

Generating Random Deviates for a Nonstationary Fixed-Planned Rate Process

MARCOS ANTONIO MASNIK FERREIRA

Programa de Pós-Graduação em Métodos Numéricos em Engenharia
Universidade Federal do Paraná
Centro Politécnico- Curitiba - Paraná
BRAZIL

CELSO CARNIERI

Programa de Pós-Graduação em Métodos Numéricos em Engenharia
Universidade Federal do Paraná
BRAZIL

RUI CARLOS BOTTER

Escola Politécnica
Universidade de São Paulo
BRAZIL

Abstract: - This paper presents the algorithms and techniques for generating random deviates for a nonstationary fixed-planned rate process (NSFPP), which we define as a stochastic process whose arrival rate by period is determined by a future demand. In the first part, the complete model is described and its objectives and variables are presented. After that, we show the algorithms and equations that implement the solution using Mixed-Integer Linear Programming (MILP) and goal programming. We show also how to get a random sequence with a fixed number of values bounded into an interval fitted to a specified probability distribution.

Key-Words: - Random Deviates, Input Data Analysis, Simulation, Modeling, Stochastic Process.

1 Introduction

Generating good random numbers is essential for a simulation project. A random number generator (GNA) aims to imitate or simulate the randomness required by the model. Typically, a GNA output is assumed to be a sequence of random numbers independent and identically distributed (IID) according to the uniform distribution $U \sim [0,1]$ [1]. These random numbers are then transformed into other probability distributions, such as the exponential, Poisson, normal or Gaussian, Weibull, etc, using basically the inverse transform, composition, convolution and acceptance-rejection methods [2]. These random sequences are used in simulation to model stationary process which means, in a wide sense, that the 1st and 2nd moments do not vary with respect to time. Nevertheless, these assumptions do not hold always in real systems. For example, assuming stationarity for the calls rate in a call center simulation model would lead to inaccurate results. Harrod and Kelton [3] review the risks of ignoring nonstationarity in processes and present three algorithms for the generation of nonstationary Poisson process (NSPP): direct, first walkup and second walkup algorithms. All these algorithms

assume Poisson arrivals (interevent time distribution is exponential) and the rate r_k for each k segment is a required input data. NSPP can be used to model natural phenomena like departures of commercial aircrafts, calls in a call center, telecommunication networks and Internet traffic. Most academic and commercial packages [4] have tools and functions that help the development of both stationary and NSPP models. For NSPP, the model builder should determine the r_k arrival rate from the data system. Nevertheless, this is not an easy task and it may even be impossible to determine it for some systems using the well-known input data analysis methods [5]. In a bulk cargo port system [6], for example, the truck and train arrival rate, on a daily basis, is determined not only by randomness but also by a future demand determined by the ships that will show up in order to load the goods. Therefore, the model builder can not set the arrival rate to any particular probability distribution. At first, it is necessary to estimate the arrival rate for each period and change it during the simulation. All of this is clumsy and a lot of time must be spent in order to deal with these matters inside the simulation model. There is a consensus among the simulation community that simulation modeling and analysis can be time consuming and

expensive [7]. Even considering hardware advances that permit rapid running, simulation packages should also improve their attributes by means of providing advanced tools and functions. As a consequence, simulation builders efforts could be concentrated in the model construction rather than in statistical matters. This paper presents a simulation model that implements the complexities and intricacies of the authors call a nonstationary fixed-planned rate process (NSFPP).

2 Fixed-Planned Rate Nonstationary Process

In a nonstationary fixed-planned rate process, the arrival rate by period is determined by a future demand in a specific destination. Consequently, the arrival rate should be calculated in such a manner that, at the end of a replication, the total generated amount of an entity for a demand period d and a class c must be in accord to a predefined demand $D_{d,c}$ ($d=[1...DP]$ and $c=[1...C]$). Basically, NSFPP differ from NSPP in three aspects: i) the arrival rate is not an input data but it must be calculated ii) the arrival distribution is not necessarily Poisson and iii) during a period, in spite of the interevent times being a random process, the exact amount determined by the rate must be generated. Furthermore, every destination has a buffer with a limited storage capacity (SC) and also a limited reception management capacity (RC) by period. Transportation means (TM) are used to transport the demand units to the destination in order to meet the demand $D_{d,c}$ and they range from 1 to ATM. Every TM has a specified capacity (CTM_{tm}) which means that each TM arrival brings into the simulation CTM_{tm} demand units. It is assumed that the transportation means capacity is the same for all classes. The programming planning for the purpose of meeting $D_{d,c}$ may be established WP periods before the starting of demand periods. The WP periods may be seen as the warm-up period in simulation and $[-WP, \dots, -1, 1, \dots, DP]$ is all the programming planning range.

