



Faculdade de Ciências Exatas e Tecnologia – FAET
Curso de Bacharelado em Ciência da Computação

Liliane Frez da Silva

Otimização de tráfego viário através da
aplicação de Algoritmos Evolutivos.

Brasília
2006

LILIANE FREZ DA SILVA

OTIMIZAÇÃO DE TRÁFEGO VIÁRIO ATRAVÉS DA APLICAÇÃO DE ALGORITMOS
EVOLUTIVOS

Monografia apresentada como requisito parcial do Projeto de Iniciação Científica, no Centro Universitário de Brasília, UniCEUB, DF.

Orientador: Prof. Osmar Quirino da Silva

Brasília
2006

LILIANE FREZ DA SILVA

**OTIMIZAÇÃO DE TRÁFEGO VIÁRIO ATRAVÉS DA APLICAÇÃO DE
ALGORITMOS EVOLUTIVOS**

Esta monografia foi julgada adequada para apresentação de trabalho científico através do PIC (Programa de Iniciação Científica) realizado pelo Curso de Ciência da Computação do Centro Universitário de Brasília, Brasília – DF.

Área de concentração: Inteligência Artificial

31 de agosto de 2006.

Prof. Osmar Quirino
UniCEUB
Orientador

RESUMO

O presente trabalho apresenta o estudo dos principais Algoritmos Evolutivos e suas aplicações. Esses algoritmos simulam o mundo natural e encontram-se na área da Inteligência Artificial. Os Algoritmos Evolutivos são métodos de busca estocástica baseados nos princípios da evolução biológica que abstraem e imitam alguns dos mecanismos evolutivos para resolução de problemas que requerem adaptação, busca e otimização.

Dentre os Algoritmos Evolutivos estudados, encontram-se o Sistema Imunológico Artificial (SIA), Otimização por Colônia de Partículas (OCP), Otimização por Colônia de Formigas (ACO) e Algoritmos Genéticos (AG). O SIA é o algoritmo que simula o sistema imunológico biológico e, portanto, possui grande destaque na área de segurança computacional, por exemplo, no combate a vírus. A OCP e a ACO são algoritmos que simulam o comportamento social entre indivíduos da mesma espécie. A OCP estuda a experiência que o indivíduo adquire ao longo da sua vida, vinculada a sua interação com outros animais, a busca pelo alimento, etc. A ACO diferencia-se por estudar especificamente o comportamento de busca de alimento das formigas. Estas utilizam o feromônio como ferramenta de localização e rastro. O AG baseia-se no Darwinismo, por se tratar de um processo evolucionário que utiliza técnicas de evolução e seleção natural. Os indivíduos pertencentes ao conjunto de soluções deverão passar por um processo de adaptação, e os melhores selecionados representarão as melhores soluções. Através do estudo e fundamentação dos Algoritmos Genéticos, é proposto o desenvolvimento de uma solução para o problema de tráfego urbano em rotas arteriais. O descontrole do tráfego urbano é um problema que, hoje, atinge boa parte das grandes cidades brasileiras, principalmente, devido ao aumento da quantidade de veículos nas vias. O desenvolvimento do software de simulação deste trabalho será feito com o intuito de implementar a otimização através do uso dos Algoritmos Genéticos para a solução do problema proposto. A solução proposta deverá ser capaz de gerar informações sobre a melhor rota a se seguir em um determinado momento, visando otimização, previsão e orientação do fluxo viário.

Palavras-chave: Algoritmos Evolutivos, Inteligência Artificial, Sistema Imunológico Artificial, Otimização por Colônia de Partículas, Otimização por Colônia de Formigas, Algoritmos Genéticos, tráfego viário e otimização.

ABSTRACT

The following work deals with a study concerning the main Evolutionary Algorithms. These algorithms make a simulation of the natural world and belong to the Artificial Intelligence field. Evolutionary Algorithms are a kind of stochastic search method, which are based on biological evolution principles. In order to solve problems that require adaptation, search, and optimization, these algorithms make an abstraction and a copy of evolutionary mechanisms.

Among Evolutionary Algorithms analyzed in this work, we can mention the following: Artificial Immune System (AIS), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Genetic Algorithms (GA). The AIS algorithm simulates the biological immune system having, thus, great importance within computers security field, such as in virus combat. The PSO and the ACO algorithms make a simulation of how individuals from the same specie behave within a social context. The PSO studies the experiences that each individual have acquired throughout its life, as well as its interaction with other animals, the search for food, among other things. However, the ACO algorithm makes a particular study on the ant's search for food, which uses pheromone as a localization mechanism and also to leave traces on the ground. The GA algorithm is based on Darwinism, once it is an evolutionary process that uses evolution and natural selection techniques. Those individuals who belong to the solution set shall pass through an adaptation process, and those who were best selected, will then represent the best solutions.

A proposal for the development of arterial routes as a solution for urban traffic problem is made upon a study of GA algorithms, likewise its backgrounds. Today, the uncontrolled traffic is a problem that exists in many Brazilian cities, mainly because the number of vehicles on the roads has increased in the last years. To solve this problem, a simulation software will be developed to put into practice traffic optimization by using Genetic Algorithms. The solution proposed in this work shall be capable of generating information on what is the best route to take, in order to optimize, forecast, and manage traffic flow.

Keywords: Evolutionary Algorithms, Artificial Intelligence, Artificial Immune System, Particle Swarm Optimization, Ant Colony Optimization, Genetic Algorithms, traffic, optimization.

LISTA DE FIGURAS

Figura 1.1: Exemplos de sistemas naturais. (a) Formigas (colônia de insetos), (b) Linfócitos (sistema imunológico).

Página: 11

Figura 2.1: Estrutura multicamada do sistema imunológico.

Página: 16

Figura 2.2: Representação do comportamento social das aves.

Página: 19

Figura 2.3: Fluxograma para o algoritmo OCP básico.

Página: 20

Figura 2.4: Formigas entre o formigueiro e o alimento.

Página: 22

Figura 2.5: Obstáculo entre o formigueiro e o alimento.

Página: 22

Figura 2.6: Melhor caminho escolhido, e portanto, com a maior quantidade de feromônio.

Página: 23

Figura 2.7: Esquematização de um cromossomo.

Página: 24

Figura 3.1: Estrutura básica de um AG simples.

Página: 30

Figura 3.2: Operador de cruzamento de um ponto de corte.

Página: 32

Figura 3.3: Operador de cruzamento de dois pontos de corte.

Página: 32

Figura 3.4: Operador de cruzamento uniforme.

Página: 33

Figura 3.5: Operador de mutação.

Página: 33

Figura 4.1: Grafo ilustrativo para um conjunto de locais e trechos.

Página: 41

Figura 4.2: Árvore com caminhos possíveis para B-H.

Página: 43

Figura 4.3: Diagrama de Classe de Domínio para o Sistema Simulador de Otimização de Tráfego Viário.

Página: 46

Figura 4.4: Diagrama de caso de uso para o Sistema Simulador de Otimização de Tráfego Viário.

Página: 47

Figura 4.5: Diagrama de Atividades Fluxo de Processamento de Informação dada pelo usuário (motorista).

Página: 48

Figura 4.6: Diagrama de Atividade Fluxo de Cálculo de Rotas.

Página: 49

LISTA DE TABELAS

Tabela 4.1: Tabela Trecho/Tempo.

Página: 42

Tabela 4.2: Conjunto de trechos possíveis da origem (B) ao destino (H) e o tempo total (TT) gasto para cada um dos caminhos.

Página: 42 e 43

Tabela 4.3: Dados pertencentes a cada trecho (T_c).

Página: 44

LISTA DE ABREVIATURAS

- ACO Termo que abrevia as palavras *Ant Colony Optimization*.
- ACS Termo que abrevia as palavras *Ant Colony Systems*.
- AE Termo utilizado para abreviar as palavras Algoritmos Evolutivos.
- AG Termo utilizado para abreviar as palavras Algoritmo Genético.
- IA Termo utilizado para abreviar as palavras Inteligência Artificial.
- OCP Otimização por Colônia de Partículas.
- OO Orientação a objetos.
- TT Tempo Total.
- UML Unified Modeling Language.

SUMÁRIO

1 – INTRODUÇÃO	- 11 -
2 - ALGORITMOS EVOLUTIVOS.....	- 16 -
2.1. Sistema imunológico artificial.....	- 16 -
2.2. Colônias de partículas.....	- 18 -
2.3. Colônia de formigas	- 20 -
2.4. Algoritmos Genéticos	- 23 -
3 - ALGORITMOS GENÉTICOS.....	- 27 -
3.1. Como os AGs funcionam	- 27 -
3.2. Operadores genéticos	- 30 -
3.3. Comparando AG com outros procedimentos de otimização	- 35 -
3.4. Otimização por Algoritmos Genéticos.....	- 36 -
4 – SIMULAÇÃO DE OTIMIZAÇÃO DE TRÁFEGO	- 39 -
4.1 Estudo de caso.....	- 39 -
4.1.1 O problema.....	- 39 -
4.1.2 Fundamentação teórica.....	- 40 -
4.1.3 Modelagem da solução	- 44 -
4.1.3.1 Especificação	- 45 -
5 – CONCLUSÃO E PERSPECTIVAS	- 50 -
GLOSSÁRIO	- 52 -
REFERÊNCIA	- 54 -
REFERÊNCIA BIBLIOGRÁFICA	- 56 -

1 – INTRODUÇÃO

Idéias extraídas de sistemas naturais vêm sendo utilizadas com muito sucesso para o desenvolvimento de ferramentas tecnológicas capazes de resolver problemas de complexidade elevada, cujas soluções eram, até então, desconhecidas ou inatingíveis. Dentre o mundo natural, o sistema que mais tem recebido atenção é o cérebro humano, devido a sua grande capacidade de processamento de informações. A Inteligência Artificial se desenvolveu em um estágio inicial de investigação acerca da inteligência humana, visando compreender como um cérebro de dimensões físicas reduzidas e consumo de energia limitado, poderia perceber, compreender, prever e manipular um mundo extremamente diversificado. A IA levou ao desenvolvimento dos computadores como são conhecidos atualmente, e vem sendo aplicada a problemas em diversas áreas, como, por exemplo, busca e otimização.

