



PROBLEMA DO CAIXEIRO VIAJANTE UTILIZANDO ALGORITMO GENÉTICO NO AMBIENTE EM NUVEM

HAAS, Alexander¹
PADILHA, Renata J.²
MONTEIRO, Taísa B.³
BECK, Matheus B.⁴

Resumo: A computação em nuvem é uma tecnologia que está em pleno desenvolvimento e é cada vez mais utilizada, tanto por pessoas, quanto por empresas, pode-se dizer que consiste no acesso de arquivos, aplicações e serviços, que não estão armazenados localmente, fazendo o acesso aos mesmos por meio da internet. Uma das grandes vantagens da *cloud computing* está ligada ao fato de não haver necessidade de investimentos dispendiosos na aquisição de *hardware* e *software*. Assim, cabe ao fornecedor do serviço encarregar-se das tarefas, antes executadas pelo usuário, de armazenar, manter, atualizar, fazer *backup*. Apesar de todos os benefícios propiciados pela computação em nuvem, existe uma grande preocupação no que diz respeito à segurança, uma vez que não é possível precisar onde os dados armazenados pelo cliente estão, nem no que diz respeito à privacidade e ao sigilo, considerando que estes dados estão em computadores de terceiros, bem como o envio destes dados até a nuvem, pois a rede de internet não foi desenvolvida para ser um meio de comunicação segura. O estudo objeto deste trabalho tem seu foco no funcionamento de uma nuvem, a disponibilidade dos recursos que um ambiente em nuvem pode oferecer, e qual sua capacidade de processamento para códigos que necessitam de um alto consumo de hardware para sua execução.

Palavras-chave: Computação em Nuvem. Algoritmo Genético. Caixeiro Viajante.

Abstract: *Cloud computing is a technology that is in full development and is increasingly used by both people, or companies, can be said that is the access files, applications and services, which are not stored locally, making the access to them through the internet. One of the great advantages of cloud computing is linked to the fact that there is need for expensive investments in the acquisition of hardware and software. It is therefore up to the service provider will take care of the tasks formerly performed by the user to store, maintain, upgrade, backup. Despite all the benefits provided by cloud computing, there is a concern with regard to safety, since it is not possible to specify where the data are stored by the client, not with respect to privacy and confidentiality, where as these data are in third computer, and sending these data to the cloud because the internet network is not designed to be a secure means of communication. The study object of this work is focused on operating a cloud, the*

¹ Alexander Haas, UNICRUZ – Fundação Universidade de Cruz Alta. E-mail: alexander.haas@hotmail.com.

² Renata Junges Padilha, UNICRUZ – Fundação Universidade de Cruz Alta. E-mail: renatajunges@yahoo.com.br.

³ Taísa Belzarena Monteiro, UNICRUZ – Fundação Universidade de Cruz Alta. E-mail: taisa.belzarena@hotmail.com.

⁴ Matheus Beck Brondani, UNICRUZ – Fundação Universidade de Cruz Alta. E-mail: theus_brondani@hotmail.com.



availability of resources to a cloud environment can offer and what it's processing capacity for codes that require a high consumption of hardware for execution.

Keywords: *Cloud computing. Genetic Algorithm. Traveling salesman.*

1. INTRODUÇÃO

A computação em nuvem é um recurso disponível, com uma capacidade de processamento e armazenamento quase que ilimitado, possuindo uma característica, fundamental para sua evolução, que é a elasticidade nestes processos, de acordo com a necessidade do usuário. Está em pleno crescimento no mundo e é cada vez mais utilizada por empresas e pessoas que não dispõem de *hardwares* com capacidade de processamento de dados de alto desempenho ou de um bom *storage* de dados com facilidades da acessibilidade. São vários os motivos que levam a implementação de uma nuvem, sendo que um deles é a indisponibilidade da parte física que deveria estar de acordo com a necessidade do usuário, assim seria necessário um alto investimento financeiro a ser realizado. Esta solução forneceria o que é preciso sem a necessidade de grandes investimentos e sem a necessidade de se preocupar com manutenção, atualizações e instalação do *hardware*, a única preocupação está no fato de concentrar apenas em seus trabalhos específicos e fazer a aplicação financeira no seu foco de trabalho.

A principal barreira, que retarda o crescimento da área, é a questão de quais são os processos que acontecem na nuvem, se os métodos utilizados de replicação, armazenamentos, acesso e distribuição são seguros. Verificando onde são armazenados os dados, se estão seguros contra ataques de *hackers*, problemas de *hardware*, desastres naturais, estes fatores são alguns dos motivos que podem fazer usuários desistir de utilizar a computação em nuvem.

