A Method to Generate Random Deviates for a Fixed-Planned Rate Nonstationary Renewal Process

Marcos A. Masnik Ferreira, Celso Carnieri, Rui C. Botter

Abstract**—**The algorithms and techniques necessary to generate random deviates for a fixed-planned rate nonstationary renewal process (FPRNSRP) are presented in this paper. The authors define this process as a stochastic renewal process (RP) whose arrival rate by period is determined by a future demand. At first, the complete model is described and its objectives and variables are presented. After that, the algorithms and equations, that implement the solution using Mixed-Integer Linear Programming (MILP) and goal programming, are presented. It is also presented an algorithm to get a random sequence with a fixed number of values bounded into an interval fitted to a specified probability distribution.

Keywords**—**Input Data Analysis, Random Deviates, Simulation, Modeling, Stochastic Process.

I. 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~[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 always hold 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 allow 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 what the authors call a fixed-planned rate nonstationary renewal process (FPRNSRP).

II. Fixed-Planned Rate Nonstationary RENEWAL Process

Fig 1 shows the relationship among all the FPRNSRP

Manuscript received December 10, 2008.

Marcos A. Masnik Ferreira works at Central Bank of Brazil. In addition, he is a Ph.D student at the Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE) of Federal University of Paraná (UFPR) (e-mail: masnik@gmail.com). The views expressed in this work are those of the authors and do not necessarily reflect those of Central Bank of Brazil or its members

Celso Carnieri is Professor at Federal University of Paraná. (e-mail: carnieri@gmail.com).

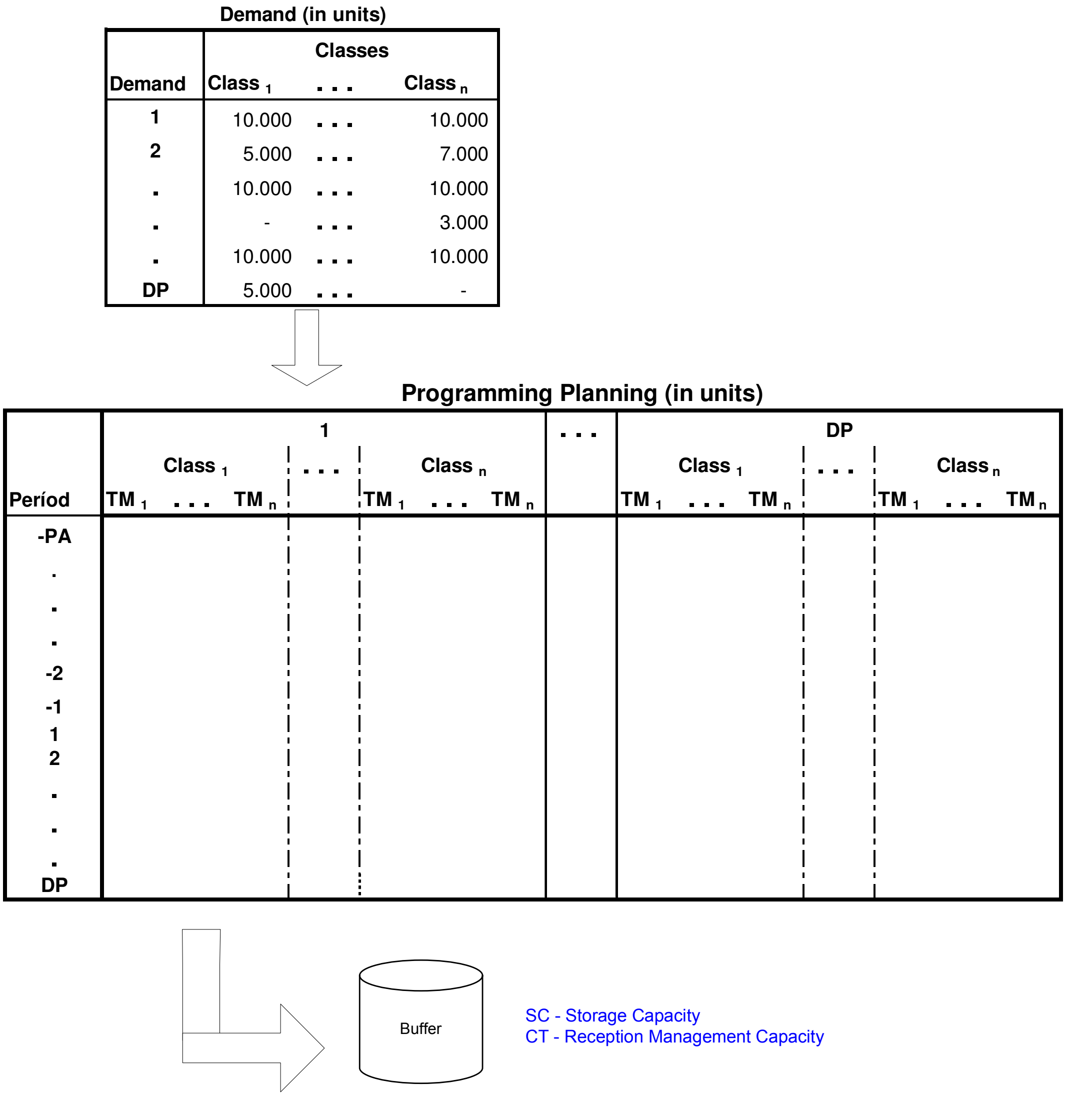
Rui Carlos Botter is Professor Polytechnic School of the University of São Paulo (e-mail: rcbotter@usp.br).