A natureza fornece muitos exemplos destes tipos de sistemas que são compostos por elementos simples, nos quais uma interação cooperativa e competitiva resulta num comportamento global complexo (SILVA L., 2001). Exemplos de sistemas complexos na Figura 1.1.

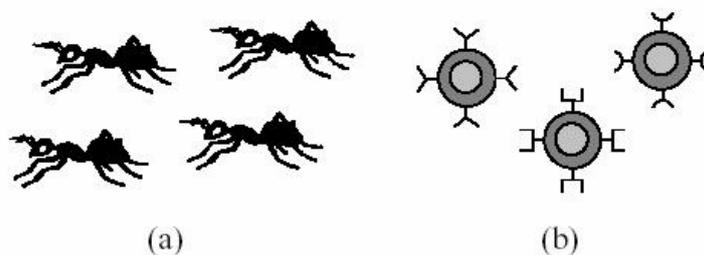


Figura 1.1: Exemplos de sistemas naturais. (a) Formigas (colônia de insetos), (b) Linfócitos (sistema imunológico).

Uma das principais características destes sistemas é a robustez, expressa sob a forma de uma tolerância a pequenas perturbações em componentes individuais. Esta robustez engloba o princípio da distributividade, em que

componentes (agentes) individuais do sistema contribuem de forma infinitesimal para o desempenho global (SILVA L., 2001).

Quando os diversos agentes operam em conjunto, comportamentos complexos podem emergir. Os sistemas vivos são, sem sombra de dúvidas, sistemas complexos. Estes sistemas podem ser vistos como conjuntos de objetos, agentes, elementos, ou processos, que exibem um comportamento dinâmico e agregado, em que a ação de um objeto possivelmente afeta as ações subseqüentes dos outros objetos, englobando o comportamento coletivo e sua interação com o ambiente (SILVA L., 2001).

Para simular estes comportamentos naturais que fazem parte do mundo biológico, foram descobertas várias técnicas que utilizam estes comportamentos para tornar possível o desenvolvimento de problemas ou soluções. Contribuindo para a eficiência e eficácia na resolução destes.

Estes métodos têm recebido crescente interesse nas últimas décadas, devido principalmente a sua versatilidade para a resolução de problemas complexos nas áreas de otimização e aprendizado de máquinas. O desenvolvimento de modelos computacionais, inspirados nos mecanismos evolutivos, caracteriza-se pela configuração de algoritmos de otimização robustos e sistemas adaptativos (COELHO, 2003).

Os Algoritmos Evolutivos (AEs) são métodos de otimização e busca estocástica baseados em princípios da evolução biológica natural. Os AEs representam métodos computacionais baseados em modelos naturais que simulam a evolução de sistemas via processo de seleção natural. Os princípios dos AEs derivam do “mundo biológico”, ou seja, buscam características da evolução dos seres. Os AEs não são algoritmos computacionais como sugere o nome, são princípios que tentam abstrair e imitar alguns dos mecanismos evolutivos à resolução de problemas que requerem adaptação, busca e otimização.

Então por que utilizar os algoritmos evolutivos? Existem várias razões:

- 1: são métodos de buscas estocásticas, ou seja, são métodos que baseiam-se em probabilidades;
- 2: tratam adequadamente os sistemas sujeitos a restrições;
- 3: não requerem as informações relativas a derivadas, estas são usualmente necessárias em métodos convencionais de otimização;

- 4: adequam-se à implementação em paralelo e distribuídas;
- 5: possibilitam a utilização do conhecimento obtido *a priori* pelo projetista;
- 6: tratam com sistemas complexos e espaços de busca com múltiplas modas e/ou múltiplos objetivos.

Um dos Algoritmos Evolutivos que vem crescendo ultimamente são os chamados Sistemas Imunológicos Artificiais (SIA). Atualmente o interesse pelo estudo do sistema imunológico biológico cresce radicalmente, pois o ser humano está buscando novas respostas que possam combater doenças que atingem, hoje, o mundo inteiro, como por exemplo, a AIDS. Mas as pesquisas no campo da imunologia não estão interessando somente aos profissionais de saúde, e também a engenheiros e cientistas da computação, que tentam simular mecanismos imunológicos particulares com o objetivo de criar sistemas artificiais para a solução de problemas de engenharia. Por exemplo, podem ser traçados fortes paralelos entre o reconhecimento de padrões do sistema imunológico e o reconhecimento de padrões em computação. Uma metáfora ainda mais abrangente poderia relacionar as características de imunovigilância e resposta imune com os procedimentos de segurança computacional, que podem incluir detecção e eliminação de vírus, intrusos de rede, e assim por diante.

Nessas áreas de engenharia e computação, tem surgido um forte interesse pelo estudo dos sistemas imunológicos devido, principalmente, à sua capacidade de processamento de informação.

Uma outra abordagem emergente dos Algoritmos Evolutivos que tem recebido muita atenção é o estudo do comportamento social entre indivíduos de uma mesma espécie, chamado de Otimização por Colônia de Partículas (OCP), conhecido também como PSO (*Particle Swarm Optimization*).

A OCP é uma metodologia proposta originalmente por Kennedy & Eberhart na década de 90 (KENNEDY & EBERHART, 1995), mais precisamente em 1995 e ainda é uma área nova de estudo, com poucas pesquisas. É também uma técnica de busca estocástica e de otimização, assim como os outros AEs.

O mais importante a se destacar na OCP é que ela simula o sistema social, a coletividade através de simples indivíduos interagindo com o seu meio e uns com os outros, um exemplo é o comportamento social das aves. A busca pelo alimento ou pelo ninho e até mesmo a interação com os outros pássaros durante o voo, são

modelados neste algoritmo como um mecanismo de otimização. Os pássaros, ou partículas, usam as suas experiências e a dos outros para encontrarem o ninho ou o alimento. Dessa maneira cada partícula acumulará uma experiência que posteriormente será utilizada para a resolução do problema.

Muito parecido com a OCP, encontra-se o algoritmo chamado de Otimização por Colônia de Formigas, *Ant Colony Optimization* (ACO). Esse algoritmo é baseado no comportamento das formigas, mais precisamente pela sua forma especial de busca de alimento, que diferenciam dos outros animais.

Foi desenvolvido inicialmente por Marco Dorigo em sua tese de PhD em 1992, como uma abordagem de múltiplos agentes para otimização de problemas combinatórios difíceis. Este algoritmo segue uma linha de raciocínio muito parecida com a otimização por colônia de partículas (OCP), pela intenção de simular o processo de busca de alimento das aves e o comportamento que elas têm com as outras e com o meio. O que diferencia um comportamento do outro é que as formigas têm um método de busca por alimento diferente do das aves. Esse diferencial é principalmente porque as formigas utilizam os chamados feromônios para rastrear e comunicar as outras formigas o caminho que está percorrendo para ir atrás do alimento.

Os Algoritmos Genéticos (AGs) também fazem parte dos algoritmos evolutivos, por apresentarem a simulação de um ambiente natural. Os AGs levam esse nome por representarem um mecanismo de evolução das espécies, baseado na teoria da evolução de Darwin, naturalista e fisiologista inglês que escreveu o livro “A origem das espécies”. Darwin disse em seu livro que para um indivíduo sobreviver a um meio, ele precisa se adaptar e assim conseguirá transmitir seus genes as gerações seguintes.

A idéia por trás dos algoritmos genéticos é fazer o que a natureza faz. A exemplo dos coelhos: imagine uma população de coelhos. Alguns deles são mais rápidos e mais espertos do que outros. Estes mais rápidos e espertos são menos propícios de serem comidos pelas raposas, e portanto muitos deles sobreviverão para fazer o que os coelhos fazem de melhor: fazer mais coelhos. Claro que, alguns dos mais lentos irão sobreviver só porque são sortudos. Esta população de coelhos sobreviventes irá procriar. A procriação resultará em uma boa mistura do material genético do coelho: alguns coelhos lentos procriam com os mais rápidos, os mais

rápidos com os mais rápidos, e assim por diante. O resultado será bebês coelhos mais rápidos e mais espertos do que a população original. Isso seria muito bom se as raposas não sofressem também o mesmo processo, caso contrário os coelhos ficariam rápidos e espertos demais para as raposas capturá-los (GOLDBERG, 1989).

