

Aplicação do método Monte Carlo na Distribuição de clientes em ambiente de “Help Desk”

Lilian Fernanda Castilho de Almeida¹, Vanessa de Lima Takaoka², Prof. Dr. Roosevelt Alves da Silva³

^{1, 2, 3} Universidade do Vale do Paraíba – Faculdade de Ciência da Computação
Av. Shishima Hifumi, 2911 CEP 12244-000 São José dos Campos – SP – Brasil
E-mail: [⁽¹⁾lfca_fho, ⁽²⁾vanessatakaoka] @yahoo.com.br, ⁽³⁾roosilva@univap.br

Resumo – O presente projeto destina-se ao desenvolvimento de um software capaz de solucionar o problema de distribuição de novos clientes no setor de “*help desk*” de uma empresa, selecionando os melhores atendentes, bem como a formação de equipes especializadas para atendimento de cada novo cliente. O sistema leva em consideração as características de cada integrante com relação ao perfil cadastrado dessas novas empresas. O software usa o Monte Carlo para amostrar o conjunto de possibilidades entre equipe e cliente e maximizar o valor de uma função que mede o grau de adequação entre equipe e cliente.

Palavras-chave: Java, Método de Monte Carlo, help desk, empresas.

Área do Conhecimento: Ciência da Computação.

Introdução

Todas as pessoas e certamente várias empresas já passaram pelo aborrecimento de serem mal atendidas durante uma ligação para um suporte do tipo “*help desk*” de qualquer empresa. Ou ainda a pessoa que realizou o atendimento não possuía as qualificações adequadas para resolver o problema e sanar suas dúvidas.

Atualmente, muitos empreendimentos trabalham com o sistema “*help desk*”, realizando atendimentos por e-mails, via Chat e mais comumente por telefone.

Existem ainda empresas que trabalham com o atendimento aleatório, ou seja, qualquer pessoa independente do perfil que possui realiza o atendimento para qualquer cliente, como em uma central telefônica, por exemplo. Entretanto, há uma grande gama que trabalha com a especialização do atendimento, alocando equipes e analistas para cada cliente vinculado a empresa.

Para essa população ainda podemos constatar outros problemas para chegar a uma seleção “ótima” (ideal) para os atendimentos, tais como:

- Como selecionar a melhor equipe para atender cada cliente?
- Como selecionar os atendentes que formarão cada equipe?
- Na definição dessas equipes, quem deverá atender o novo cliente?
- O que deve ser levado em consideração para selecionar essas pessoas e

conseqüentemente na definição das equipes?

Problemas como este pode envolver diversas soluções que muitas vezes requer uma rápida resposta para maximizar a satisfação dos clientes. Por exemplo, variáveis tais como a experiência do atendente, a quantidade de clientes acumulados para uma atendente, média de atendimentos, nível de stress do cliente ou atendente, experiência, etc, são difíceis de serem percebidas e mensuradas em tempo hábil por um departamento competente para realizar a distribuição.

Métodos de otimização como o algoritmo genético e Monte Carlo, permitem amostrar o conjunto de possibilidades ou configurações do sistema e selecionar aquelas de maior importância de acordo com o modelo considerado. Em particular, neste problema de *help desk*, a determinação rápida de uma solução capaz de promover a satisfação entre cliente e equipe é uma necessidade de diversas empresas e que pode facilmente ser aplicado com o sistema aqui apresentado.

Materiais e Métodos

Para o estudo da problemática foi utilizada uma empresa de desenvolvimento de Software na área de Gestão de Escolas e Bibliotecas. Cada cliente após contratar um dos produtos da empresa, é encaminhado a um determinado analista do Suporte conforme o produto adquirido, para receber assistência técnica e suporte sobre a usabilidade do Software.

Atualmente esta “distribuição” do cliente para cada analista é realizada apenas verificando a quantidade que cada analista possui de clientes para atender, assim o técnico que tiver menor número de clientes, receberá o novo cliente que está ingressando na empresa.

