USING GENETIC ALGORITHMS WITH SIMULATION IN VALUE BASED ROUTING IN CONTACT CENTERS

Ehsan Mohammady Ardehaly, Dr. Shahriar Mohammadi
Ehsan@rasatech.com

ABSTRACT

Customer contact centers, usually handle several types of customer service requests (calls). Customer service representatives (Agents) have different skills to handle different requests. Contact centers usually use skill-based routing (SBR) as routing algorithm, to assign calls to appropriate agent. Enterprise contact centers use value-based routing (VBR) as a routing algorithm, to maximize expected value accrued from having the agent handle the call. There are several VBR algorithms, they usually use stochastic models for arrivals of call types and their service-time distributions, or use linear programming algorithms. In this paper, ***genetic algorithm*** is introduced with computer simulation with the aim of finding best expected value. The main advantage of this algorithm is easy implementation, and does not need to resolve complex linear programming. Finally, with linear programming efficiently of the algorithm will be evaluated with several samples.

KEYWORDS

Contact Center, Value Based Routing, Genetic Algorithm, Computer Simulation, Linear Programming.

1 INTRODUCTION

Today’s business environment is extremely competitive and firms everywhere are actively seeking to grow their customer base. As a result, one way to create best customer service is “contact center”. Contact center is at the very heart of the customer experience. Contact center causes higher level of customer services as well as maximum skills and availability.

Many services - from emergency to retail - are largely teleservices, in that the people who provide the service and the people who receive the service, herein called customers, are remote from each other, at least when the service is initiated. With a teleservice, the delivery of service is provided or enabled by a customer contact center; e.g.[1].

Mehrotra (1997) defines call centers as “Any group whose principal business is talking on the telephone to customers or prospects” [2]. Call centers typically handle more than one type of call, with each distinct call type referred to as a “queue”. Through Automatic Call Distribution (“ACD”) and Computer Telephony Interaction (“CTI”) devices, inbound calls can be routed to agents, groups, and/or locations, with advancements in these routing technologies supporting more and more sophisticated logic over time. Individual agents can be skilled to handle one type of call, several types of calls, or all types of calls, with different priorities and preferences specified in the routing logic. Alternative media such as email, fax, web pages and instant messaging can be supported in contact centers. Contact centers usually handle several different kinds of service requests, herein referred to as calls. For example, telephone callers may speak different languages or may call about different promotions.

This paper aims to go beyond these traditional congestion measures and zero-one skill requirements to focus on the expected value accrued from having the agent handle the call; i.e., and propose going beyond traditional SBR to obtain value-based routing (VBR). Expected value might represent expected revenue or the likelihood of first-call resolution. With modern CRM systems, such information is actually available. Value might also reflect satisfying agent call-handling preferences [3].

In this paper, the performance target that introduced in Sisselman, Whitt [3] will used to find (near) best expected value with genetic algorithms with random population. In addition, direct VBR is included in population. After running genetic algorithm in simulated contact center in last period time (for example every 1 hour or 30 minutes), best preference matrix will be found. This matrix is expected to have good efficiency in next period time.

The algorithm is validated, because can increase total value against direct VBR, in addition using linear programming in some samples, evaluate that the performance is near best possible performance.

2 VALUE BASED ROUTING

2.1 A performance target

Sisselman, Whitt defined performance target formula [3]; assume that there are m call types and n agents. Let assume that a value $v_{i,j}$ (a real number) has been assigned for agent i handling a type-j call for each i and j for which an assignment is allowed. It still wants relatively few calls to be abandoned and most calls to be answered promptly. Thus, let ascribe a cost (negative value) $c_{L,j}$ to each type-j call that is lost due to customer abandonment and a cost $c_{D,j}$ to each served type-j call that has to wait more than y_j seconds.