2.1 Problem Variables Definition

Considering the NSFPP defined above, Table 1 summarizes the input data problem.

$D_{d,c}$	The demand, in units, that must be met for a period d and a class c .
SC	Every destination has a particular storage capacity (SC), in units, available for storing all at once the demand units for all classes.

RC	Reception management capacity in units by period.
SP	$D_{d,c}$ is required to be met, ideally, SP periods before the demand period d . Also known as 'safety periods'.
DP	The number of demand periods
WP	The number of periods before the start of the first demand period. Also known as Warm-up period.
LSMin	Minimum lot sizing in units generated by period.
LSMax	Maximum lot sizing in units generated by period.
C	Number of classes.
ATM	Number of transportation means.
CTM_{tm}	Transportation means capacity in units.
PTM_{c,tm}	Percentage of $D_{d,c}$ that must be brought into the system for the class c by the transportation mean tm .
Φ	The inverse of the cumulative distribution function for the process that is being modeled. If the process is Poisson, $\Phi = -\ln(u)$.

Table 1. NSFPP Variables

In order to solve the problem the following decision variables were used.

VT_{p,d,c}	The values, in units, by period $p=[-WP, \dots, -1, 1, \dots, DP]$ generated for a class c to meet the demand $D_{d,c}$.
VTTM_{p,d,c,tm}	The values, in transportation means units, by period p generated for a class c to meet the demand $D_{d,c}$. VTTM is the transportation mean arrival rate of the nonstationary process.

Table 2. NSFPP Decision Variables

At first, the $VT_{p,d,c}$ values are calculated taking into consideration i) the future demand $D_{d,c}$ which must be met ideally until the period $(d - SP - 1)$, ii) the reception management capacity and iii) the minimum and maximum lot sizing. After that, the $VT_{p,d,c}$ values are converted to transportation means units ($VTTM_{p,d,c,tm}$).

Considering the input data and decision variables introduced above, the proposed model has the following goals:

- 1) For every simulation replication,

$$D_{d,c} = \sum_{p=-WP}^{DP} VT_{p,d,c} ;$$

- 2) $\sum_{p=-WP}^{DP} VT_{p,d,c} \leq \sum_{p=-WP}^{DP} \sum_{tm=1}^{TM} VTTM_{p,d,c,tm} * CTM_{tm}$, in the sense that if we reduce VTTM in only one unit for the transportation mean with the smallest capacity (CTM_{tm}), this restriction would not hold anymore. As a consequence, 1 e 2 guarantee that the demand will be met using the smallest amount of transportation means.
- 3) $\frac{\sum_{p=-WP}^{DP} VTTM_{p,d,c,TMs} * CTM_{TMs}}{\sum_{p=-WP}^{DP} \sum_{tm=1}^{TM} VTTM_{p,d,c,tm} * CTM_{tm}} \cong PTM_{c,TMs}$ This condition guarantees that the amount generated using a specific transportation mean (TMs) would be, approximately, in accord to the percentage defined by PTM.
- 4) $LSMin \leq VT_{p,d,c} \leq LSMax$. The values generated by each period must be in the range [LSMin, LSMax].
- 5) $D_{d,c} = \sum_{p=-WP}^n VT_{p,d,c}$, $n = -WP \dots -1, 1 \dots DP$ e $n \leq (d - SP - 1)$. The demand must be met, ideally, until the period $d - SP - 1$.
- 6) $\sum_{d=1}^{DP} \sum_{c=1}^C VT_{p,d,c} \leq RC$, $p = [-WP, \dots, -1, 1, \dots, DP]$, the total amount generated by a destination must be, ideally, less or equal the reception management capacity by each period p .

2.2 Programming Planning Complexity

Considering that the horizon plan has n (WP + DP) periods and the demand is going to be met using i periods, then there are C_n^i possibilities for the

programming planning. Then, $\sum_{i=1}^n C_n^i$ is the total

number of possibilities for setting the programming planning for each demand. As we have DP demands,

$\left(\sum_{i=1}^n C_n^i\right)^{DP}$ is the total number of possibilities for the

programming planning for all demands. It is obvious that the number of possibilities grows exponentially as the demand periods increase.