Todavia, esta solução empregada não é a mais adequada, pois outros aspectos importantes não são levados em consideração, como:

- Valor do contrato de manutenção que o cliente irá pagar;
- Número de matrículas contratadas para o Sistema;
- Classificação do cliente (prioritário, especial ou normal);
- Experiência do analista;
- Número de clientes que cada analista possui;
- Características do cliente (leigo em Informática, analista, crítico, cordial, etc.).

Há também uma variação entre as equipes do Suporte Técnico nos aspectos que devem ser analisados para a distribuição do cliente, isto torna ainda mais complexo criar uma solução para o problema, visto que um mesmo atributo (característica) pode ser extremamente importante para uma determinada equipe, e para a outra, isto pode ser totalmente irrelevante.

Cada analista possui uma carteira de clientes – terminologia empregada para denominar o conjunto de clientes sobre responsabilidade de atendimento do analista, esta carteira irá determinar a quantidade de clientes que cada analista irá atender.

A solução que foi proposta é empregar o desenvolvimento do Algoritmo Ótimo para solucionar este problema, onde o usuário pode entrar com os dados iniciais referentes ao cliente, e o programa efetuará o processamento destas informações para indicar qual o melhor analista que deverá ficar responsável pelo atendimento, baseado nos dados fornecidos inicialmente.

Após diversas pesquisas, constatamos que o problema teria uma solução mais apropriada utilizando o método de Monte Carlo. Já que o método é uma forma de resolver problemas usando números aleatórios.

Especificamente o método explora as propriedades estatísticas dos números aleatórios para assegurar que o resultado correto é computado da mesma maneira que num jogo de casino para se certificar de que a “casa” sempre terá lucro. Por esta razão, a técnica de resolução

de problemas é chamada de método de Monte Carlo.

Para resolver um problema através do método de Monte Carlo é usada uma série de tentativas aleatórias. A precisão do resultado final depende em geral do número de tentativas. Esse equilíbrio entre a precisão do resultado e o tempo de computação é uma característica extremamente útil dos métodos de Monte Carlo. Se queremos somente uma solução aproximada, então um método de Monte Carlo pode ser bastante rápido.

Como o projeto está alicerçado em aplicação de Algoritmo Ótimo será utilizada a Linguagem JAVA e JDBC para comunicação com Banco de Dados Firebird.

Resultados e Discussão

O Programa foi desenvolvido utilizando dois tipos de análises efetuadas pelo algoritmo:

- **Somente por equipe:**
Nesse caso, o algoritmo analisa com base nos atributos qual é a melhor equipe para atender o cliente, evitando sobrecarga de trabalho sobre uma mesma equipe.
- **Por equipe e atendente:**
Desta forma, o algoritmo analisa com base nos atributos qual é a melhor equipe e o atendente desta equipe que ficará responsável para ser responsável pelo atendimento do cliente ingressante.

A tela de cadastro de clientes (figura 1) foi criada com campos que procuravam captar as principais informações que deveriam ser levadas em consideração no momento da distribuição do mesmo, conforme os requisitos apurados com a empresa que utilizamos como base nesse projeto.

A imagem mostra a interface de usuário para o cadastro de um cliente. O título da janela é "Dados do Cliente" e o subtítulo é "Informações para análise da distribuição do cliente". Os campos de formulário incluem:

- Nome do Cliente: Instituto de Educação São Boa Ventura
- Sistema: SophiA Gestao Escolar
- Quantidade de Alunos: 3500
- Valor do Contrato: 2000.00
- Classificação: Prioritário (selecionado), Especial, Normal
- Estrutura Geral: Pequeno Porte, Médio Porte, Grande Porte (selecionado)
- Características do cliente: Conhec. em Informática: Conhecimentos avançados; Comportamento humano: Sensato

Botões "Confirmar" e "Cancelar" estão visíveis na base da janela.

Fig. 1- Tela de Cadastro dos Dados do Cliente

Ao finalizar o cadastro de um novo cliente, o software abre a tela de "Definição do critério de distribuição" (figura 2), onde o usuário deve configurar os parâmetros desejados para o cálculo e obtenção do "Ótimo".