Now a specific total-value function can defined. To do so, let specify a time interval over which performance is to be judged, e.g., a typical half hour. Let $N_{i,j}$ be the number of type-j calls answered by agent i within y_j seconds during the specified time period. Let L_j be the number of type-j calls lost because of customer abandonment within the designated time period, and let D_j be the number of answered type-j calls delayed beyond y_j seconds within the designated time period. Clearly, the quantities $N_{i,j}$, L_j and D_j should be regarded as random variables, which depend on the unknown pattern of arrivals and service times as well as the routing policy used.

For each routing algorithm, let the total value gained from the routing algorithm under consideration being equal to the expected total value, i.e. [3].

$$V = \sum_{i=1}^{n}\sum_{j=1}^{m}(E[N_{i,j}].v_{i,j}) - \sum_{j=1}^{m}([E[L_j].c_{L,j}] + [E[D_j].c_{D,j}])$$

The overall goal is to maximize the total value expressed in formula.

2.2 A Priority-Based Algorithm and Direct VBR

Priority matrix framework could use based on n × m priority matrix. The matrix element $P_{i,j}$ gives the priority level for agent i to handle call type j. In addition, the Ward Whitt's routing algorithm is states [3]:

- **When a new type-j call arrives**, route (assign) the call to the available agent i with the highest positive priority level $P_{i,j}$.
- **When agent i becomes free**, after completing a call, look for waiting calls to assign to agent i. Let the candidate calls to assign to agent i be from the front of the m call-type queues

Now, the goal is to find the best priority matrix in every period that creates the best-expected total value. It can be obtained direct VBR algorithm by simply identifying the required priorities with the given values, i.e., by letting $P_{i,j} = v_{i,j}$ for all i and j, where $v_{i,j}$ are the given values. In next sections shown that direct VBR usually does not cause maximum expected total value.

2.3 Simulating Contact Center

Since contact centers are complicated, especially with skill-based routing, it is natural to rely on simulation to analyze the resulting stochastic models. And, indeed, simulation has often been applied to analyze call-center models. A simulation tool called the Call Processing Simulator (CAPS) was created and extensively applied at AT&T [4].

Simulation is usually applied offline to make system studies. However, with the steady increase of computer power, it is becoming possible to perform simulations dynamically in real time to predict and control congestion. To evaluate the performance of real-time simulation, a system simulation could be performing accompanied by additional replications of transient simulations performed through simulation time in the main run [1].

The biggest challenge of call center simulation modeling is the definition and organization of model inputs. As reflected in figure 1 the basic building blocks of a call center simulation model are the calls, the agents, and the time period during which the call center is open. In turn, the basic routing logic connects the way that the calls interact with the people during that time period [2].