Enforcing that the planning is made in consecutive periods could reduce significantly the number of possibilities. Considering that the planning for a specific demand starts at period 1, then the planning may be finished at period 1 to n . If the planning starts at period 2, the planning may be finished at period 2 to n , and so on. So, we have $n+(n-1)+(n-2)+\dots+1$

possibilities. The total number of possibilities is the sum of this arithmetic progression $\left(\frac{(1+n)n}{2}\right)$. As we

have the intention of planning DP demands, $\left(\frac{(1+n)n}{2}\right)^{DP}$ is the total number of possibilities. In

spite of the dimension reduction, it is evident that we are still facing a problem that grows exponentially as the number of demand periods increase.

Aiming at reducing more efficiently the number of possibilities, we impose that the period τ for the demand d , where τ is equal to $d - SP - 1$ ($d = 1 \dots DP$), makes part of the programming planning, which implies that $VT_{\tau,d,c} \neq 0$. So, the total number of possibilities is reduced to $(WP + DP)^{DP}$. The Fig 1 shows the relationship among WP, DP, d and τ variables. The τ period is the period immediately before the SP periods.

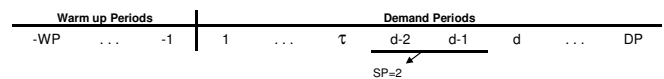


Fig 1 – Relationship among WP, DP, d and τ variables.

In order to give an idea of the reduction obtained with the procedures above, being WP=5, DP=20 and,

consequently, $n = 25$, then: $\left(\sum_{i=1}^n C_n^i\right)^{DP} =$

$$3,273E+150, \quad \left(\frac{(1+n)n}{2}\right)^{DP} = 1,728E+50 \quad e$$

$$(PA + DP)^{PD} = 9,095E+27.$$

3 Numerical Method for Realizing Nonstationary Fixed-Planned Rate Process

We present below the 'Main' algorithm that generates a vector called T whose contents are all the events for the nonstationary process needed by meeting all demands $D_{d,c}$.

```

Start Main
1  Get input data according to item 2.1
2  Set the programming planning, establishing the values
   VTp,d,c
3  Convert VTp,d,c to VTTMp,d,c,tm
4  For d = 1 to DP
5     For c = 1 to C
6         For tm = 1 to QTM
7             Call genEventsNSFPP(Td,c,tm, d, c, tm)
    
```

```

8      End for
9      End for
10     End for
11     Sort all the vectors  $T_{d,c,tm}$  by arrival time and merge
        them into a unique vector  $T$  whose entries are in the
        format: arrival time, tm, c, d.
12     Return  $T$ 
End
    
```

After getting all the required input data, the ‘Main’ algorithm calculates the $VT_{p,d,c}$ values (line 2) to meet the demands $D_{d,c}$ in accord to the objectives defined in item 2.1. In line 3, the values $VT_{p,d,c}$ (in demand units) are converted to $VTTM_{p,d,c,tm}$ (transportation means units). In line 7, the procedure *genEventsNSFPP* is called for every demand d , class c and transportation mean tm , for building the vector $T_{d,c,tm}$. Finally, all the vectors $T_{d,c,tm}$ are sorted and merged into only one vector T which is the algorithm’s goal.

3.1 Setting up the Programming Planning

Taking into consideration the input data and the decision variables presented, the programming planning problem is solved by the following mixed-integer linear programming (MILP).

$$\text{Min } Z = w1 * RME + w2 * WPE + w3 * SPA + w4 * DPA \quad (1)$$

-----Calculating RME-----

$$\text{ER}_p = \sum_{d=1}^{DP} \sum_{c=1}^C VT_{p,d,c}, \forall p = 1 \dots (WP + DP) \quad (2)$$

$$\text{ER}_p - RC - RMEP_p + RMEN_p = 0 \quad (3)$$

$$RME = \sum_{p=1}^{WP+DP} RMEP_p \quad (4)$$

-----Calculating WPE-----

-----Warm-Up Period Excess-----

$$\sum_{p=1}^{WP} \text{ER}_p - SC - WPE + WPB = 0 \quad (5)$$

-----Calculating SPA-----

-----Safety Periods Amount-----

$$SPI_{d,c} = \sum_{p=d+WP-SP}^{d+WP-1} VT_{p,d,c}, \forall d = 1 \dots DP \text{ e } c = 1 \dots C \quad (6)$$