Os AGs foram introduzidos por John Holland (HOLLAND, 1975) e popularizados por um dos seus alunos David Goldberg (GOLDBERG, 1989). Holland e alguns de seus colaboradores da Universidade de Michigan estavam interessados em sistemas artificiais complexos, capazes de adaptarem-se a mudanças de condições ambientais. Holland compreendeu que os mecanismos biológicos permitiam a adaptação (assim como nos coelhos) do sistema natural biológico de forma que poderiam ser expressas matematicamente, e simuladas computacionalmente. Assim nasceu o AG, que hoje é usado para resolver problemas de grande complexidade e principalmente de otimização.

2 - ALGORITMOS EVOLUTIVOS

2.1. Sistema imunológico artificial

Constituído por componentes e mecanismos distintos, porém que atuam de forma conjunta e notavelmente eficaz, o sistema imunológico proporciona ao corpo humano resistência às enfermidades. Os anticorpos, por exemplo, são gerados por células denominadas linfócitos em resposta aos antígenos (agentes infecciosos), e sua presença em um indivíduo reflete as infecções às quais o mesmo já foi exposto. Os linfócitos (veja na figura abaixo a estrutura do sistema imunológico) são capazes de desenvolver uma memória imunológica, ou seja, reconhecer o mesmo estímulo antigênico caso ele entre novamente em contato com o organismo, evitando assim o restabelecimento da doença. Portanto, mecanismos de aprendizagem e memória dão ao sistema imunológico a capacidade de extrair informações dos agentes infecciosos e disponibilizá-las para uso futuro em casos de novas infecções pelos mesmos agentes ou agentes similares (SILVA L., 2001).

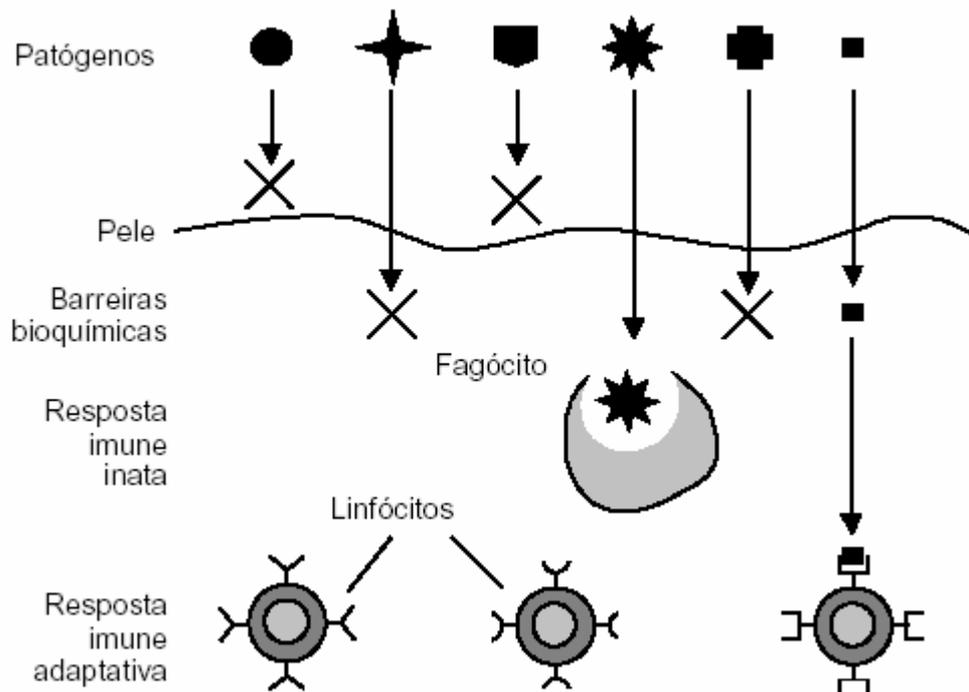


Figura 2.1: Estrutura multicamada do sistema imunológico.

Sob uma perspectiva de engenharia, existem diversas características do sistema imunológico que podem ser destacadas:

- Unicidade: cada animal possui seu próprio sistema imunológico, com suas capacidades e vulnerabilidades particulares;
- Reconhecimento de padrões internos e externos ao sistema: as células e moléculas que não pertencem ao organismo são reconhecidas e eliminadas pelo SI;
- Detecção de anomalia: o SI pode detectar e reagir a agentes patogênicos (causadores de anomalias) a que o organismo nunca havia sido exposto anteriormente;
- Detecção imperfeita (tolerância a ruídos): um reconhecimento perfeito não é necessário para que o SI reaja contra um elemento causador de patologia (*patógeno*);
- Diversidade: existe uma quantidade limitada de células e moléculas no SI que são utilizadas para se obter o reconhecimento de um número praticamente infinito de elementos, incluindo aqueles sintetizados em laboratório;
- Aprendizagem por reforço: a cada encontro com o mesmo patógeno, o sistema imunológico melhora a qualidade de sua resposta; e
- Memória: os componentes do SI bem sucedidos no reconhecimento e combate às patologias são armazenados para uma resposta futura mais intensa e efetiva.

Essas características despertaram a atenção de engenheiros e cientistas que buscam soluções para problemas computacionais, que hoje estamos vivenciando. As pesquisas nesse campo começaram há aproximadamente 15 anos e originaram um novo ramo da teoria de sistemas inteligentes, os sistemas imunológicos artificiais.

A diversificação, que permite ao sistema enviar simultaneamente vários anticorpos contra uma grande variedade de agentes infecciosos em diferentes partes do corpo humano, é outra habilidade biológica adaptada para a solução de problemas práticos. Um software com características operacionais similares, poderia ajudar a tornar mais eficiente e econômico o sistema de entregas de uma distribuidora de bebidas, por exemplo. A partir da análise de todos os dados envolvidos na operação, tais como número de caminhões disponíveis, distâncias a

serem percorridas, custo das viagens e tempo gasto nas entregas, o programa escolheria as rotas com a melhor relação entre custo e benefício.

A mesma ferramenta contribuiria para a produção de chips de computadores mais velozes que os atuais. O sistema otimizaria de tal forma o posicionamento dos micro-componentes do circuito que os bilhões de pulsos binários percorreriam distâncias cada vez menores no processamento de informações.

O sistema imunológico e a segurança em sistemas de computadores possuem um ponto em comum; proteger o sistema contra ataques e intrusões que violem a política de segurança estabelecida. Portanto, como mais uma aplicação para os sistemas imunológicos artificiais ficará o de detecção de intrusos (hackers) em uma rede de computadores e na constatação e eliminação de vírus computacionais, áreas em que a analogia entre proteção do organismo e segurança computacional é mais evidente.

Porém suas aplicações ainda esbarram no desconhecimento de particularidades do funcionamento do sistema biológico que são decisivas para o desenvolvimento e o êxito de mecanismos artificiais de proteção. Uma delas é a capacidade do sistema imunológico de distinguir as células e moléculas do próprio organismo, e moléculas estranhas, que são, em princípio, indistinguíveis. Se essa distinção não ocorre, então uma resposta imunológica é desencadeada contra as próprias células, causando as doenças auto-imunes.

Os resultados do uso de um sistema de detecção e neutralização de vírus computacionais que não incorporasse essa capacidade seriam, então, desastrosos: poderia induzir o computador a erros grosseiros, como a identificação errônea de softwares legítimos como ilegítimos e a eliminação equivocada desses arquivos.

2.2. Colônias de partículas

A otimização por colônia de partículas (OCP) é uma abordagem de algoritmos evolutivos baseada em uma população de indivíduos (soluções) e motivada pela simulação de comportamento social em vez da sobrevivência do indivíduo mais apto e da evolução da natureza como em outros AEs, tais como algoritmos genéticos.

Toma-se como o exemplo o comportamento das aves. Quando as aves estão em busca do alimento ou de um ninho (observe a figura abaixo), elas utilizam de

suas experiências com essas atividades e a experiência do bando. Essas experiências são utilizadas como mecanismos de otimização, para encontrar a solução para o problema. Falando de uma maneira mais técnica, a área sobrevoada pode ser identificada com o espaço de projeto e encontrar o local com a comida ou o ninho é semelhante a encontrar o ótimo da solução.

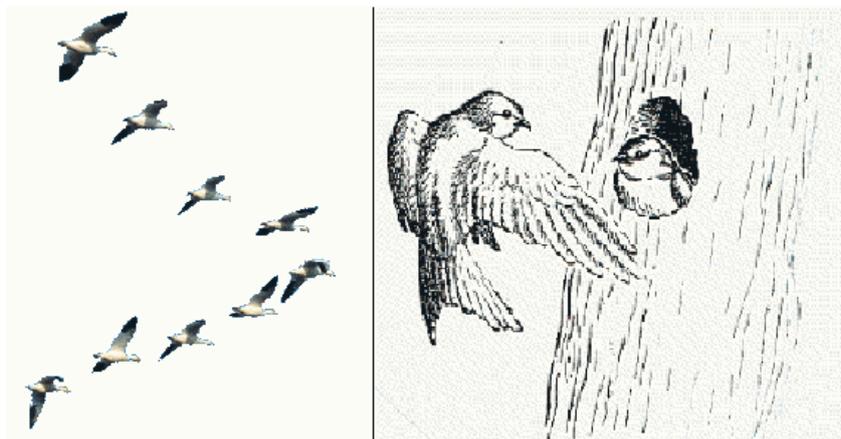


Figura 2.2: Representação do comportamento social das aves.