A realização deste estudo tem como objetivo aplicar o algoritmo genético para a resolução do caixeiro viajante, utilizando um ambiente em nuvem para executar o código e verificar os resultados obtidos. Assim, verificar se ao utilizar um ambiente em nuvem para resolução de problemas que necessitam um alto consumo de hardware possui resultados satisfatórios.

2. REVISÃO DA LITERATURA

As subseções a seguir irão abordar o estudo realizado para o desenvolvimento da pesquisa.



2.1 Computação em Nuvem

Segundo WINKLER (2011) a computação na nuvem - *cloud computing* – é um dos resultados da rede de computadores, pois atualmente com a comunicação globalizada, disponibilizada pela internet, a mesma tornou-se cada vez mais viável e foi ganhando força e visibilidade. Basicamente, a *cloud computing* pode ser considerada como um conjunto de hardwares de alto desempenho que se encontram em um mesmo local ou em diferentes lugares no mundo, que estão conectados na rede e se comunicam por meio dela, e fornecem seus recursos de *hardware* para que usuários possam usufruir deles. Ela tem a função de armazenar e processar dados, com a disponibilidade deste serviço é possível fazer sua utilização do lugar em que seu usuário estiver apenas se conectando a internet, utilizando da capacidade de *hardwares* de alto desempenho para os serviços necessários, com uma capacidade de processamento e armazenamento muitas vezes superior à máquina disponível ao usuário.

Cloud Computing é modelo de Tecnologia da Informação (TI) ou ambiente de computação composto por componentes de TI (*hardware*, *software*, redes e serviços), bem como os processos de toda a implantação destes elementos que, juntos, nos permitem desenvolver e oferecer serviços em nuvem através da internet ou de uma rede privada (WINKLER, 2011).

Com este serviço sendo utilizado, não há necessidade de se preocupar com a capacidade do computador ou servidor para realizar tarefas que requeiram muitos recursos de *hardware*, as quais podem não estar disponíveis por causa de seu alto custo de aquisição e manutenção.

Bem, podemos dizer que a computação em nuvem é um termo para descrever um ambiente de computação baseado em uma imensa rede de servidores, sejam estes virtuais ou físicos. Uma definição simples pode ser “um conjunto de recursos como capacidade de processamento, armazenamento, conectividade, plataformas, aplicações e serviços disponibilizados na Internet” (TAURION, 2009).

A *cloud* oferece recursos que estão disponíveis conforme a necessidade, eliminando um custo desnecessário em servidores e computadores locais de alto desempenho e com sua manutenção. Estas máquinas, que muitas vezes, não tem utilizadas totalmente sua capacidade de *hardware*, o tempo inteiro, apenas em momentos de picos, podem ter sido utilizadas como



parâmetro para adquirir este *hardware* de alta capacidade, mas não levado em conta o tempo ocioso que ele se encontraria quando não estiver nestes momentos, um desperdício de dinheiro com sua manutenção e de recursos de *hardwares* (SOSINSKY, 2011).

A computação em nuvem possui algumas arquiteturas, que estão disponíveis aos seus usuários, cada uma com suas características.

2.1.1 Modelos de arquiteturas em Nuvem

O modelo *Software as a Service* (SaaS) oferece um serviço que utiliza os recursos de armazenamento e processamento através de uma interface simples, que tem como características o controle quase que total do fornecedor sobre o controle de gerenciamento da infraestrutura da nuvem, que não é autorizado ao seu usuário. A segurança se torna eficiente neste sentido, pois minimiza os riscos de segurança por parte do gerenciamento dos recursos da rede e dos softwares que estão disponíveis apenas para o fornecedor do serviço (CLOUD SECURITY ALLIANCE, 2009).

O modelo de arquitetura *Plataform as a Service* (PaaS) tem suas características a possibilidade de desenvolver e executar programas, que são implementados utilizando os serviços que estão disponíveis pelo provedor, o usuário não tendo autonomia sobre o controle da infraestrutura da nuvem, apenas utilizando de seus benefícios disponíveis (CLOUD SECURITY ALLIANCE, 2009).