Fig. 2 – Tela de Seleção e Atribuição dos Pesos

Com base nesses parâmetros o sistema irá realizar o cálculo da F(x) (função máxima), a qual é uma função crescente, como mostra o gráfico abaixo (Figura 3) e em seguida o cálculo da mesma:

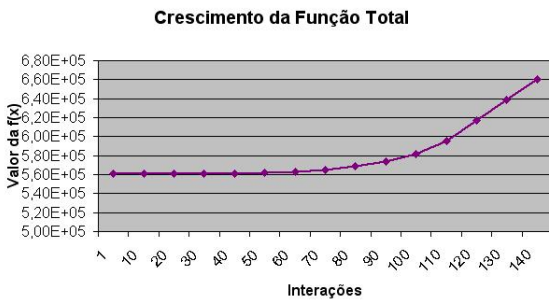


Fig. 3 - Gráfico de Crescimento da Função

$$\begin{aligned}
 fx = & ((pri_class/5) * p_classificacao) + \\
 & ((pri_nrAlu/5) * p_nrAlunos) + \\
 & ((pri_sw/5) * p_sw) + ((pri_manut/5) * \\
 & p_valorManut) + ((pri_perCli/5) * \\
 & p_perfCli) + ((pri_perAten/5) * \\
 & p_perfAten) + ((pri_nrCliAten/5) * \\
 & p_nrCliAten) + ((pri_expAte/5) * \\
 & p_expAtendente);
 \end{aligned}$$

onde para cada valor da tela de parâmetros, é dividido por 5 (valor máximo das prioridades) e multiplicado pelos pesos informados, logo todos são somados, nos resultando a função máxima.

O gráfico apresenta o crescimento dessa função, a qual ao final dos cálculos nos determina a escolha da equipe com seus melhores atendentes para o cliente informado na primeira tela de cadastro do sistema.

Com a aplicação do método Monte Carlo e da perturbação sobre todos os atributos das equipes, o algoritmo processa de modo recursivo até identificar qual seria a equipe ideal para atender o novo cliente. A equipe eleita é ilustrada através da figura 4 onde o patamar superior indica que a melhor solução fora alcançada:

Crescimento da função da equipe B

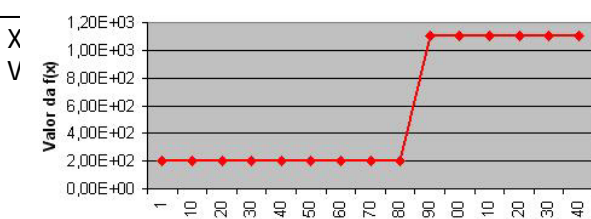


Fig. 4 – Gráfico da função da equipe ótima processada pelo método Monte Carlo

Todos os dados informados são gravados em um banco de dados, pois conforme a obtenção de clientes para cada atendente, o programa é capaz de mensurar as novas informações e analisar do mesmo modo os dados dos atendentes a cada inserção de novos clientes.

Na figura 5 mostramos o modelo entidade relacionamento (MER) do banco de dados do programa:

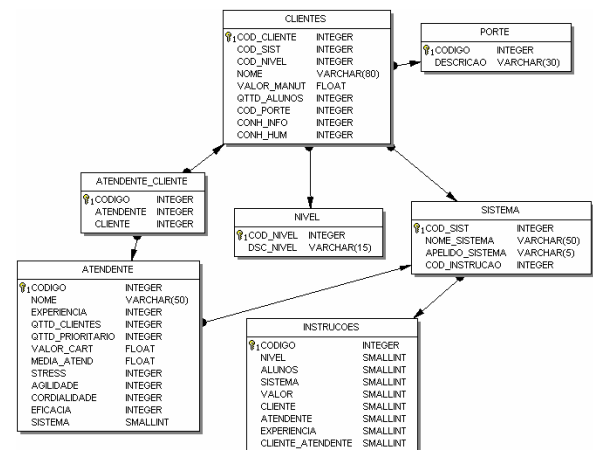


Fig. 5 – MER do Banco de Dados do Sistema

A seguir pode ser visto parte da codificação utilizada para a conexão via JDBC com o banco de dados Firebird:

```

url="jdbc:firebirdsql:localhost/3050:C:\\
ProjetoVII\\BD\\PROJETOVII.GDB"; //url de
conexão
try
{
    Class.forName("org.firebirdsql.jdbc
.FBDriver");
}
catch (ClassNotFoundException e) {
    System.out.println("Unable to
load Driver Class");
return;
}