components. In this type of model, the arrival entities are generated to a specific destination. Besides, in FPRNSRP model, 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 with a predefined demand $D_{d,c}$ (d=[1…DP] and c=[1…C]).

Basically, FPRNSRP 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*,…,-1,1,…*DP*] is all the programming planning range.



Demand (in units)

	Classes		
Demand	Class $_1$	. . .	Class $_n$
1	10.000	. . .	10.000
2	5.000	. . .	7.000
.	10.000	. . .	10.000
.	-	. . .	3.000
.	10.000	. . .	10.000
DP	5.000	. . .	-

Fig 1-Basic components of a FPRNSRP simulation model

A. *Problem Variables Definition*

Considering the FPRNSRP defined above, Table I summarizes the input data problem.

TABLE I: FPRNSRP VARIABLES

$D_{d,c}$	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.
$\boldsymbol{CTM_{tm}}$	Transportation means capacity in units.
$\boldsymbol{PTM_{c,tm}}$	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, Φ=-ln(u).

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

TABLE II: FPRNSRP DECISION VARIABLES

$\boldsymbol{VT_{p,d,c}}$	The values, in units, by period *p*=[-*WP*,…,-1,1,…*DP*] generated for a class *c* to meet the demand $D_{d,c}$.
$\boldsymbol{VTTM_{p,d,c,tm}}$:	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.

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* by 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 (TM) would be, approximately, in accord with 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*...-1,1...*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*,…,-1,1,…*DP*],

the total amount generated by a destination must be, ideally, less or equal the reception management capacity by each period *p*.

Based on the input data and objectives specified above, a vector T_i, whose entries represent the events for the FPRNSRP process, must be generated.

B. Programming Planning Complexity

Taking into account 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)+...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…DP), makes part of the programming planning, which implies $VT_{\tau,d,c} \neq 0$. So, the total number of possibilities is reduced to $(WP + DP)^{DP}$. Fig 2 shows the relationship among WP, DP, d and τ variables. The τ period is the period immediately before the SP periods.

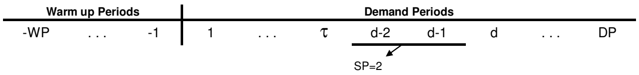


Fig 2– Relationship among WP, DP, d and τ variables.

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, $\left(\frac{(1+n)n}{2}\right)^{DP}$ = 1,728E+50 e $(PA+DP)^{PD}$ = 9,095E+27.