Este modelo de estudo, através das aves, é a técnica desenvolvida pelo biólogo Frank Heppner, que é baseado no seguinte comportamento: pássaros estão dispostos aleatoriamente por um território a procura de comida e de um local para construir seus ninhos, eles não sabem onde está esse lugar e este é único. A indagação é qual o melhor comportamento que os pássaros terão que realizar para conseguir efetuar o objetivo disposto. Inicialmente os pássaros voam sem nenhuma orientação prévia, eles se aglomeram em bandos, até que conseguem encontrar o ninho e os outros os seguem (ROCHA & SARAMAGO, 2005).

Na OCP, cada solução candidata (as aves, ou partículas) possui associada a ela um vetor de velocidade e outro de posição. A posição de cada partícula é atualizada de acordo com a velocidade atual, que por sua vez é ajustada através de uma equação de atualização que considera a experiência da partícula correspondente e a experiência das outras partículas da população. O fluxograma a seguir (Figura 2.3) representa um esboço do caminho realizado pelo algoritmo (ROCHA & SARAMAGO, 2005).

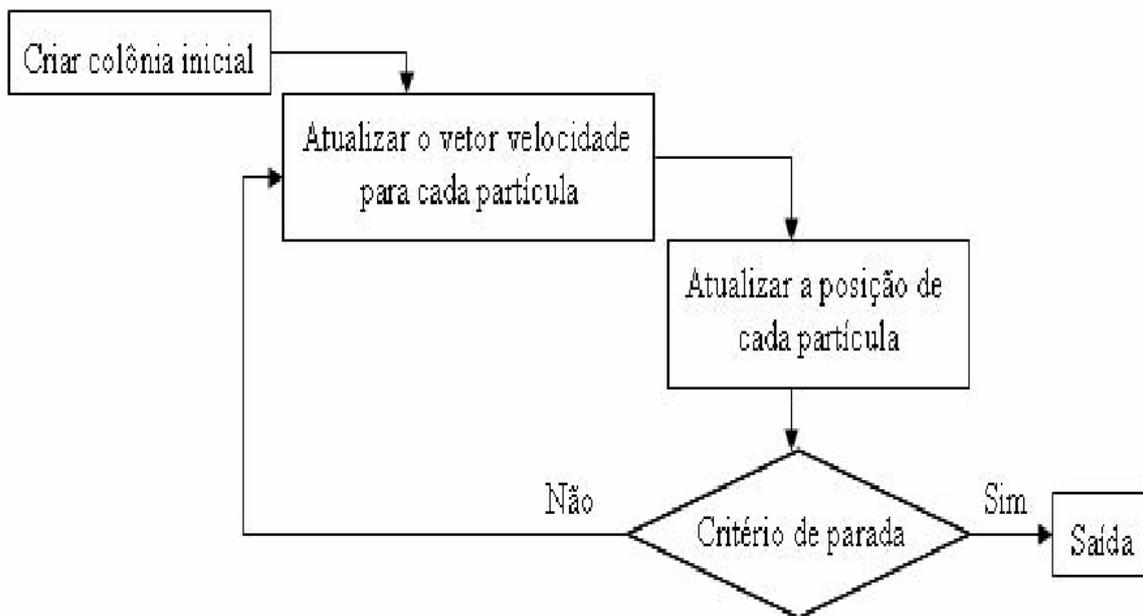


Figura 2.3: Fluxograma para o algoritmo OCP básico.

De forma similar a outros AEs, a OCP é iniciada com uma população (colônia inicial) de soluções gerada aleatoriamente. Já de forma diferente dos AEs, cada uma dessas soluções potenciais (indivíduos) é também atribuída uma velocidade aleatória. As soluções potenciais denominadas partículas são então “movimentadas” pelo espaço de busca do problema (COELHO, 2003).

Cada partícula conserva o conhecimento do seu melhor valor da função de aptidão denotada por *pbest* (versão local). Um outro melhor valor é “seguido” pela versão global, *gbest*, do otimizador por colônia de partícula e sua localização obtida de alguma partícula que compõe a população (COELHO, 2003).

O conceito de OCP consiste de, a cada passo iterativo, mudar a velocidade (acelerando) de cada partícula em direção as localizações do *pbest* e do *gbest*. A aceleração desta busca é ponderada através de um termo gerado de forma aleatória, vinculados estes de forma separadas as localizações do *pbest* e do *gbest* (COELHO, 2003).

2.3. Colônia de formigas

O algoritmo de colônia de formigas, ou *ant colony systems* (ACS) é inspirado pelo comportamento de colônia de formigas reais, em particular, pelo seu

comportamento de procura de alimento. Uma das idéias centrais do ACS é a comunicação indireta entre a colônia, baseada em trilhas de feromônios.

As trilhas de feromônios são um tipo de informação numérica distribuída que é modificada pelas formigas para refletir sua experiência quando da resolução de um problema em particular (COELHO, 2003).

O procedimento das formigas para encontrar o caminho mais curto entre o ninho e a fonte de alimento é auto catalítico, ou seja, um processo que se alimenta de seu próprio impulso para se desenvolver. Quando a formiga toma o melhor caminho baseado na quantidade de feromônio contido na trilha correspondente ela está ao mesmo tempo contribuindo para melhorar a precisão da informação, naquele ponto, que ajudará a próxima formiga a tomar o melhor caminho. Isso funciona de forma circular resultando na melhora constante da solução. O efeito auto catalítico combinado com a avaliação implícita da solução é o que produz o efeito desejado nos sistemas que utilizam o comportamento da formiga como inspiração (TELES, 2003).

Uma observação a ser feita é que a intensidade de feromônio funciona como uma meta heurística, ou seja, quanto maior a quantidade de feromônio naquela trilha melhor será a trilha que permite à formiga ter localmente informações sobre o que fazer depois (TELES, 2003).

Note também o caráter coletivo do modo como as formigas conseguem descobrir o melhor caminho. A descoberta da solução só é possível se toda a colônia agir conjuntamente. A atividade de uma única formiga isolada não possui inteligência suficiente para determinar o melhor caminho, mas a atividade de toda uma colônia possui essa inteligência (TELES, 2003).

Na versão artificial, as formigas são componentes de software e as taxas de feromônio são valores contínuos associados a arestas de um grafo que representa o ambiente para o problema em questão. As formigas cooperam entre si para resolver problemas difíceis de otimização como o problema do caixeiro viajante (TELES, 2003).

As formigas navegam por estados discretos de um determinado contexto. Através de transições entre estados vizinhos ela busca pela melhor solução do problema que geralmente é o menor número de transições possíveis para se chegar a um estado específico, partindo de outro. No problema do caixeiro viajante por

exemplo, os estados discretos são as cidades por onde o caixeiro passa e as transições são as ligações entre as cidades (TELES, 2003).

Nas figuras abaixo (2.4, 2.5 e 2.6) será mostrada uma simulação para que o leitor entenda melhor como as formigas se adaptam as alterações no ambiente e como o melhor caminho, mais curto, tende a ficar com mais feromônios.

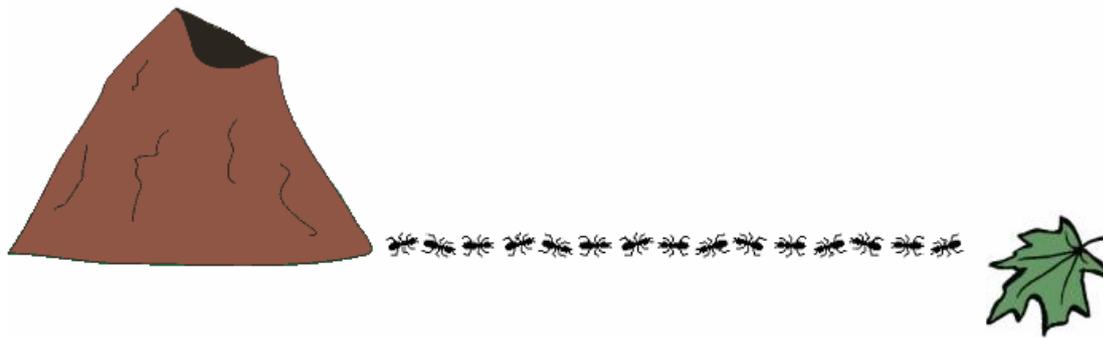


Figura 2.4: Formigas entre o formigueiro e o alimento.

Na situação acima já existe uma trilha formada indicando que é o melhor caminho entre o formigueiro e o alimento. Quando coloca-se um obstáculo no caminho (Figura 2.5) as formigas inicialmente se dividem na escolha do caminho para desviar o obstáculo. Mas depois de algum tempo um número maior de formigas começa a utilizar o menor caminho, onde está sendo depositada a maior quantidade de feromônio (Figura 2.6).