$$SPA = \sum_{d=1}^{DP} \sum_{c=1}^C SPI_{d,c} \quad (7)$$

-----Calculating DPA-----

-----after Demand Period Amount-----

$$LDD_{d,c} = \sum_{p=d+WP}^{WP+DP} VT_{p,d,c}, \forall d = 1 \dots DP \text{ e } c = 1 \dots C \quad (8)$$

$$DPA = \sum_{d=1}^{DP} \sum_{c=1}^C LDD_{d,c} \quad (9)$$

Subject to:

$$\text{--- If } D_{d,c} = 0 \text{ then } DX_{d,c} = 0 \text{-----}$$

$$\text{--- If } D_{d,c} \neq 0 \text{ then } DX_{d,c} = 1 \text{-----}$$

$$D_{d,c} \geq DX_{d,c}, \forall d = 1 \dots DP \text{ e } \forall c = 1 \dots C \quad (10)$$

$$D_{d,c} \leq \text{BIG_M} * DX_{d,c}, \forall d = 1 \dots DP \text{ e } c = 1 \dots C \quad (11)$$

$$\text{--- If } VT_{p,d,c} = 0 \text{ then } X_{p,d,c} = 0 \text{-----}$$

$$\text{--- If } VT_{p,d,c} \neq 0 \text{ then } X_{p,d,c} = 1 \text{-----}$$

$$VT_{p,d,c} \geq X_{p,d,c}, \forall p = 1 \dots WP + DP, \quad (12)$$

$$VT_{p,d,c} \leq \text{BIG_M} * X_{p,d,c} \text{ d}=1 \dots DP \text{ e } c=1 \dots C \quad (13)$$

-----Restrictions for setting the planning contiguously before the safety period (SP).-----

$$X_{(d+WP-SP-1),d,c} = DX_{d,c}, \forall d = 1 \dots DP \text{ e } c = 1 \dots C \quad (14)$$

$$VT_{p,d,c} \geq VT_{p-1,d,c}, \forall d = 1 \dots DP, c = 1 \dots C \text{ e } p = 2 \dots d + WP - SP - 1, \quad (15)$$

$$VT_{p,d,c} \geq VT_{p+1,d,c}, \forall d = 1 \dots DP, c = 1 \dots C \text{ e } p = d + WP - SP \dots WP + DP - 1, \quad (16)$$

-----The total amount planned for a demand d must be equal to $D_{d,c}$.-----

$$D_{d,c} = \sum_{p=1}^{WP+DP} VT_{p,d,c}, \forall d = 1 \dots DP \text{ e } c = 1 \dots C \quad (17)$$

-----The total amount planned for a period must be in the range [LSMin, LSMax].-----

$$VT_{p,d,c} \geq \text{LSMin} * X_{p,d,c}, \forall p = 1 \dots WP + DP, \text{ d}=1 \dots PD \text{ e } c=1 \dots C \quad (18)$$

$$VT_{p,d,c} \leq \text{LSMax} * X_{p,d,c}, \forall p = 1 \dots WP + DP, \text{ d}=1 \dots DP \text{ e } \forall c = 1 \dots C \quad (19)$$

-----Binary Variables-----

$$X_{p,d,c} = \begin{cases} 0 & \text{if } VT_{p,d,c} = 0 \\ 1 & \text{if } VT_{p,d,c} \neq 0 \end{cases} \quad (20)$$

$$DX_{d,c} = \begin{cases} 0 & \text{if } D_{d,c} = 0 \\ 1 & \text{if } D_{d,c} \neq 0 \end{cases}$$

The MILP goal is to provide good alternatives for the programming planning instead of optimizing it, since simulation will be used for inspecting the alternatives’ adaptability. Depending on, essentially, the future demands $D_{d,c}$, the storage capacity (SC),

the safety periods (SP) and the reception management capacity (RC), it may not be possible to meet all the restrictions simultaneously. For this reason, these restrictions were relaxed by means of a weighted function [8]. The MILP mainly purpose is to get valid $VT_{p,d,c}$ values as a result of changing the weights w_1, w_2, w_3 and w_4 . Nevertheless, the $VT_{p,d,c}$ values are in demands units, so they need to be converted to $VTTM_{p,d,c,tm}$ – which defines the arrival rate for the nonstationary process.