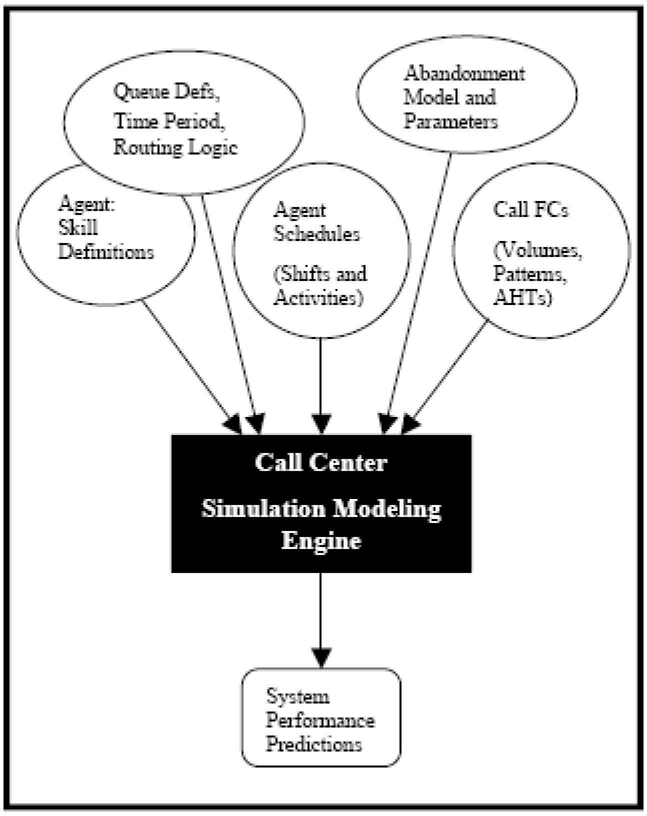


Fig. 1 Call Center Simulation Modeling Engine

In order to applying the algorithm, simulation must use online in every period, and tries to find best matrix that result the maximum value, then contact center must use this matrix for next period.

2.4 Stochastic Models

As part of the long research tradition, it has been standard to use stochastic models, especially queuing models. The workhorse queuing models have been the Erlang models, known as (M / M / s) in the standard Kendall queuing notation. The most common extensions considered attempt to account for customer abandonment, customer retrials, non-exponential call-holding-time distributions and time-varying arrival rates, but even these familiar phenomena pose serious analysis challenges.

Largely stimulated by the rapid growth of contact centers, the more academic published research literature on contact centers is now available. It can be seen from the recent survey on queuing models of call centers by Koole and Mandelbaum [5], the longer tutorial paper by Gans, Koole and Mandelbaum [6] and research bibliography by Mandelbaum [7]. Nevertheless, much remains to be done. This paper, however, uses Erlang-A model (M / M / s + M) for simulation.

2.5 Genetic Algorithms

Evolutionary algorithms (EAs) are stochastic search methods that have been successfully applied in many search, optimization, and machine learning problems. Unlike most other optimization techniques, EAs maintain a population of tentative solutions that are competitively manipulated by applying some variation operators to find a satisfactory, if not globally, optimum solution. Among the well-accepted subclasses of EAs, genetic algorithms (GAs) have been widely studied [8].

An implementation of genetic algorithms begins with a population of (typically random) chromosomes. One then evaluates structures and allocates reproduction opportunities in such a way that those chromosome witch represent a better solution to the target problem are given more chances to “reproduce” than those chromosomes witch are poor solutions. The “goodness” of a solution is typically defined with respect of current population [9].

Genetic algorithm can simply show as below:

- Generate a set of random solutions (Population)
- Repeat
  - Test each of them with fitness function (Evaluation)
  - Remove, duplicate, modify or mutate solutions (Reproduction)
- Until best solution is good enough

3 ROUTING ALGORITHM

3.1 The scenario

In proposed simulation, (M / M / s + M) model is implemented. Suppose that there are some agent pools, with same skill and value for every skill, and make the model large. The large size makes it possible to apply deterministic fluid approximations, which ignore all stochastic fluctuations [10]. Let assume mean service time (MST) and average maximum wait time be 3 minutes. Following scenario is implemented:

- Generate calls as Poisson process, and service time and maximum wait time as exponential random variable with mean MST.
- Set priority matrix same as value matrix (Direct VBR) for first period
- Repeat every periods (30 minutes)
  - Simulate contact center with corresponding priority matrix.
  - Suppose same load as last period happened.
  - Try to find best priority matrix that results best total value for next period using genetic algorithm.
  - Set result to priority matrix for next period.

3.2 Using genetic algorithms

Now using genetic algorithms can find near best priority matrix. Following algorithm is introduced with 4 phases:

- **Population**: Assume value matrix (direct VBR) and last best priority matrix (if not first period) and several random matrixes. Because direct VBR usually have good total value, following population can guarantee that total value is starting with good situation and the algorithm can work better than direct VBR.
- **Fitness Function**: Let set fitness function same as performance target that is defined. With this fitness function, performance target will maximize.
- **Evaluation**: In this phase, contact center is simulated for next period with same load as last period and is evaluated with fitness function.
- **Reproduction**: In this phase, the best-evaluated matrix (B) is selected and included in next population. Then other populations (M) are modified as following:
  - $M_{i,j}$ = $M_{i,j}$ or $B_{i,j}$ (with 50 percent probability)
  - $M_{i,j}$ = random (with mutation probability; 5%)