```

onde: `url` é a variável que contém o driver de conexão juntamente com o caminho onde está armazenado o Banco de dados. Com o comando `Class.forName("org.firebirdsql.jdbc.FBDriver")` criamos a conexão via JDBC e na emissão de algum erro uma exceção é levantada através do `catch (ClassNotFoundException e)` informando que não foi possível carregar o driver de conexão.

O "loop" de perturbação realizado em cada equipe e por sua vez em cada atendente característico do Monte Carlo, está representando abaixo:

```
/* Início da simulação para identifica a
equipe ótima*/

for (n=1;n<=3;n++){/*até 3 por causa do
número de equipes*/

    mm=(int)(Math.random() * 2.9999) +
1; //Equipe sorteada
    jj=(int)(Math.random() * 3.9999) +
1; //Individuo sorteado
    perturbacao=0;//Inicia perturbação

    while (perturbacao==0){

        candidato=(int)(Math.random() *
999.999) + 1;
        if (INDICE[candidato]==0){
            perturbacao=1;
        }
    }
}
```

Onde para cada equipe e atendentes sorteados, essa parte do algoritmo realizará os testes de realizar trocas entre os mesmos até encontrar a melhor solução.

Ao final do processamento do algoritmo, o sistema nos retorna a melhor equipe e o melhor atendente para aquele cliente (Figura 6).

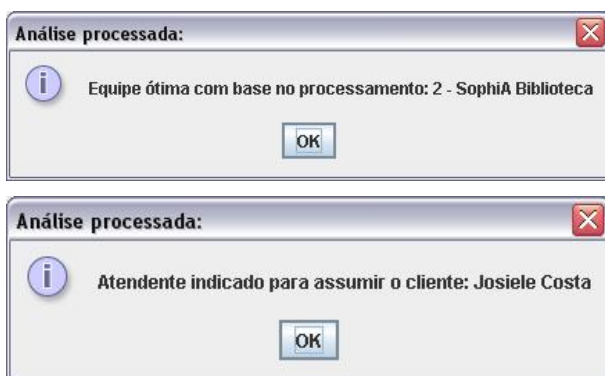


Fig. 6 – Apresentação dos resultados do algoritmo

Esta representação indica o tipo de modelagem efetuada pelo Modelo B, ou seja, análise por equipe e atendente, caso fosse escolhido apenas a análise sobre a equipe, o

Sistema iria retornar apenas o nome da equipe ideal para o atendimento, sem indicar o nome de um atendente em especial.

Com base na solução encontrada pelo algoritmo, o responsável pela distribuição do cliente possui uma ferramenta auxiliar na execução desta tarefa, indicando qual a equipe e analista que pode vir a atender o cliente.

Conclusão

Após a finalização das etapas de desenvolvimento do algoritmo e analisando-se os dados obtidos, conclui-se que os requisitos funcionais iniciais foram plenamente atingidos com sucesso. As técnicas de programação foram aprimoradas, possibilitando um bom resultado em relação às expectativas iniciais para o Sistema.

O Sistema pode ser aperfeiçoado para atender requisitos mais gerais e adequar-se a um conjunto de empresas que requerem decisões ótimas e rápidas nesta área de *help desk*. Ainda, o sistema poderia ter um módulo de interface gráfica para emitir gráficos/relatórios envolvendo o histórico das distribuições, dados estatísticos relacionados aos atributos de maior frequência, etc, de modo que o gerenciamento do setor *help desk* seja facilitado.

Referências

- PRADO, Darci. Teoria das filas e da simulação. Belo horizonte: Dg, 1999. 122 p (Pesquisa operacional; v.1)

- Método de Monte Carlo, em 14/04/07
<http://www.inf.ufrgs.br/proctpar/disc/cmp134/trabs/T2/021/html/index.html>.