3.2 Converting $VT_{p,d,c}$ to $VTTM_{p,d,c,tm}$

The following algorithm shows how to convert $VT_{p,d,c}$ to $VTTM_{p,d,c,tm}$. At first hand, the total number of transportation means ($valTM_{tm}$), needed for bringing into the system the amounts specified by $D_{d,c}$, must be calculated. This is the number per transportation mean, in accord to $PTM_{c,tm}$, that must be generated in order to meet the $D_{d,c}$. Assuming $PTM_{c,tm}$ is sorted by capacity descending, the algorithm assures that the values of $valTM_{tm}$ will be the minimum amount needed to meet the demands $D_{d,c}$.

```

Start
1 If  $D_{d,c} \neq 0$  then
2   totalGen = 0
3   For  $tm = 1$  to QTM
4     If  $tm < QTM$  then
5        $valTM_{tm} = round\left(\frac{D_{d,c} * PTM_{c,tm}}{CTM_{tm}}\right)$ 
6       totalGen = totalGen + ( $valTM_{tm} *$ )
7     Else
8        $valTM_{tm} = \left(\frac{D_{d,c} - totalGen}{CTM_{tm}}\right)$ 
9        $valTM_{tm} = round(valTM_{tm} + 0.5)$ 
10    End if
11  End for
12  Convert  $VT_{p,d,c}$  to  $VTTM_{p,d,c,tm}$  in accord to  $valTM_{tm}$ 
13End if
End
    
```

Following the determination of $valTM_{tm}$, the next step (line 12) is to determine in which way these values will be distributed among the planning periods defined in $VT_{p,d,c}$ matching, as much as possible, the values $VT_{p,d,c}$ by period. Let PPA be the amount of planning periods programmed ($VT_{i,d,c} \neq 0, \forall i=1 \dots PPA$), where 'i=1' means the first period where $VT_{i,d,c} \neq 0$. The following integer linear programming problem (ILPP) makes the conversion from $VT_{p,d,c}$ to $VTTM_{p,d,c,tm}$.

$$\text{Min } Z = \sum_{i=1}^{PPA} XP_i \tag{21}$$

Subject to:

$$\left(\sum_{i=1}^{PPA} (VTTM_{i,d,c,tm} * CTM_{tm}) \right) + XN_i - XP_i = VT_{i,d}. \tag{22}$$

$$\sum_{i=1}^{PPA} VTTM_{i,d,c,tm} = valTM_{tm} \tag{23}$$

-----Binary Variable-----

$$VTTM_{i,d,c,tm}, \forall i = 1 \dots PPA, d = 1 \dots DP, \\ c = 1 \dots C \text{ e } tm = 1 \dots QTM$$

The $VT_{p,d,c}/VTTM_{p,d,c,tm}$ conversion is also done using goal programming. The ILPP objective is to i) assure that the sum of all transport means distributed among the periods (1...PPA) be equal to $valTM_{tm}$ (23) and, simultaneously, ii) minimize the difference, in excess,

$$\text{between } VT_{p,d,c} \text{ and } \left(\sum_{i=1}^{PPA} (VTTM_{i,d,c,tm} * CTM_{tm}) \right).$$