E, também, o modelo *Infrastructure as a Service* (IaaS) oferece a autonomia de controle sobre o sistema operacional, como o armazenamento de dados, instalação de *softwares* e aplicações implementadas, possuindo um ambiente conforme as necessidades de seu usuário, mas tendo como risco a má utilização dos serviços pelo usuário e podendo comprometer o serviço disponível, em relação ao seu ambiente, pois este não possui acesso as configurações da infraestrutura da nuvem (CLOUD SECURITY ALLIANCE, 2009).

2.2 Algoritmos Genético

O algoritmo genético é inspirado no princípio Darwiniano da evolução tendo como teoria que apenas os mais fortes sobrevivem, ou seja, os mais aptos irão se reproduzir. O algoritmo genético que é utilizado para resolver problemas de busca e otimização, é



construído sobre este princípio, sempre buscando a melhor solução para o problema que for proposto, utilizando assim os meios de evolução de populações, selecionando os mais aptos para se reproduzirem (PACHECO,1999).

A implementação do algoritmo utiliza funções de aptidão, elitismo, seleção, cruzamento e mutação. A utilização destas funções é iniciada após a geração da população inicial dos cromossomos, cada cromossomo representa uma possível solução para o problema proposto, a primeira geração é aleatória para poder aplicar as funções do algoritmo genético (MALAQUIAS, 2006).

Com a população inicial aplica-se, a função de aptidão que determina o valor de cada cromossomo, esse valor é uma nota que é dada a cada um quanto melhor maior, sabendo-se assim quais são as melhores soluções que tendem a se reproduzir durante o processo do algoritmo. Com o valor de aptidão de cada cromossomo da geração, o elitismo tem como função selecionar o melhor desta geração, garantido assim que o melhor cromossomo irá se reproduzir na geração seguinte. Os métodos de seleção utilizados tendem a selecionar os melhores cromossomos de uma população, mas não descartando a possibilidade de selecionar os piores indivíduos, pois sempre se tem esta possibilidade mesmo que mínima, de selecionar estes com menores aptidões, e com isto evita-se que o algoritmo encontre apenas ótimos locais, e assim explore todas as possibilidades (MALAQUIAS,2006).

O método de seleção por roleta tem como princípio a mesma funcionalidade de uma roleta comum de uma agulha, que tem como objetivo sortear um cromossomo para ser utilizado na reprodução, para fazer este sorteio a roleta é dividida em fatias e cada fatia representa um cromossomo da população, o tamanho desta fatia é dada pelo valor de aptidão de cada um, ou seja, quanto melhor for sua aptidão maior a sua fatia, conseqüentemente maior a probabilidade de ele ser selecionado, e assim os cromossomos com aptidões ruins também possuem chance de serem selecionados e se reproduzir (MALAQUIAS,2006).

Para reprodução dos cromossomos são utilizados os métodos de cruzamento e mutação. O cruzamento (*crossover*) é um método que tem como objetivo realizar o cruzamento entre dois cromossomos para gerar um novo que irá ocupar uma posição na nova geração, esta função é feita a partir de um ponto de corte nos cromossomos selecionados cortando em uma determinada posição os indivíduos e juntando as partes com as do outro indivíduo gerando os novos cromossomos que serão salvos na próxima geração. A operação de mutação utiliza apenas um indivíduo que tem como objetivo realizar a troca de posições



dos genes do cromossomo, assim criando um novo que será armazenado na nova geração (MALAQUIAS,2006).

Por meio da população inicial gerada aleatoriamente, são criadas outras aplicando-se estes métodos sucessivamente, até que se chegue em um resultado aceitável para o problema, convergindo sempre para o melhor, isto sendo garantido pelo método de elitismo, o qual garante o melhor de cada geração na próxima (MALAQUIAS,2006).

2.2.1 Problema do Caixeiro Viajante

O problema do caixeiro viajante tem como objetivo encontrar o menor caminho, entre um conjunto de cidades com distâncias definidas, passando apenas uma vez em cada e retornando para a cidade de origem. O grande problema desta questão é o número de possibilidades de rotas que podem ser encontradas quando se tem um conjunto de cidades muito grande, necessitando de uma enorme quantidade de cálculos para se chegar no melhor caminho, tendo que o número de rotas possíveis é o número de cidades na fatorial (MALAQUIAS,2006).

Aplicar o problema com ajuda computacional não o torna simples, pois o número de caminhos possíveis a serem calculados mesmo por uma máquina são relevantemente altos, necessitando de um alto consumo de *hardware* da máquina utilizada para se chegar no melhor caminho possível que só pode ser encontrado ao se analisar todos caminhos possíveis, e, assim, comparar todos os caminhos gerados e ver qual é o melhor (MALAQUIAS,2006).