3.3 Pseudo Code

In this section, pseudo codes are shown for some of main parts of simulation:

- Generating Poisson random variable:

```
L = $e^{-\lambda}$
k = 0
p = 1
Repeat until p >= L
    k = k + 1
    R = random (between 0 and 1)
    p = p * R
Return (k – 1) as random Poisson variable
```

- Performance value function:

```
Val = 0
Repeat for any completed call
  If Call is abandon Then
    Val = Val + $c_{L,j}$
  Else If Wait Time > $y_j$ Then
    Val = Val + $c_{D,j}$
  Else
    Val = Val + $v_{i,j}$

Return Val
```

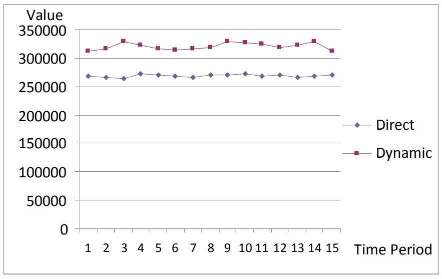
- Genetic search algorithm:

```
// Generate MaxPop populations
For i = 1 to MaxPop
   If i = 1 Then
      Chromosome[i] = Value matrix
   If i = 2 Then
      Chromosome[i] = Last preference matrix
   Else
      Chromosome[i] = Random matrix

// Genetic loop
Repeat MaxLoop time

   // Evaluation
   For any Chromosome
      Set Chromosome[i] as priority matrix
      Simulate Contact Center with last period calls
      Fitness = Performance Value
   Select max Fitness for BestPop

   // Reproduction
   For any Chromosome (k) except BestPop
      For any call type (j) and agent (i)
         If Random < 50% Then
            $Chromosome[k]_{i,j} = BestPop_{i,j}$
         If Random < Mutation Then
            $Chromosome[k]_{i,j} = Random$

Return BestPop as best founded priority matrix
```

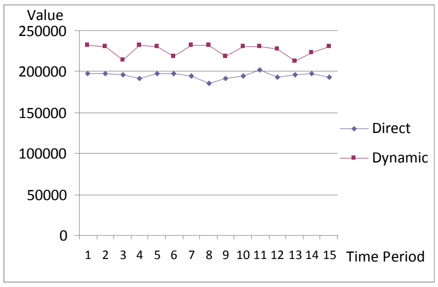
3.4 Samples and Result

In this section, some results are shown for some sample value matrixes. Total value is calculated according direct VBR and proposed genetic algorithm VBR for 8 hours (in every 30 minutes periods). First period is not included in graph. Graphs show that the algorithm result higher value than direct VBR:

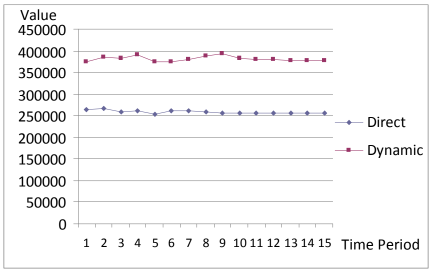
Sample 1:

Value Matrix: $\begin{bmatrix} 90 & 10 \\ 100 & 90 \end{bmatrix}$ λ = 360 Agent Pools = 200 (Total 400 agents) Direct VBR total value: 4033020 Our VBR total value: 4805080	

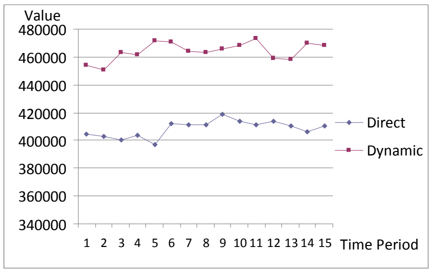
Sample 2:

Value Matrix: $\begin{bmatrix} 90 & 10 \\ 100 & 90 \end{bmatrix}$ λ = 360 Agent Pools = 150 (Total 300 agents) Direct VBR total value: 2925760 Our VBR total value: 3394620	

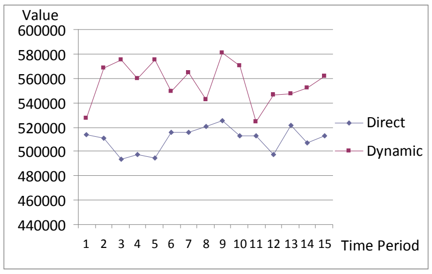
Sample 3:

Value Matrix: $\begin{bmatrix} 6 & 2 \\ 25 & 100 \end{bmatrix}$ λ = 720 Agent Pools = 400 (Total 800 agents) Direct VBR total value: 3875736 Our VBR total value: 5720059	

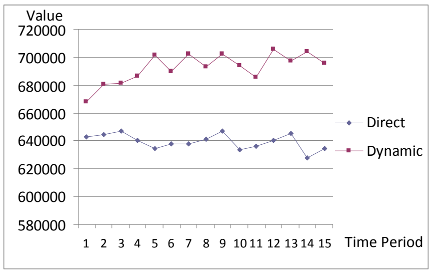
Sample 4:

$\begin{bmatrix} 80 & 70 & 30 & 25 \\ 75 & 75 & 20 & 15 \\ 95 & 95 & 85 & 75 \\ 90 & 95 & 80 & 90 \end{bmatrix}$ λ = 600 Agent Pools = 150 (Total 600 agents) Direct VBR total value: 6129605 Our VBR total value: 6962755	

Sample 5:

$\begin{bmatrix} 80 & 70 & 30 & 25 \\ 75 & 75 & 20 & 15 \\ 95 & 95 & 85 & 75 \\ 90 & 95 & 80 & 90 \\ 70 & 65 & 30 & 25 \\ 75 & 65 & 20 & 35 \\ 90 & 80 & 75 & 80 \\ 95 & 90 & 65 & 85 \end{bmatrix}$ λ = 900 Agent Pools = 100 (Total 800 agents) Direct VBR total value: 7651915 Our VBR total value: 8345300	

Sample 6:

$\begin{bmatrix} 80 & 70 & 30 & 25 \\ 75 & 75 & 20 & 15 \\ 95 & 95 & 85 & 75 \\ 90 & 95 & 80 & 90 \\ 70 & 65 & 30 & 25 \\ 75 & 65 & 20 & 35 \\ 90 & 80 & 75 & 80 \\ 95 & 90 & 65 & 85 \end{bmatrix}$ λ = 900 Agent Pools = 120 (Total 960 agents) Direct VBR total value: 9591000 Dynamic VBR total value: 10388630	

The samples show that proposed algorithm works more efficient than direct VBR in various cases. In addition, they show that when agents are more than λ, the expected values have fewer differences in different periods. However, in the cases that agents are less than λ, there are significant differences in different periods, and system seems too dynamic, but the algorithm is more efficient in dynamic cases too.

4 VALIDATION

In this section, linear programming is used to show the algorithm results are acceptable. Assume that there are m call types and n agents, to reduce the importance of stochastic fluctuations, let make the model large. The large size makes it possible to apply deterministic fluid approximations, which ignore all stochastic fluctuations. The fluid approximations justify the approximate performance descriptions given below, but the approximations should be convincing directly [10].

Suppose there are p/n agents in each agent pool (totally p agents), and all call service times are independent and identically distributed (i.i.d.) exponential random variables with mean 3 minutes. Calls of the call types arrive according to independent Poisson processes with arrival rates λ/m (calls per mean service time). Thus type-I calls have offered load λ/m, so that the total offered load is λ. In addition, assume that waiting customers may abandon. Let the times to abandon be i.i.d. exponential random variables with mean 3 minutes.