3.3 Generating Random Deviates According to $VT_{p,d,c,tm}$

What has been done up to this point was the computation of $VTTM_{p,d,c,tm}$, which is the arrival rate for the nonstationary process. Therefore, the events generation, according to this rate, is the next step (line 7 of the 'Main' algorithm). The following algorithm was developed based largely on the algorithms presented in [3]. In spite of the similarities, two distinct aspects must be pointed out: i) $VTTM_{p,d,c,tm}$ is a fixed-planned rate (the exact number of events must be generated) and ii) the nonstationary process is not necessarily Poisson. At first hand, in line 1, the vector $T_{d,c,tm}$ is dimensioned to $\sum_{p=1}^P VTTM_{p,d,c,tm}$ (where $P = WP + DP$), since it has a fixed size. All the events are generated from line 7 to 14. In line 10, it was used a function Φ that is the inverse of the cumulative distribution function (CDF), with arrival rate equal to 1, of the process that is going to be considered (e.g. $\Phi = -\ln(u)$ if the process is Poisson). Taking into account that the events generation is governed by a stationary stochastic process (Φ function), it may be possible to go unwantedly beyond the limits of the segment B_k . In this case, from line 15 to 23, the 'n' generated events have their interarrival time compressed, nevertheless keeping their proportionally, in order to be fitted inside the interval $[B_{k-1}, B_k]$.

```

Start genEventsNSFPP( $T_{d,c,tm}$ , d, c, tm)
1 Dimension the vector  $T_{d,c,tm}$  to  $\sum_{p=1}^P VTTM_{p,d,c,tm}$ 
2 n = 0
3 For k = 1 to P
4   If  $VTTM_{k,d,c,tm} \neq 0$  then
5     tnPrevious =  $B_k - 1$ 
6     iP = n + 1
7     For i = 1 to  $VTTM_{k,d,c,tm}$ 
8       n = n + 1
9       Generate a random number  $u \sim U[0,1]$ 
10      A =  $\Phi(u)$ 
11      W =  $A / VTTM_{k,d,c,tm} * (B_k - B_{k-1})$ 
12       $T_{d,c,tm,n} = tnPrevious + W$ 
13      tnPrevious =  $T_{d,c,tm,n}$ 
14    End for
15    If  $T_{d,c,tm,n} > B_k$  then
16      ajustPrevious =  $B_{k-1}$ 
17      notAjustPrevious =  $B_{k-1}$ 
18      For i = iP to n
19        temp = ajustPrevious +
                 $\frac{(T_{d,c,tm,i} - notAjustPrevious)}{T_{d,c,tm,n} - B_{k-1}} * (B_k - B_{k-1})$ 
20        notAjustPrevious =  $T_{d,c,tm,i}$ 
21         $T_{d,c,tm,i} = temp$ 
22        ajustPrevious =  $T_{d,c,tm,i}$ 
23      End for
24    End if
25    If n =  $C_{d,c,tm,P}$  then
26      Exit for
27    End if
28  End for
29 Return  $T_{d,c,tm}$ 
End

```

4 Conclusion.

This paper provided a detailed description of how to generated events for a nonstationary fixed-planned rate process. The proposed algorithms can be implemented before the beginning of a simulation or before the beginning of a replication. In spite of the option chosen, changing the weighted vector (W_i) may be used as a strategy for modifying the planning arrival rate ($VTTM_{p,d,c,tm}$). If the algorithms implementation is put in the beginning of the simulation, the model builder should consider, for each replication, execute again the 'genEventsNSFPP' algorithm in order to guarantee variability in the interarrival time sequence. The idea, in the whole simulation, is to analyze the effects of a specific programming planning during all replications. In addition, the authors believe that the generalization of the Φ function may be a key factor for extending

the modeling usefulness over a broad range of system applications. The Φ function allows the model simulation builder to specify the probability distribution that best fits the arrival rate process. In addition, for those distributions whose CDF does not have an inverse form, any other methods, like acceptance-rejection, could be used.

References:

- [1] Pierre L'Ecuyer, Good Parameters and Implementations for Combined Multiple Recursive Random Number Generators. Operations Research, Vol. 47, N° 1, Jan-Feb, 1999.
- [2] R. C. H. Cheng, "Random Variate Generation", Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice, J. Banks, New York, John Wiley, 1998.
- [3] S. Harrod, W. D. Kelton, Numerical Methods for Realizing Nonstationary Poisson Process with Piecewise - Constant Instantaneous - Rate Functions, Simulation: Transactions of The Society for Modeling and Simulation International, 2006.
- [4] W. D. Kelton, R. P. Sadowski, D. A. Sadowski. Simulation with Arena. Second Edition, McGraw-Hill, Boston, 2001
- [5] Stephen Vicent, "Input Data Analysis", Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice, J. Banks, New York, John Wiley, 1998.
- [6] M. A. Masnik Ferreira, R. Mendes Júnior, C. Carnieri, Dock Systems Performance Analysis Trough Mathematics And Statistics Simulation, Revista Produção On-line, Florianópolis, Vol. 7, N° 3, Nov, 2007.
- [7] Jerry Banks, "Principles of Simulation", Handbook of Simulation: Principles, Methodology, Advances, Applications and Practice, J. Banks, New York, John Wiley, 1998
- [8] S. M. Lee, Goal programming for decision analysis, Auerback, Philadelphia, 1972