3. METODOLOGIA E/OU MATERIAL E MÉTODOS

O estudo em desenvolvimento classifica-se como qualitativo, pois vai explicar como ocorre os processos de comunicação na nuvem da *Google*, e quais são os seus protocolos, sua distribuição dos dados nos servidores que formam a nuvem, assim, possibilitando o alto desempenho ao acesso e armazenamento dos dados. Foram realizadas as seguintes etapas:

Etapa 1 – Estudo e implementação: Estudo teórico referente aos seguintes temas: a nuvem da *Google*; distribuição dos dados na nuvem da *Google* e seu armazenamento; formas de seguranças utilizadas na comunicação e no acesso a nuvem, e aos dados nela armazenada; alto desempenho disponível pela nuvem da *Google* como ele funciona, como é feito a



utilização dos recursos disponíveis para cada usuário, ferramentas disponíveis pela *Google* que utilizam sua nuvem; qual seria o propósito dessa distribuição no que aumentaria o seu desempenho ao acesso a eles; arquitetura utilizada na nuvem para a sua comunicação e distribuição dos dados; algoritmo genético e suas aplicações; métodos de seleção e suas limitações; utilização de *hardware* e capacidade computacional para problemas de altos processamentos; problema do caixeiro viajante; métodos desenvolvidos para solução do problema do caixeiro viajante.

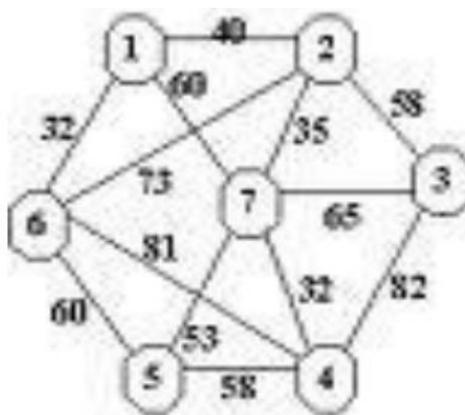
Etapa 2 - Validação: aplicação do teste caixa branca onde foi executado o método do algoritmo genético na geração do melhor caminho.

4. RESULTADOS E DISCUSSÕES

O programa foi desenvolvido na linguagem *Hypertext Preprocessor* (PHP) e utilizando o *Xampp Control Panel* como servidor para executar o código localmente. A máquina que executou o código localmente possui um processador Intel Core I7 -3610QM 2.30 GHz, 8GB de memória *Random Access Memory* (RAM). A execução do código em ambiente em nuvem aplicou-se com a ferramenta do *AppEngine* da *Google* que disponibiliza um limite de recursos para projetos serem testados utilizando a nuvem e recursos da *Google*.

Aplicou-se o algoritmo genético com o método de seleção da roleta com uma porcentagem definida de 70% mutação, 29% cruzamento e 1% elitismo, definida uma população de 100 cromossomos por geração. O grafo aplicado para realização dos testes (Figura 1) possui 7 vértices.

Figura 1. Grafo dos Vértices





O grafo foi traduzido em uma tabela de custos (Figura 2) para ser interpretado pelo código, e assim, por meio dela fazer o cálculo da distância de cada caminho gerado pelos cromossomos das diferentes gerações. Na tabela de custos pode-se observar que possuem valores altos que seriam as penalizações que são representadas pelo número mil, este valor é introduzido quando as cidades não possuem ligações entre elas, e quando se parte de um vértice e vai para ele mesmo, assim estas condições são penalizadas quando estão presentes em algum percurso. Conseqüentemente, o resultado daquela rota que foi penalizada terá um valor alto, pois além dos valores das distâncias entre os vértices terá mais mil incrementado no seu resultado o tornando pior, pois o objetivo é chegar ao menor caminho possível para percorrer todos os vértices. Em tempo de execução do programa também é verificado cada rota gerada, se a mesma possui uma cidade mais de uma vez no mesmo percurso, sendo assim penalizado no seu resultado com o incremento do valor mil.

Figura 2. Tabela de Custos

	0	1	2	3	4	5	6
0	1000	40	1000	1000	1000	32	60
1	40	1000	58	1000	1000	1000	35
2	1000	58	1000	82	1000	1000	65
3	1000	1000	82	1000	58	1000	32
4	1000	1000	1000	58	1000	60	53
5	32	73	1000	81	60	1000	1000
6	60	35	65	32	53	1000	1000

Fonte: Alexander Haas (2015).