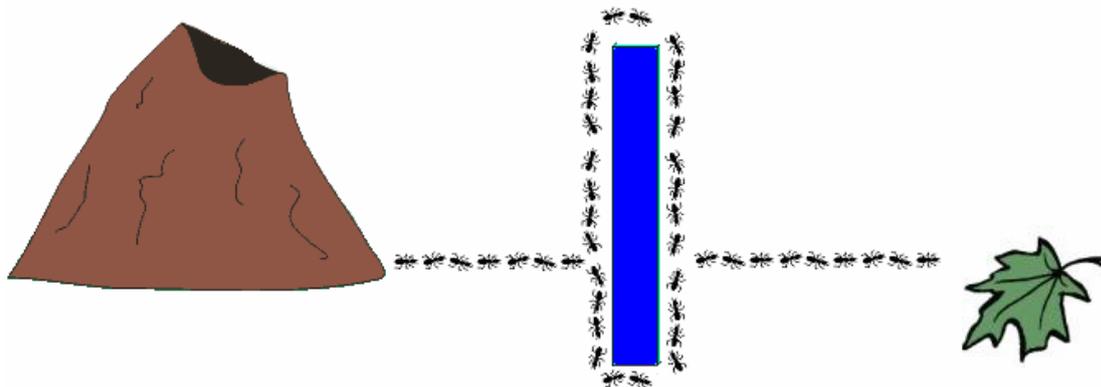


Figura 2.5: Obstáculo entre o formigueiro e o alimento.

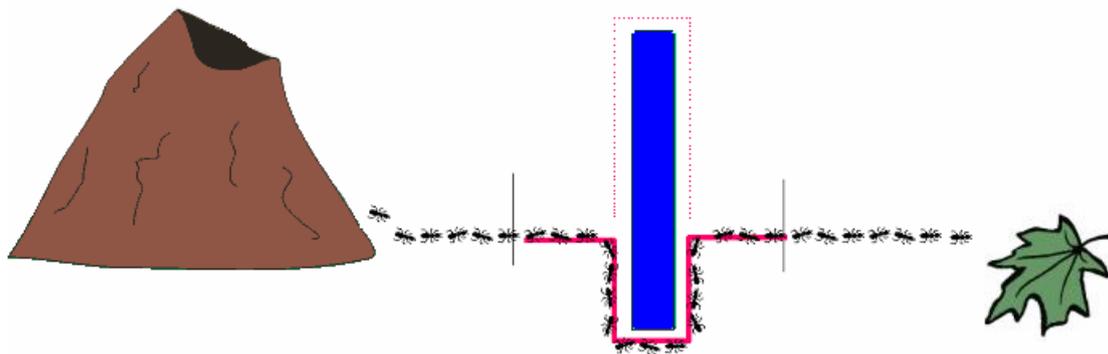


Figura 2.6: Melhor caminho escolhido e, portanto, com a maior quantidade de feromônio.

2.4. Algoritmos Genéticos

MICHALEWICZ, 1991:

A metáfora subjacente aos algoritmos genéticos é a seleção natural. Na evolução, o problema que cada espécie encara é a constante procura por melhores adaptações em um meio-ambiente complicado e sempre em mutação. O "conhecimento" que cada espécie adquire é incorporado na composição dos cromossomos de seus membros.

De acordo com o pensamento, os algoritmos genéticos são baseados na teoria da evolução de Darwin (Darwinismo). Desta forma, a maneira como cada membro de um grupo se comporta afetará diretamente o decorrer da evolução da espécie.

Os AGs trabalham de forma que considera um espaço de busca, onde é determinado um conjunto de soluções, ao qual denominado população. Esta população contém um conjunto de indivíduos, chamados em termo técnico de cromossomos, que são compostos por uma cadeia de genes (string ou cadeia de bits). Os genes estão localizados em posições no cromossomo chamadas de "loci" (posição da string). Cada gene contém a herança de uma ou várias características da geração anterior.

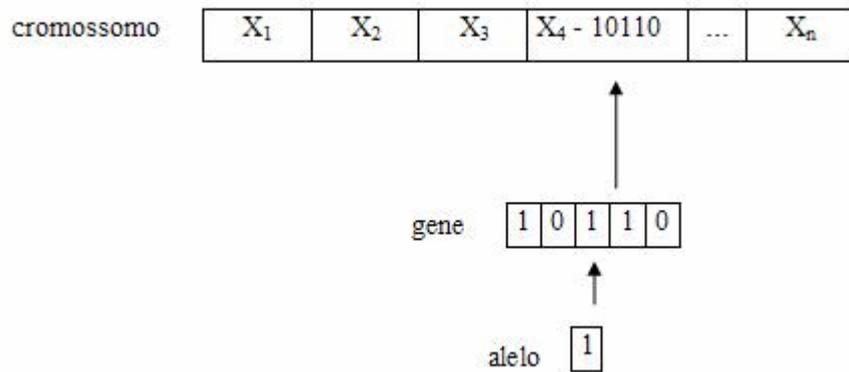


Figura 2.7: Esquematização de um cromossomo.

Os AGs atribuem a cada membro da população o valor de uma função objetivo a ser otimizada, chamada de *fitness* ou adequabilidade. A busca no espaço de soluções (i.e. onde estão todas as possíveis soluções para problema) se baseia no conjunto de indivíduos gerados e na avaliação do *fitness* de cada indivíduo. Este valor indica a qualidade do indivíduo na população, sendo uma medida de sua adaptabilidade ao ambiente. Indivíduos mais fortes, com melhor *fitness*, terão mais chance de sobreviver e passar para a próxima geração. Assim como sendo um algoritmo iterativo, a cada iteração a população é modificada herdando estas melhores características da geração anterior.

Quando se diz que o algoritmo genético é a metáfora da teoria de Darwin, está explicando-se que a idéia e os termos utilizados provêm da genética, seleção natural ocorrida no meio ambiente. Isto explica porque os AGs usam muitos termos da biologia. A relação abaixo descreve os principais termos encontrados na literatura:

- Cromossomo e Genoma: na biologia, genoma é o conjunto completo de genes de um organismo. Um genoma pode ter vários cromossomos. Nos AGs, os dois representam a estrutura de dados que codifica uma solução para um problema, ou seja, cromossomo ou genoma representa um simples ponto no espaço de busca.
- Gene: na biologia, é a unidade de hereditariedade que é transmitida pelo cromossomo e que controla as características do organismo. Nos AGs, é um parâmetro codificado no cromossomo, ou seja, um elemento do vetor que representa o cromossomo.

- Indivíduo: um simples membro da população. Nos AGs, um indivíduo é formado pelo cromossomo e sua aptidão.
- Genótipo: na biologia, representa a composição genética contida no genoma. Nos AGs, representa a informação contida no cromossomo ou genoma.
- Fenótipo: nos Algoritmos Genéticos, representa o objeto, estrutura ou organismo construído a partir das informações do genótipo. É o cromossomo decodificado. Por exemplo, considere que o cromossomo codifica parâmetros como as dimensões das vigas em um projeto de construção de um edifício, ou as conexões e peso de uma rede neural. O fenótipo seria o edifício construído ou a rede neural.
- Alelo: na biologia, representa uma das formas alternativas de um gene. Nos AGs, representa os valores ou características que o gene pode assumir. Por exemplo, um gene que representa o parâmetro cor de um objeto poderia ter o alelo azul, preto, etc.

Outros critérios também devem ser aprendidos para se entender melhor os procedimentos existentes no algoritmo:

- População inicial: pode ser gerada de várias maneiras. Se uma população inicial for gerada aleatoriamente, provavelmente, algumas regiões do espaço de busca não serão representadas. Este problema pode ser minimizado gerando a população inicial de maneira uniforme (i.e. com pontos igualmente espaçados). Outra alternativa é gerar a primeira metade da população aleatoriamente e a segunda metade a partir da primeira, invertendo os bits. Isto garante que cada posição da cadeia de bits tenha um representante na população.
- Função objetivo: a função objetivo consiste representação do *fitness* de um indivíduo, contendo o melhor valor de adequabilidade para um indivíduo.
- Critérios de parada: alguns dos vários critérios de parada dos AGs são: 1. quando o AG atingir um dado número de gerações; 2. chegada do valor ótimo da função objetivo, se este é conhecido; 3. convergência, isto é, quando não ocorrer melhoramento significativo no cromossomo de maior aptidão por um dado número de gerações.

Algumas aplicações dos AGs em áreas científicas, (MIRANDA, 2000):

- Síntese de circuitos analógicos: para uma certa entrada e uma saída desejada, por exemplo tensão, o AG gera a topologia , o tipo e o valor dos componentes do circuito;
- Síntese de protocolos: determinação de quais funções do protocolo devem ser implementadas em hardware e quais devem ser implementadas em software para que um certo desempenho seja alcançado;
- Programação Genética: gera a listagem de um programa, numa determinada linguagem especificada, para que um determinado conjunto de dados de entrada forneça uma saída desejada;
- Gerenciamento de redes: supervisão do tráfego nos links e das filas nos "buffers" de roteadores para descobrir rotas ótimas e para reconfigurar as rotas existentes no caso de falha de algum link;
- Computação Evolutiva: gera programas que se adaptam a mudanças no sistema ao longo do tempo.
- Otimização evolutiva multi-critério: otimização de funções com múltiplos objetivos que sejam conflitantes.
- Problemas de otimização complexos: problemas com muitas variáveis e espaços de soluções de dimensões elevadas. Ex: problema do caixeiro viajante;
- Ciências biológicas: modela processos biológicos para o entendimento do comportamento de estruturas genéticas.
- Autômatos auto-programáveis.