Thus, the total system is an Erlang-A model (M/M/s + M), since the individual customer abandonment rate equals the individual service rate, the stochastic process representing the number of customers in the system, either waiting or being served, in the (M/M/s + M) model has the same probability law (finite-dimensional distributions) as the stochastic process representing the number of busy servers in the associated in infinite-server (M/M/∞) model (with same arrival process and service-time distribution). Since a Poisson distribution with a large mean is approximately normally distributed, the steady-state number of customers in the (M/M/s + M) system is approximately normally distributed with mean λ. Since the variance of a Poisson distribution equals its mean, the standard deviation is $\sqrt{\lambda}$. Suppose p is more than 6 standard deviations above the mean, so the steady-state probability a customer is delayed is negligible. Also suppose λ < p so calls are only delayed very rarely, the vast majority of all calls are answered immediately upon arrival.

Now let $\lambda_{i,j}^{opt}$ be the rate (again per mean service time) that agents from pool i are processing type-j calls. This yields an approximate total value of

$$V^{opt} = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_{i,j}^{opt} v_{i,j}$$

Also because every call types arrive with rate λ/m and every agent pool can accept at most p/n calls, the following limitations are existed:

$$\sum_{i=1}^{n} \lambda_{i,j}^{opt} = \frac{\lambda}{m}$$

$$\sum_{j=1}^{m} \lambda_{i,j}^{opt} \leq \frac{p}{n}$$

$$\lambda_{i,j}^{opt} \geq 0$$

Now linear programming can be used to find $\lambda_{i,j}^{opt}$ values that maximize V^{opt}. However, the important issue is that it cannot find preference matrix that yields this call rate matrix. Therefore, this maximum value may not reach with any preference matrix, but our goal is near it. Now, let apply the linear programming for some of samples in 4.4 that met requirements (λ < p):

- **Example 1:**

After resolving linear programming, it could find that

$$\lambda^{opt} = \begin{bmatrix} 160 & 0 \\ 20 & 180 \end{bmatrix}$$

$$V^{opt} = 32600$$

Also easily can show that direct VBR yields following rate matrix (because agent 2 always have higher preference)

$$\lambda^{D} = \begin{bmatrix} 80 & 80 \\ 100 & 100 \end{bmatrix}$$

$$V^{D} = 27000$$

$$V^{opt} / V^{D} \approx 1.207$$

Because simulation is used for 8 hours and first period (30 minutes) is excluded from statistics, so the total value that is extracted in 450 minutes is 150 service times, so V^{opt} = 32600 * 150 = 4890000 and V^{D} = 27000 * 150 = 4050000 and these values are very near values that was calculated in simulation and it shows that the simulation is validated for direct VBR and reaches near optimized answer.

- **Example 3:**

After resolving linear programming it could find that

$$\lambda^{opt} = \begin{bmatrix} 320 & 0 \\ 40 & 360 \end{bmatrix}$$

$$V^{opt} = 38920$$

Also easily can show that direct VBR yields following rate matrix (because agent 2 always have higher preference)

$$\lambda^{D} = \begin{bmatrix} 160 & 160 \\ 200 & 200 \end{bmatrix}$$

$$V^{D} = 26280$$

$$V^{opt} / V^{D} \approx 1.48$$

Because simulation is used for 8 hours and the first period (30 minutes) is excluded from statistics, so the total value that is extracted in 450 minutes is 150 service times, so V^{opt} = 38920 * 150 = 5838000 and V^{D} = 26280 * 150 = 3942000 and these values are very near values that was calculated in simulation and it shows that the simulation is validated for direct VBR and reaches near optimized answer.