O código foi desenvolvido possuindo funções específicas para tratar cada parte do processo do algoritmo genético. Foram utilizadas cinco funções iniciais (Figura 3), a primeira para gerar a tabela de custos da matriz, a segunda com a função de gerar a população inicial com cem cromossomos aleatoriamente, a função de mostrar ao usuário a geração criada aleatoriamente, e a função para calcular o valor do caminho de cada cromossomo da população gerada.



Figura 3 – Funções Iniciais

```
function custos_mat()  
{  
    global $matriz;  
  
function gera_cromossomo($tam_cam)  
{  
    global $matriz, $populacao;  
  
function salva_cromossomo($posi,$vet,$tam)  
{  
  
function mostra_cromossomos ($tam_cr,$pop,$cromo_pai)|  
{  
  
function calcula_caminho($cromossomo, $populacao, $n_vert, $num_geracao,$matriz)  
{
```

Fonte: Alexsander Haas (2015).

Após as funções iniciais, foram aplicadas as funções que executam os princípios do algoritmo genético como pode-se observar na Figura 4. A primeira função tem como objetivo realizar o cálculo de quantos cromossomos serão selecionados para o cruzamento, mutação e elitismo de acordo com as porcentagens definidas. A segunda função vai aplicar o elitismo que seleciona o indivíduo da geração no qual está sendo analisada e o salva na geração seguinte já garantido o melhor da geração para se reproduzir futuramente. Na função seguinte de cruzamento será selecionado dois cromossomos para serem cruzados e gerar novos que serão salvos na geração seguinte, para selecionar os cromossomos é aplicado o método de seleção da roleta. A função da mutação é selecionado um cromossomo da geração para mutar e gerar um novo salvando na geração seguinte e para selecionar o indivíduo a ser realizado a mutação é utilizado o método da roleta. A função da roleta define a roleta em si, o tamanho de cada fatia dos cromossomos da geração de acordo com os valores de sua aptidão. A última função tem como objetivo girar a roleta para sortear os cromossomos que serão cruzados e mutados, está é chamada dentro das funções de cruzamento e mutação.



Figura 4. Funções do Algoritmo Genéticos

```
function calcula_porcentagem($elite,$cruza,$muta,$pop,$cid)
{

function melhor_pai($tam_cam,$pop,$cromo_pai,$elite)
{

function cruzamentos_filho($tam_cam,$pop,$cruz,$elite,$cid)
{

function mutacao_filhos($tam_cam,$pop,$cromo,$muta,$cruza,$elite)
{
//*****FIM DOS CRUZAMENTOS E MUTAÇÕES*****

function roleta_peso($r_cromo,$r_pop,$r_vert)
{

function sorteia_cromossomo($cromo_peso,$s_pop,$s_vert)
{
```

Fonte: Alexander Haas (2015).

O código foi executado no ambiente em nuvem e na máquina local com o propósito de se obter as informações de tempo de execução para se chegar a um resultado aceitável, e quantidade de memória RAM utilizada para se chegar a tal resultado, e em qual geração foi encontrado.

Os resultados dos recursos utilizados quando executado em uma máquina local foram compatíveis com a realidade dos computadores de uso pessoal, utilizando-se pouco memória e processamento do mesmo, e chegando a um resultado em um tempo muito bom. Pode-se observar na Tabela 1 os resultados dos testes realizados na máquina local.

Tabela 1. Resultados Máquina Local

Máquina Local				
Teste Nº	Geração do Melhor	Valor do Melhor	Memória Utilizada (Máximo)	Tempo de Execução (s)
1	2	345	0,64 MB	0,01
2	4	345	0,76 MB	0,01
3	17	345	1,60 MB	0,06
4	67	345	4,80 MB	0,22
5	52	345	3,84 MB	0,17

Os resultados obtidos com o algoritmo sendo executado utilizando os recursos da nuvem foram satisfatórios, conseguindo-se chegar a um resultado cabível com um tempo de



execução baixo, e utilizando pouca memória. É possível observar na Tabela 2 os resultados dos testes realizados no ambiente em nuvem.