No próximo capítulo, os AGs serão estudados em detalhe para que mais tarde estes conceitos possam ser aplicados na implementação da otimização do tráfego viário, já apresentada como objeto de estudo deste trabalho.

3 - ALGORITMOS GENÉTICOS

Após uma leve introdução do conjunto de algoritmos evolutivos, nesta seção será discutida a funcionalidade dos AGs, de forma a detalhar a operação desse algoritmo.

3.1. Como os AGs funcionam

A parte principal do estudo é entender o funcionamento dos AGs. Como dito anteriormente os AGs são métodos de buscas estocásticas, ou seja, são algoritmos que trabalham em cima de um espaço de soluções possíveis para o determinado problema. Esta busca é baseada em um conjunto de soluções candidatas denominadas população.

Um Algoritmo Genético (como qualquer AE) precisa possuir os seguintes cinco componentes para um problema particular (MICHALEWICZ, 1991):

- Uma representação genética para as potenciais soluções para o problema;
- Um caminho para criar uma população inicial de soluções potenciais;
- Uma função objetivo que representa todo o ambiente, selecionando soluções através de seus “*fitness*”;
- Operadores genéticos que mudam a composição dos filhos;
- Valores para os vários parâmetros que os AGs utilizam (tamanho da população, probabilidades utilizadas nos operadores genéticos, etc).

Inicialmente, para a aplicação do algoritmo para um problema qualquer, deve-se encontrar alguma representação cromossômica conveniente, cuja população representará o espaço de busca do problema, com vetores binários de zero e um, que poderão ser geradas aleatoriamente ou por resultado de algum processo heurístico.

Após definido os cromossomos e gerado os genes aleatoriamente, formando a população inicial, o passo seguinte é submeter estes cromossomos aos processos chamados de operadores genéticos. Esses operadores são conhecidos como: Reprodução - alguns livros citam como seleção, exemplos: (COELHO, 2003) e

(MICHALEWICZ, 1991), Cruzamento, ou *crossover* e, finalmente, operador de Mutação. Cada um desses operadores tem a finalidade de criar uma nova geração de possíveis soluções para o problema.

No operador de reprodução as cadeias de genes passarão por equações que irão determinar o valor final de cada uma de suas funções objetivo, ou seja, o valor da adaptação de cada gene. A adaptação do gene irá definir sua próxima geração, se o gene possui uma alta adequabilidade então a probabilidade de permanecer na população também será, proporcionalmente, alta. Portanto, quanto maior o valor da função objetivo do gene, maiores são as chances do indivíduo sobreviver no ambiente e reproduzir, passando parte do seu material genético as gerações posteriores.

No operador de cruzamento algumas cadeias de genes irão sofrer uma troca de uma parte de sua composição. Serão escolhidos alguns pares de cadeias randomicamente, onde uma posição k será definida, também randomicamente, e a partir desta posição haverá a troca dos alelos entre as cadeias A e B escolhidas. A partir daí obtêm-se uma nova geração.

O operador de mutação é o próximo, após o cruzamento, a ser executado, onde alguns indivíduos são escolhidos a partir da nova geração irão sofrer uma mudança na sua composição. Essa mudança consiste na troca de um ou mais alelos, quando for 0 passará a ser 1, e quando for 1 passará a ser 0. Essa mudança ocorre aleatoriamente com uma determinada probabilidade e implicará na formação de uma nova geração. A mutação é utilizada para garantir uma varredura do espaço de estados e evitar que o algoritmo genético convirja muito cedo para mínimos ou máximos locais.

Esses processos ocorrerão até que se atenda ao critério de parada e assim escolhida a melhor população, com o melhor conjunto de soluções para o problema.

Resumindo, (LACERDA & CARVALHO, 1999):

O primeiro passo de um AG típico é a geração de uma “população inicial de cromossomos”, que é formada por um conjunto aleatório de cromossomos que representam possíveis soluções do problema a ser resolvido. Durante o “processo evolutivo”, esta população é avaliada e cada cromossomo recebe uma nota (denominada de “aptidão” no jargão da literatura de AGs), refletindo a qualidade da solução que ele representa. Em geral, os cromossomos mais aptos são selecionados e os menos aptos são descartados (Darwinismo). Os “membros” selecionados podem sofrer modificações em suas características fundamentais através dos “operadores” de *crossover* e “mutação”, gerando descendentes para a

próxima geração. Este processo é repetido até que uma solução satisfatória seja encontrada.

Considerando a codificação binária dos parâmetros, as etapas do AG podem ser descritas da seguinte forma:

1. considerar uma população inicial de N indivíduos gerada aleatoriamente;
2. calcular o *fitness* de cada indivíduo da população proposta;
3. repetir os passos seguintes até formar uma nova população:
 - 3.1. aplicar o operador de seleção: selecione um par de indivíduos da população de forma que cada indivíduo V^i da população é selecionado ou não, segundo uma probabilidade de seleção $P_s = (P f_i / f)$, onde f_i é o seu *fitness* – a escolha pode permitir que um indivíduo selecionado anteriormente pode ser escolhido novamente para ser pai;
 - 3.2. efetuar o operador de cruzamento: cruza-se as cadeias cromossômicas dos indivíduos, escolhendo-se o ponto de corte, o qual define as sub-cadeias a serem utilizadas, por uma probabilidade de *crossover* fixa P_c , simulando a reprodução sexuada, onde há troca de fragmentos de cromossomos (sub-cadeias de bits). O cruzamento resulta em dois indivíduos filhos, cada qual formado pela primeira sub-cadeia de um pai com a segunda sub-cadeia do outro;
 - 3.3. aplicar o operador de mutação aos novos indivíduos: a mutação ocorre segundo uma probabilidade P_m e consiste em selecionar uma posição na cadeia cromossômica e trocar seu valor binário. A mutação introduz material genético novo na população e permite a exploração de novas regiões no espaço de parâmetros;
4. substituir a população anterior pela nova população;
5. verificar se o critério de parada foi atendido, caso contrário retornar ao passo 2;
6. informar o indivíduo de maior *fitness* - fim.

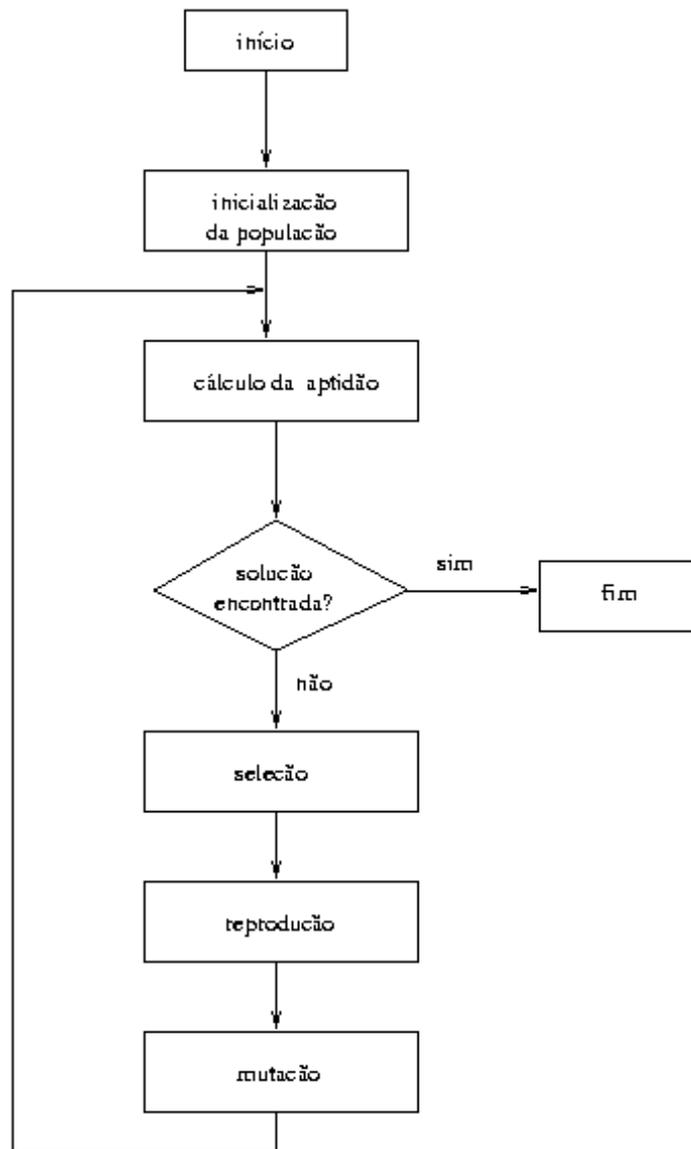


Figura 3.1: Estrutura básica de um AG simples (Miranda, 2000).