III. Numerical Method for Realizing A Fixed-Planned Rate Nonstationary Renewal Process

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

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 with the objectives defined in item II.A. 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 *genEvents*FPRNSRP 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.

```
Start Main
1  Get input data according to item 2.1
2  Set the programming planning, establishing the values $VT_{p,d,c}$
3  Convert $VT_{p,d,c}$ to $VTTM_{p,d,c,tm}$
4  For d = 1 to DP
5    For c = 1 to C
6      For tm = 1 to QTM
7        Call genEventsFPRNSRP ($T_{d,c,tm}$, 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
```

A. 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):

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

----------------------Calculating RME---------------------
-----------Reception Management Excess---------------

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

$$ERp \text{ - } RC \text{ - } RMEP_p + RMEN_p = 0 \text{ , } \forall\ p = 1...(\ WP + DP) \quad (3)$$

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

----------------------Calculating WPE---------------------
------------------Warm-Up Period Excess---------------

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

----------------------Calculating SPA----------------------
------------------Safety Periods Amount-----------------

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

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

----------------------Calculating DPA---------------------
------------------after Demand Period Amount---------

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

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

Subject to:

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

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

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

$$D_{d,c} \leq\ BIG\_M * DX_{d,c} \text{ ,} \forall\ d = 1...DP \text{ and } c = 1...C \quad (11)$$

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

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

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

$$VT_{p,d,c} \leq BIG\_M * X_{p,d,c} \text{ ,} \forall\ p=1...WP+DP,\ d=1...DP \text{ and } c=1...C(13)$$

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

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

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

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

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

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

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

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

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

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

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

$$DX_{d,c} = \begin{Bmatrix} 0 & if & D_{d,c} = 0 \\ 1 & if & D_{d,c} \neq 0 \end{Bmatrix} \quad (21)$$

Fig 3 presents a numerical example to demonstrate the relationship among the proposed model variables. For this example, we have a demand period equal to 5 and a warm-up period of 14. That means there are still 14 periods until the start of the demand periods. In addition, it was established 2 periods as the safety period. Then, it is desirable that all the units must already been generated preferably 2 periods before the start of demand periods.

Fig 3 also shows the *VT* values that will be generated by period in order to achieve the needed demand. According to this programming planning, the *SPA* and *DPA* values are equal to zero, since it was not necessary to use neither the safety nor the demand periods. The total amount generated during the warm-up period ($-14 \leq p \leq -1$) is equal to 106.000. If we consider that the value of the *SC* variable provided is equal to 96.000, then *WPE* would be equal to 10.000. In this case, there was an excess of 10.000 units inside the warm-up period. For the great number of periods considered, the *VT* summation, for every demand, attains at maximum 8.216 units. As a consequence, if we consider *RC* equal to 9.000, *RME* would be equal to zero. Finally, it should be pointed out that the programming planning achieves the demand for every period.

	Period	Demand	Amount by Period	VT				
				1	2	3	4	5
W P	-14		5.347	2500	705	2143	0	0
	-13		5.347	2500	705	2143	0	0
	-12		8.147	2500	705	2143	0	2800
	-11		8.147	2500	705	2143	0	2800
	-10		8.216	2500	773	2143	0	2800
	-9		8.216	2500	773	2143	0	2800
	-8		8.216	2500	773	2143	0	2800
	-7		8.216	2500	773	2143	0	2800
	-6		8.216	2500	773	2143	0	2800
	-5		8.216	2500	773	2143	0	2800
	-4		8.216	2500	773	2143	0	2800
	-3		8.216	2500	773	2143	0	2800
	-2		8.143	0	3000	2143	0	3000
	-1		5.143	0	0	2143	0	3000
D P	1	30.000	0	0	0	SPA	0	3000
	2	12.000	0	0	0		0	3000
	3	30.000	0	0			0	0
	4	-	0	0	DPA		0	0
	5	40.000	0	0	0		0	0

Fig 3– Relationship among *WP*, *DP*, *VT, SPA* and *DPA* variables.

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 *w1*, *w2*, *w3* and *w4*. 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.