- **Example 6:**

The Excel solver has used for resolving linear programming:

$$\lambda^{opt} = \begin{bmatrix} 120 & 0 & 0 & 0 \\ 0 & 120 & 0 & 0 \\ 0 & 0 & 120 & 0 \\ 0 & 0 & 0 & 120 \\ 0 & 75 & 0 & 0 \\ 105 & 0 & 0 & 0 \\ 0 & 0 & 105 & 15 \\ 0 & 30 & 0 & 0 \end{bmatrix}$$

$$V^{opt} = 71775$$

In this example, it is too hard to find V^{D} and we can only find V^{opt}. Because simulation is used for 8 hours and the first period (30 minutes) is excludes from statistics, so the total value that is extracted in 450 minutes is 150 service times, so V^{opt} = 71775 * 150 = 10766250 and this value is near value that the simulation was calculated.

5 CONCLUSION

It has been shown how genetic algorithms can use to implement value-based routing (VBR), which aims to assign calls to agents to achieve the maximum expected value, subject to constraints ensuring that traditional congestion constraints are still met. Expected value might represent expected revenue or the likelihood of first-call resolution. Value might also reflect satisfying agent call-handling preferences. Indeed, there might be some composite value that reflects all these factors.

The result of research shows that direct VBR is not guarantee maximum expected value, and in many cases indirect VBR has more efficiency. The simulation is implemented to verify the proposed algorithm works properly. Direct VBR is included in starting population to ensure that the proposed algorithm works more efficient than (or equal if direct VBR is maximized answer for given value matrix) direct VBR. Finally, linear programming is used to show the algorithm work satisfactory. The linear programming can find best call rate matrix (per agent per call type), but cannot easily find corresponding preference matrix (if it exists), but the samples show that proposed algorithm can quiet close to the maximum expected value.

The main advantage of proposed algorithm is that it does not need to resolve complex linear programming, and easily can deploy for any value matrix, while most other algorithms, can only deploy for the special matrixes and need the complex linear programming implementation. In addition, the algorithm is suggested preference matrix for next period time, so it can work parallel with traditional call centers that support PBR, and only need to modify preference matrix in every period.

6 REFERENCES

1. W. Whitt, "Stochastic Models For The Design And Management Of Customer Contact Centers: Some Research Directions", Columbia University, 2002
2. V. Mehrotra, J. Fama, "Call center simulationmodelling: methods, challenging, and opportunities", Proceedings of the 2003 Winter Simulation Conference
3. M. Sisselman (New York, NY), W. Whitt (Columbia University) , "Value-Based Routing and Preference-Based Routing in customer contact centers", Producation and Operations Management, 2004
4. A. Brigandi, D. Dargon, M. Sheehan, T. Spencer III, "AT&T's call processing simulator (CAPS): operational design for inbound call centers. Interfaces", doi: 10.1287/intel.24.1.6 Interfaces January/February 1994 vol.24 no.1 6-28
5. G. Koole, A. Mandelbaum, "Queueing models of call centers: an introduction", Columbia University, Oct 9, 2001
6. N. Gans, G. Koole, A. Mandelbaum, "Telephone call centers: A tutorial and literature review", Computer Access and Internet Use, Columbia University, Sep 2, 2002
7. A. Mandelbaum, "Call Centers (Centres): Research bibliography with abstracts", Faculty of Industrial Engineering and Management, Technion, Israel Institute of technology, Dec 23, 2004
8. E. Alba, F. Luna, Antonio J. Nebro, "Advanced in parallel heterogeneous genetic algorithms for continuous optimization", Int. J. Appl. Math. Comput. Sci., 2004, Vol. 14, No. 3, 317–333
9. D. Whitley, "A Genetic Algorithm Tutorial", Computer science department, Colorado State University, 1994
10. W. Whitt, "A multi-class fluid model for a contact center with skill-based routing", International Journal of Electronics and Communications (AEÄU) 60 (2), 95-102, 2006