3.2. Operadores genéticos

De acordo com o que acontece na evolução dos seres, citado em Lacerda e Carvalho (LACERDA & CARVALHO, 1999):

Na biologia, a teoria da evolução diz que o meio ambiente seleciona, em cada geração, os seres vivos mais aptos de uma população para sobrevivência. Como resultado, somente os mais aptos conseguem se reproduzir, uma vez que os menos adaptados geralmente são eliminados antes de gerarem descendentes. Durante a reprodução, ocorrem fenômenos como mutação e crossover (recombinação), entre outros, que atuam sobre o material genético armazenado nos cromossomos. Estes fenômenos levam à variabilidade dos seres vivos na população. Sobre esta população diversificada age a seleção natural, permitindo a sobrevivência apenas dos seres mais adaptados.

Como citado anteriormente, os operadores genéticos são os responsáveis por todo o processo realizado no ciclo evolutivo dos AGs. É a partir destes que se obtém o resultado final do ciclo de otimização de forma eficiente e eficaz. O princípio básico desses operadores é transformar a população através de sucessivas gerações, estendendo a busca até chegar a um resultado satisfatório. Eles são necessários para que a população se diversifique e mantenha características de adaptação adquiridas pelas gerações anteriores.

Um algoritmo genético padrão evolui, em suas sucessivas gerações, mediante o uso de três operadores básicos. *Seleção*: realiza o processo de adaptabilidade e sobrevivência; *crossover* (cruzamento): representa o acasalamento entre os indivíduos; *mutação*: introduz modificações aleatórias (SILVA, 2003).

Observe o pseudocódigo a seguir, (COELHO, 2003):

```

t := 0;
iniciar P(t): P(0) := {x1(0), x2(0), ..., xα(0)}
avaliar P(0) := {Φ(x1(0)), Φ(x2(0)), ..., Φ(xα(0))}
enquanto uma condição de parada não é satisfeita {
    realizar crossover: P(t) := cec(P(t))
    realizar mutação: P'(t) := mem(P(t))
    avaliar P'(t) := {Φ(x'1(t)), Φ(x'2(t)), ..., Φ(x'α(t))}
    selecionar P(t+1) := ses(P'(t))
    t := t+1;
}

```

O pseudocódigo sintetiza as etapas mencionadas para os AGs, com as seguintes convenções:

x: indivíduo da população antiga;

x': indivíduo da população atual;

α: número de indivíduos da população;

P(t) := {x₁(t), x₂(t), ..., x_α(t)}: população na geração t;

P'(t) := {x'₁(t), x'₂(t), ..., x'_α(t)}: população atual na geração t;

Φ: I → R: mapeamento da função de aptidão;

m_{em}: operador de mutação com parâmetro de controle Θ_m;

c_{ec}: operador de cruzamento (crossover) com parâmetro de controle

Θ_c;

s_{es} : operador de seleção;

I : conjunto canônico, isto é, as variáveis com valores 0 ou 1.

O operador genético de cruzamento é o operador principal do algoritmo genético, e é responsável pela troca de material genético entre os indivíduos com maior probabilidade de reproduzirem, os indivíduos mais aptos ao ambiente.

O(s) ponto(s) de cruzamento é escolhido aleatoriamente. Após a escolha randômica, os pontos de dois cromossomos ancestrais são trocados, assim ocorrendo a formação dos cromossomos descendentes (COELHO, 2003). O cruzamento é responsável pela recombinação de características dos pais durante a reprodução, permitindo que as próximas gerações herdem essas características.

Pode, ainda, ser utilizado de várias maneiras. As mais utilizadas são:

- **um-ponto:** um ponto é escolhido e a partir desse ponto as informações genéticas são trocadas pelos pais.

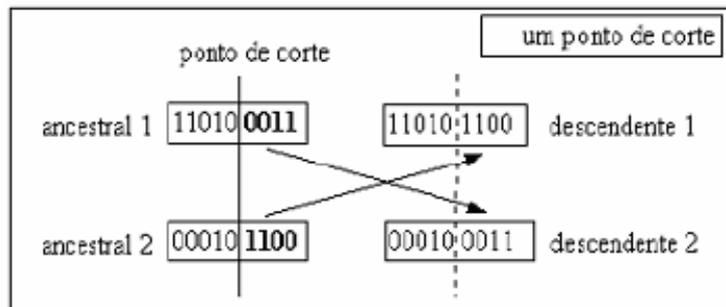


Figura 3.2: Operador de cruzamento de um ponto de corte (COELHO, 2003).

- **multi-pontos:** generalização do corte por um-ponto, porém são escolhidos diversos pontos de cruzamento.

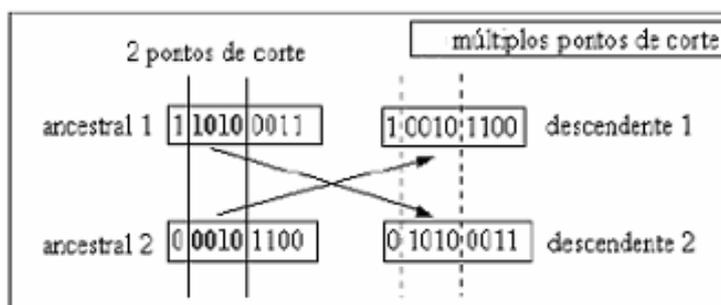


Figura 3.3: Operador de cruzamento de dois pontos de corte (COELHO, 2003).

- **uniforme:** não utiliza pontos de cruzamento, mas determina, através de um parâmetro global, qual a probabilidade de cada variável ser trocada entre os pais.

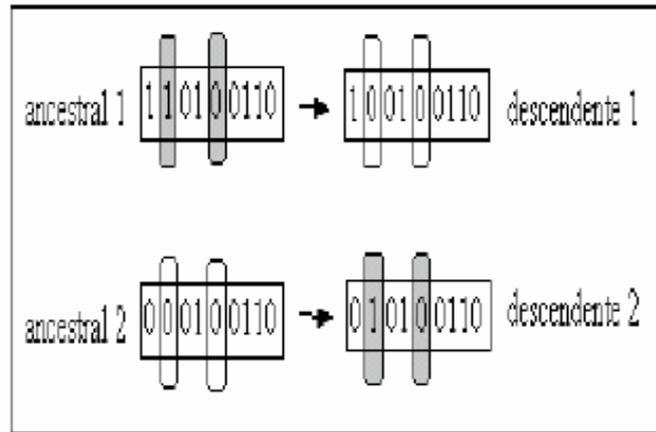


Figura 3.4: Operador de cruzamento uniforme (COELHO, 2003).

O operador de mutação é necessário para a introdução e manutenção da diversidade genética da população, alterando arbitrariamente um ou mais componentes de uma estrutura escolhida, fornecendo assim, meios para introdução de novos elementos na população. Este é introduzido para prevenir a convergência prematura para ótimos locais, através da amostragem de novos pontos no espaço de busca. O operador de mutação é aplicado, a cada descendente individualmente, após a aplicação do operador de cruzamento. Consiste na mudança aleatória de uma parte da string (Figura 3.5), que representa o indivíduo, apresentando pequena probabilidade.

A mutação poderá ocorrer ou não, isso dependerá de uma probabilidade dada pela taxa de mutação P_m . O valor da taxa de mutação usualmente é pequeno (P_m e $[0..1]$), sendo alguns dos valores mais comumente usados $P_m=0.001$ e $P_m=0.01$ (SILVA, 2003).

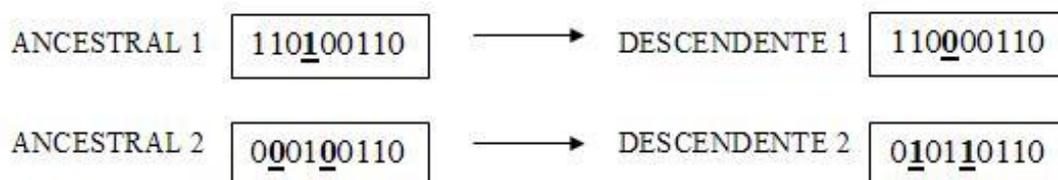


Figura 3.5: Operador de mutação.

Já o operador de seleção, ou também chamado de reprodução, emprega o princípio de sobrevivência dos indivíduos mais aptos, através de uma metáfora aos procedimentos de reprodução assexuada e seleção natural, de acordo com o grau de adaptação do indivíduo ao ambiente. O objetivo básico deste operador é enfatizar as melhores soluções que constituem uma população. O operador não cria nenhuma nova solução, este apenas seleciona as soluções relativamente aptas de uma população e remove as soluções remanescentes (COELHO, 2003).

A verificação se uma solução é apta (ou não) em uma população é baseada no valor da função de aptidão (*fitness*) da referida solução. Para que uma solução tenha maior aptidão deve ter uma alta probabilidade de seleção.

A reprodução caracteriza-se por um parâmetro denominado pressão seletiva, relacionado ao tempo de “preenchimento” do operador. O tempo de “preenchimento” é definido pela velocidade com que a melhor solução, na população inicial, pode ocupar toda a população através da aplicação repetida do operador de seleção (COELHO, 2003).

Quando o operador de reprodução apresenta uma grande pressão seletiva faz com que a população perca a diversidade rapidamente, ocasionando uma convergência prematura, para uma solução inadequada. Contudo, um operador de seleção com pressão seletiva pequena apresenta uma baixa convergência e permite aos operadores de crossover e mutação iterações suficientes à busca no espaço de soluções (COELHO, 2003).