In order to calculate the reception management excess, by period (*RME*), (2) sets to ER_p the amount of units generated for every demand and class. Equation (3) sets to $RMEP_p$ the excess in units that must have taken place compared to the Resource Capacity (*RC*) and, in this case, sets zero to $RMEN_p$. On the contrary, (2) sets zero to $RMEP_p$ and the difference between *RC* and ER_p. Equation (4) sets to *RME* the total amount of excess that must have happened in all periods (warm-up and demand).

Supposing that the programming planning has exceeded the storage capacity (*SC*), during the warm-up, (5) sets this excess to *WPE* and zero to *WPB*. On the contrary, (5) sets zero to *WPE* and the difference between ER_p and *SC* to *WPB*.

The *SPA* evaluation is done by (6) and (7). Equation (6) sets to $SPI_{d,c}$ the summation of *VT*, during the safety period, for every demand and class. Equation (7) sets to *SPA* the summation of every *SPI* variables.

Equations (8) and (9) calculate the amount of units generated after the starting of the demand period. At first, (8) sets to $LDD_{d,c}$ the summation of *VT*, after the starting of the demand period, for every demand and class. Equation (9) sets to *DPA* the summation of every *LDD* variables.

Both (10) and (11) make the $DX_{d,c}$ variables assume the values 0 and 1 if the corresponding $D_{d,c}$ variables are equal to or different from zero. Equations (12) and (13) make the $X_{p,d,c}$ variables assume the values 0 and 1 if the corresponding $VT_{pd,c}$ variables are equal to or different from zero.

Equation (14) obligates the period ($d + WP - SP - 1$), immediately before the starting of the safety period, to be part of the programming planning. If the variable $DX_{d,c}$ is different from zero, $X_{d+WP-SP-1}$ will be too. Equations (15) and (16) force the programming planning to be done in a sequential manner i.e., the programming planning will be done contiguously.

Equation (17) implies that the amount generated by the programming planning must meet the $D_{d,c}$ values. Equations (18) and (19) define the limits by period that must be considered by the programming planning. Equations (20) and (21) define the $X_{p,d,c}$ and $DX_{d,c}$ variables as binary.

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

The $VT_{p,d,c}$ values, calculated by the MILP described above, are given in demand units. Nevertheless, according to the proposed simulation model, the arrival entities are transportation means, which have a specified transport capacity defined by CMT_{mt}. For this reason, it is necessary to convert the $VT_{p,d,c}$ values to $VTTM_{p,d,c,tm}$ units.

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              $valTM_{tm} = \left(\frac{D_{d,c}\ \text{-}\ totalGen}{CTM_{tm}}\right)$
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…*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 $VTTM_{p,d,c,tm}$.

$$\textbf{Min}\ \ Z = \sum_{i=1}^{PPA} XP_i \tag{22}$$

Subject to:

$$\left(\sum_{i=1}^{PPA}\left(VTTM_{i,d,c,tm} * CTM_{tm}\right)\right) + XN_i - XP_i = VT_{i,d,c} \tag{23}$$

$$\sum_{i=1}^{PPA} VTTM_{i,d,c,tm} = valTM_{tm} \tag{24}$$

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

$$VTMT_{i,d,c,tm},\ \forall\ i = 1....PPA,\ d = 1....DP,\ c = 1....C \text{ and } tm = 1....QTM \qquad (25)$$

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 *valTMtm* in (24) and, simultaneously, ii) minimize the difference, in excess, between $VT_{p,d,c}$ and $\sum_{i=1}^{PPA} VTTM_{i,d,c,tm} * CTM_{tm}$.

Taking into consideration the following input entries, Table III summarizes the numerical algorithm results for the $VT_{p,d,c}/VTTM_{p,d,c,tm}$ conversion.

- $valTM_{tm}$={249,65}
- CTM_{tm}={36,324}
- PTM_{tm}={0.3, 0.7}
- $VT_{i,d,c}$={3000, 3000, 3000, 3000, 3000, 3000, 3000, 3000, 3000, 3000}

For this programming planning, there are 2 transportation means with capacities equal to 36 and 324 units, 10 consecutive periods planning with 3000 demand units each. The planning must be arranged by 249 and 65 transportation means units tm=1 and tm=2, respectively.

