prudius, A.A. and S. Andradóttir. 2005. Two simulated annealing algorithms for noisy objective functions. In *Proceedings of the 2005 Winter Simulation Conference*, M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Joines, eds., 797-802.

Rinott, Y. 1978. On two-stage selection procedures and related probability-inequalities. *Communications in Statistics - Theory and Methods* A7(8) 799-811.

Roenko, N. 1990. Simulated annealing under uncertainty. *Technical Report*, Institute for Operations Research, University of Zurich.

Rockwell Arena - Version 11.00.00 CPR 7, Copyright© 1983-2006 Rockwell Automation Incorporated.

Sammarra, M., J.-F. Cordeau, G. Laporte and M.F. Monaco. 2007. A tabu search heuristic for the quay crane scheduling problem. *Journal of Scheduling* 10, 327-336.

Schruben, L.W.. 2000. Mathematical programming models of discrete event system dynamics. In *Proceedings of the 2000 Winter Simulation Conference*, J.A. Joines, R.R. Barton, K. Kang and P.A. Fishwick, eds., 381-385.

Shabayek, A.A. and W.W. Yeung. 2002. A simulation model for the Kwai Chung container terminals in Hong Kong. *European Journal of Operational Research* 140, 1-11.

Silberholz, M.B., B.L. Golden and E.K. Baker. 1991. Using simulation to study the impact of work rules on productivity at marine container terminals. *Computers & Operations Research* 18(5), 443-452.

Song, W.T. and B.W. Schmeiser. 1995. Optimal mean-squared-error batch sizes. *Management Science* 41: 110-123.

Stahlbock, R. and S. Voß. 2008. Operations research at container terminals: a literature update. *OR Spectrum* 30(1), 1-52.

Steenken, D., S. Voß and R. Stahlbock. 2004. Container Terminal Operation and Operations Research - a Classification and Literature Review. *OR Spectrum* 26, 3-49.

Steiger, N.M. and J.R. Wilson. 2002. An improved batch means procedure for simulation output analysis. *Management Science* 48(12): 1569-1586.

Titterington, D.M. A.F.M. Smith and U.E. Makov. 1985. *Statistical Analysis of Finite Mixture Distributions*. Wiley, New York.

UNCTAD Transport Newsletter No. 24 (2004). UNCTAD/WEB/SDTE/TLB/2004/2. United Nations Conference on

Trade and Development (UNCTAD) Transport Section, Trade Logistics Branch, Geneva www.UNCTAD.org.

Vis, I.F.A. and R. De Koster. 2003. Transvesselment of Containers at a Container Terminal: an Overview. *European Journal of Operational Research* 147(1), 1-16.

Wilcox, R.R.. 1984. A table for Rinott's selection procedure. *Journal of Quality Technology* 16(2): 97-100.

Yücesan, E. and L.W. Schruben. 1992. Structural and behavioral equivalence of simulation models. *ACMTransactions on Modeling and Computer Simulation* 2(1), 82–103.

Yun, W.Y. and Y.S. Choi. 1999. A simulation model for container-terminal operation analysis using an object-oriented approach. *International Journal of Production Economics* 59, 221-230.

Zhang, C., Y. Wan, J. Liu and R.J. Linn. 2002. Dynamic crane deployment in container storage yards. *Transportation Research Part B* 36, 537-555.