Tabela 2. Resultados da Nuvem

Ambiente em Nuvem				
Teste Nº	Geração do Melhor	Valor do Melhor	Memória Utilizada (Máximo)	Tempo de Execução (s)
1	42	345	3,12 MB	0,48
2	31	345	2,55 MB	0,22
3	68	345	4,79 MB	0,81
4	8	345	1,84 MB	0,08
5	9	345	1,14 MB	0,12

Como mostrado nas tabelas foram realizados cinco testes em cada plataforma, sendo que nos resultados utilizando uma máquina local mostrou-se levemente melhor que quando executado no ambiente em nuvem. Os valores das rotas foram os mesmos em ambos, as gerações no qual foram encontradas não são elevadas, a memória utilizada para a execução do algoritmo foi relativamente baixa nos dois ambientes, o tempo de execução também foi outro aspecto bem positivo, não levando nem um segundo para se chegar nos resultados.

5. CONSIDERAÇÕES FINAIS

Os resultados obtidos ao executar o código do algoritmo genético no ambiente em nuvem foram satisfatórios, pois a capacidade de processamento, utilização de recursos e o tempo de execução, equiparou-se a uma máquina local considerando que o ambiente utilizado para realizar o teste é gratuito com limites de recursos disponíveis diariamente.

Foi utilizado um grafo com poucos vértices pela questão do ambiente ter limites de recursos disponibilizados gratuitamente, sendo assim, com um grafo de sete vértices foi respeitado estes limites, podendo-se executar o código chegando nos resultados desejáveis. Para realizar testes com grafos maiores que necessitam uma utilização de *hardware* mais elevadas seria necessário pagar pelo que for utilizado dos recursos, o que não se pode fazer por questões financeiras.

Ao analisar que mesmo em um ambiente em nuvem com limites de recursos obteve-se bons resultados, pode-se concluir que para se executar códigos que necessitam de alta capacidade de *hardware* para poder chegar nas soluções dos problemas nos quais foram



aplicados não é necessário um supercomputador, pois o ambiente em nuvem disponibiliza estes recursos sem a necessidade de pagar um alto valor para utiliza-lo, paga-se apenas pelo que for utilizado, pois para a aquisição de um *hardware* com um poder computacional grande precisa-se de um alto investimento inicial e geralmente não é utilizada toda sua capacidade de processamento diariamente, assim uma das vantagens dos desenvolvedores que querem realizar testes e executar seus códigos não necessitam fazer um investimento desnecessário.

O problema do caixeiro viajante sendo aplicado em um grafo de apenas sete vértices teve os resultados aceitáveis como as rotas para serem percorridas, ao se aplicar o mesmo problema numa quantidade de mais de vinte vértices, a questão do tempo para se chegar a um resultado aceitável começa a se tornar inviável com a realidade se executado em máquinas não específicas para grande poder computacional, observando-se que com acesso a uma aplicação que não utilize os recursos locais para se chegar nos resultados desejados, torna-se viável conseguir-se uma solução de um problema rapidamente a partir de qualquer dispostos que possua acesso à internet que se comunique com o ambiente em nuvem no qual o programa está alocado para se conseguir os resultados.

Este estudo é parte integrante do trabalho de conclusão de curso de ciência da computação.

REFERÊNCIAS BIBLIOGRAFICAS

CLOUD SECURITY ALLIANCE. **Security Guidance for Critical Areas of Focus in Cloud Computing**. Abril 2009. Disponível em: < <https://cloudsecurityalliance.org/guidance/csaguide.v1.0.pdf>>. Acesso em: 14 out. 2014.

MALAQUIAS, Neli Gomes Lisboa. **Uso dos algoritmos genéticos para a otimização de rotas de distribuição**. Dissertação de Mestrado em Ciências, Universidade Federal de Uberlândia – UFU, 2006.

PACHECO, Marco Aurélio Cavalcanti. **Algoritmos Genéticos: Princípios e Aplicações**. Julho 1999. Disponível em: < www.ica.ele.puc-rio.br>. Acesso em: 30 mai. 2015.

SOSINSKY, Barrie. **Cloud Computing Bible**. 1ª ed. Indianápolis: Wiley, 2011.



XVII

Seminário Internacional
de Educação no MERCOSUL



www.unicruz.edu.br/mercosul

TAURION, Cezar. **Cloud Computing: Computação em Nuvem – Transformando o Mundo da Tecnologia da Informação**. Rio de Janeiro: Brasport, 2009.

WINKLER, Vic (J.R.). **Securing the Cloud: Cloud Security Techniques and Tactics**. 1ª ed. Waltham: Syngress, 2011.