The results show up that the 3000 demand units by period were respected, as much as possible, since the transportation means capacities are not multiples of the demands. In addition, the total amount planned (30024) is the best solution that meets the demand. If only one unit of the transportation mean with the lower capacity (36) is removed from the programming planning, the total demand would not be met anymore. Finally, the percentage of each transportation mean, defined by PTM_{tm}, were respected as much as possible, since $8964/30024 \cong 30\%$ and $21060/30024 \cong 70\%$.

TABLE III: ALGORITHM RESULTS FOR THE $VT_{P,D,C}/VTTM_{P,D,C,TM}$ CONVERSION

Total by period					
	TMs units		Demand units		
Period	tm=1	tm=2	tm=1	tm=2	Total
1	65	2	2340	648	2988
2	3	9	108	2916	3024
3	3	9	108	2916	3024
4	2	9	72	2916	2988
5	2	9	72	2916	2988
6	2	9	72	2916	2988
7	3	9	108	2916	3024
8	2	9	72	2916	2988
9	83	0	2988	0	2988
10	84	0	3024	0	3024
Total	249	65	8964	21060	30024

C. 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 an 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 genEventsFPRNSRP ($T_{d,c,tm}$, 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
```

```
            Generate a random number u~U[0,1]
                $A = \Phi(u)$
                $W = A / VTTM_{k,d,c,tm} * (B_k - B_{k-1})$
                $T_{d,c,tm,n}$ = tnPrevious + W
                tnPrevious = $T_{d,c,tm,n}$
            End for
            If $T_{d,c,tm,n} > B_k$ then
                ajustPrevious = $B_{k-1}$
                notAjustPrevious = $B_{k-1}$
                For i = iP to n
                    $temp = ajust\Pr evious + \frac{(T_{d,c,tm,i} - notAjustPrevious)}{T_{d,c,tm,n} - B_{k-1}} * (B_k - B_{k-1})$
                    notAjustPrevious = $T_{d,c,tm,i}$
                    $T_{d,c,tm,i}$ = temp
                    ajustPrevious = $T_{d,c,tm,i}$
                End for
            End if
            If n = $C_{d,c,tm,P}$ then
                Exit for
            End if
        End if
    End for
    Return $T_{d,c,tm}$
End
```

D. Algorithm Results

The following input data were used to show the algorithm results:

- *SC = 96000*
- *RC = 13000*
- *SP = 2*
- *DP= 30*
- *WP = 14*
- *LSMin = 300*
- *LSMax = 3000*
- *C = 3*
- *ATM = 2*
- $CMT_{tm} = \{36, 324\}$
- $PMT_{c,tm} = \begin{bmatrix} 0.3 & 0.7 \\ 0.5 & 0.5 \\ 0.7 & 0.3 \end{bmatrix}$
- $\Phi = -ln(u)$
- $D_{d,c} =$

	Classes		
	1	2	3
DP	**Class1**	**Class2**	**Class3**
1	10.000	10.000	10.000
2	5.000	-	7.000
3	10.000	10.000	10.000
4	-	-	-
5	10.000	20.000	10.000
6	-	-	-
7	5.000	10.000	20.000
8	-	-	-
9	2.000	5.000	3.000
10	10.000	5.000	25.000
11	10.000	10.000	10.000
12	-	-	-
13	2.000	2.000	6.000
14	-	-	-
15	12.000	8.000	20.000
16	-	-	-
17	-	-	-
18	20.000	-	-
19	12.000	1.000	3.000
20	-	-	-
21	-	-	-
22	-	40.000	-
23	-	-	-
24	20.000	-	20.000
25	30.000	-	-
26	-	-	-
27	-	-	-
28	5.000	-	-
29	-	-	-
30	-	-	-
Total	163.000	121.000	144.000

As it has been already explained, the proposed method generates a vector T_i whose entries have the following format: *arrival time, tm, c, d.* Table IV summarizes the amount of generated events by the proposed method, by each class, transportation mean and demand period.

TABLE IV: AMOUNT OF GENERATED EVENTS BY THE PROPOSED METHOD

DP	Class 1				
	# (Units)		# Demand Units		Total (Demand Units)
	tm=1	tm=2	tm=1	tm=2	
1	80	22	2880	7128	10008
2	40	11	1440	3564	5004
3	80	22	2880	7128	10008
5	80	22	2880	7128	10008
7	40	11	1440	3564	5004
9	20	4	720	1296	2016
10	80	22	2880	7128	10008
11	80	22	2880	7128	10008
13	20	4	720	1296	2016
15	100	26	3600	8424	12024
18	169	43	6084	13932	20016
22			0	0	0
24	169	43	6084	13932	20016
19	100	26	3600	8424	12024
25	249	65	8964	21060	30024
28	40	11	1440	3564	5004
Total	1347	354	48492	114696	163188

DP	Class 2				
	# (Units)		# Demand Units		Total (Demand Units)
	tm=1	tm=2	tm=1	tm=2	
1	143	15	5148	4860	10008
2			0	0	0
3	143	15	5148	4860	10008
5	277	31	9972	10044	20016
7	143	15	5148	4860	10008
9	67	8	2412	2592	5004
10	67	8	2412	2592	5004
11	143	15	5148	4860	10008
13	29	3	1044	972	2016
15	115	12	4140	3888	8028
18			0	0	0
22	554	62	19944	20088	40032
24			0	0	0
19	10	2	360	648	1008
25			0	0	0
28			0	0	0
Total	1691	186	60876	60264	121140

DP	Class 3				
	# (Units)		# Demand Units		Total (Demand Units)
	tm=1	tm=2	tm=1	tm=2	
1	197	9	7092	2916	10008
2	141	6	5076	1944	7020
3	197	9	7092	2916	10008
5	197	9	7092	2916	10008
7	385	19	13860	6156	20016
9	57	3	2052	972	3024
10	488	23	17568	7452	25020
11	197	9	7092	2916	10008
13	113	6	4068	1944	6012
15	385	19	13860	6156	20016
18			0	0	0
22			0	0	0
24	385	19	13860	6156	20016
19	57	3	2052	972	3024
25			0	0	0
28			0	0	0
Total	2799	134	100764	43416	144180

The amount of generated events of each class is not exact, since the demand units and the transportation means capacities are not multiples. However, the values got by the method are as close as to the demand units.

It should be regarded that the percentages $PMT_{c,tm}$ are respected as much as possible to the amount of units generated. In addition, $\sum_{p=-WP}^{DP}\sum_{tm=1}^{TM} VTTM_{p,d,c,tm} * CTM_{tm}$ is equal to 428.508 units – very close to $\sum_{d=1}^{DP}\sum_{c=1}^{C} D_{d,c}$ = 428.000 (demanded units).

IV. Conclusion

This paper provided a detailed description of how to generate 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. Regardless 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 made at the beginning of the simulation, the model builder should consider, for each replication, to execute again the '*genEvents*FPRNSRP' 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.

The authors also believe that the method proposed could de useful for developing simulation models in the area of production control systems, as in [9]. It remains to be investigated the method's usefulness to simulation modeling in other areas like computers networks [10], [11].

References

[1] Pierre L'Ecuyer, Good Parameters and Implementations for Combined Multiple Recursive Random Number Generators. Operations Research, Vol. 47, Nº 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.

[9] K. Onan, B. Sennarouglu, A Comparative Study of Prodction Control Systems through Simulation , Proceedings of the 12th WSEAS Int. Conf. on APPLIED MATHEMATICS, December, 2007.

[10] P.M.Papazoglou, D.A.Karras, R.C.Papademetriou. Improved Simulation System Performance for Wireless Communications using Efficient Multi-Threading Architectures, Proceedings of the 11th WSEAS International Conference on COMMUNICATIONS, July, 2007

[11] M. Porjazoski, B. Popovski. Simulation Analysis of Packet Scheduling Algorithm for WWW and Video Streaming Service in UMTS Downlink FDD Mode, Wseas Transactions on Communications, Issue 3, Vol 2, July, 2003.