O *crossover* é o principal mecanismo de busca do AG. Ele é capaz de combinar as boas porções dos cromossomos pais, isto é, os bons blocos de construção. Como resultado, cromossomos filhos com aptidões mais elevadas que as dos pais podem ser produzidos. A mutação também desempenha importante papel, uma vez que permite que qualquer ponto do espaço de busca seja alcançado. A fase de seleção descarta os cromossomos com baixa aptidão e com eles muitos genes também são eliminados. Na ausência de mutação, genes presentes apenas nos cromossomos descartados não serão mais recuperados, pois o *crossover* não gera novos genes, apenas combina os que já existem.

O desaparecimento de determinados genes da população no decorrer das gerações impossibilita um AG de explorar completamente o espaço de busca. Como

resultado, o AG pode convergir para um mínimo ou máximo local. Este problema é conhecido como *Convergência Prematura*. Portanto, com taxas adequadas de mutação, é possível manter uma boa diversidade de genes da população a fim de contrabalançar estas perdas de genes (LACERDA & CARVALHO, 1999).

3.3. Comparando AG com outros procedimentos de otimização

Os AGs diferem da maioria dos tipos normais dos procedimentos de busca e otimização de quatro maneiras (GOLDBERG, 1989):

1. Os AGs trabalham com uma codificação do conjunto de parâmetros, não estes por si só;
2. Os AGs procuram por uma população de pontos e não somente por um ponto;
3. Os AGs usam informações da função objetivo, não dos derivados ou dos outros conhecimentos auxiliares.
4. Os AGs usam probabilidades de transição de regras e não regras determinadas.

Utilizadas juntas, essas quatro diferenças – uso direto de uma codificação, busca de uma população, informações auxiliares escondidas e operadores aleatórios – contribuem para a robustez dos algoritmos genéticos e resultam na vantagem sobre outras técnicas usadas mais comumente (GOLDBERG, 1989).

Em muitos métodos de otimização, a solução pode passar de um ponto simples no espaço de decisão para o próximo usando alguma regra de transição para determinar o próximo ponto. Esse método de ponto-a-ponto é perigoso porque é a perfeita prescrição para localizar falsos picos em espaços de busca multimodais (muitos picos). Em contraste, os AGs trabalham com uma rica base de dados de pontos simultaneamente (uma população de strings), escalando muitos picos em paralelo; portanto, a probabilidade de achar um pico falso é reduzida sobre os métodos que vão ponto-a-ponto (GOLDBERG, 1989).

Diferente de muitos métodos, os AGs usam regras de transições probabilísticas para guiar a sua busca. Para pessoas familiarizadas com os métodos determinísticos isso parece esquisito. Mas o uso de probabilidade não sugere que o método é alguma simples busca aleatória; isso não é uma decisão tomada por mera

sorte. Os Algoritmos Genéticos usam a escolha aleatória como uma ferramenta para guiar a pesquisa em direção a regiões do espaço de busca com uma melhora considerável (GOLDBERG, 1989).

Pela simulação de processos evolucionários naturais, um AG pode eficientemente realizar buscas de soluções no domínio do problema através de uma população base, ao invés de uma simples solução, e empregar heurísticas para desenvolver as melhores soluções. A facilidade de reiniciar uma busca iterativa de uma ampla variedade de pontos de partida fornece uma segurança. Dessa forma fazem dos Algoritmos Genéticos superiores aos métodos de busca convencionais (Ma e Abdulhai, 2001).

3.4. Otimização por Algoritmos Genéticos

A otimização é a busca da melhor solução para um dado problema. Consiste em tentar várias soluções e utilizar a informação obtida neste processo de forma a encontrar soluções cada vez melhores.

Um exemplo para ilustrar esse processo poderia ser a tentativa de melhorar a imagem de uma TV através de uma antena. Em cada ajuste dado na antena existe uma solução que está sendo testada, e quando escolhida a melhor posição obteve-se uma resposta ótima, uma boa imagem na TV (LACERDA & CARVALHO, 1999).

Descobrir a otimização de uma função não pode ser considerado um procedimento simples que avalia somente a comparação com os melhores critérios, perfeição. É uma avaliação da função como um todo e os ótimos que ela pode produzir. A realização da otimização é muito importante para os sistemas complexos.

Quando estamos tentando descobrir o ótimo de uma função, o que estamos tentando realizar? A visão convencional para isso é apresentada em Beightler, Phillips e Wilde (BEIGHTLER, 1979):

A aspiração do homem por perfeição encontra expressão na teoria da otimização. Ela estuda como descrever e realizar o que é Melhor, uma vez que se sabe como medir e alterar o que é Bom ou Ruim... *Teoria da otimização* abrange o estudo quantitativo do ótimo e seus métodos de procura.

Assim, a otimização procura melhorar a performance para algum ponto ou vários pontos ótimos (GOLDBERG, 1989).

A otimização de modelos envolve a seleção dos melhores conjuntos de valores para os parâmetros. Melhores valores podem ser obtidos usando algoritmo genético, alcançando uma otimização combinatorial para o objetivo do sistema através de funções de minimização dos piores valores. O processo de busca a um ponto ótimo envolve alguns métodos de busca sistemáticos para alcançar um ótimo global ou o melhor local ótimo alcançável.

Algumas vantagens de se utilizar os AGs como solução de um problema de otimização (LACERDA & CARVALHO, 1999):

- Funcionam tanto com parâmetros contínuos como discretos ou uma combinação deles;
- Realizam buscas simultâneas em várias regiões do espaço de busca, pois trabalham com uma população e não com um único ponto;
- Utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar;
- Não é necessário conhecimento matemático aprofundado do problema considerado;
- Otimizam um número grande de variáveis;
- Otimizam parâmetros de funções objetivos com superfícies complexas e complicadas, reduzindo a incidência de mínimos locais;
- Adaptam-se bem a computadores paralelos;
- Trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros;
- Fornecem uma lista de parâmetros ótimos e não uma simples solução;
- Trabalham com dados gerados experimentalmente e são tolerantes a ruídos e dados incompletos;
- São fáceis de serem implementados e portáteis, no sentido que o mecanismo da evolução é separado da representação particular do problema considerado. Assim, eles podem ser transferidos de um problema para outro;
- São flexíveis para trabalhar com restrições arbitrárias e otimizar múltiplas funções com objetivos conflitantes;
- E, são também facilmente hibridizados com outras técnicas e heurísticas.

Apesar de todas essas vantagens, os AGs não são eficientes para muitos problemas. São bastante lentos e não raro ainda estão avaliando a população inicial enquanto muitos métodos já encontraram a solução. O principal campo de aplicação do AG é em problemas complexos, com múltiplos mínimos/máximos e para os quais não existe um algoritmo de otimização tão eficiente e eficaz conhecido para resolvê-los.

4 – SIMULAÇÃO DE OTIMIZAÇÃO DE TRÁFEGO

4.1 Estudo de caso

Após o estudo e fundamentação do Algoritmo Genético, esta seção tratará o desenvolvimento do estudo de caso para solução proposta de otimização de tráfego viário.

4.1.1 O problema

A aceleração da urbanização a partir da década de 30 se devia, por um lado, à industrialização, promovida, em parte, por políticas de substituição de importação e, por outro lado, pelo êxodo rural, 50% de que ultimamente se radicou em grandes cidades atraídas pela oportunidade de emprego e pelo conforto e serviços urbanos como a educação, saúde e lazer proporcionados por elas.

Em decorrência dessa urbanização, algumas cidades brasileiras se encontram em estado de saturação. Essas cidades, que antes proporcionavam grandes ofertas de emprego e moradia, hoje passam por problemas como miséria, poluição, violência e o descontrole do tráfego urbano.

Com o crescimento populacional a quantidade de automóveis em circulação nas vias aumentou drasticamente, tornando o tráfego de cidades como São Paulo, caóticos. Hoje, São Paulo tenta contornar os congestionamentos com a prática do rodízio, mas mesmo assim, em algumas vias principais, os congestionamentos ainda ocorrem, sem contar que algumas pessoas não cumprem as regras do rodízio, utilizando seu automóvel sem autorização.

São Paulo, a maior cidade brasileira, tem 25% da frota nacional, o que hoje representa perto de cinco milhões de veículos. Praticamente há um carro para cada dois habitantes. A pesquisa Origem-Destino, realizada a cada dez anos desde 1967, abrangendo a área mais fortemente urbanizada da Região Metropolitana de São Paulo - que registra perto de seis milhões de veículos -, identificou em sua última versão 30 milhões de deslocamentos diários, sendo 10 milhões em transporte coletivo, 10 milhões em transporte individual e os restantes 10 milhões a pé. (SCARINGELLA, 2001):

Num período de cinco anos (entre 1992 e 1997), a média de quilômetros de congestionamento medidos pela CET no sistema viário principal da cidade passou de 40 km, na hora de pico da tarde, para 120 km. Hoje há congestionamentos significativos em corredores da mais longínqua periferia e em todos os quadrantes. O grau de "viscosidade" urbana aumenta, e a crise de mobilidade se agrava. Os deslocamentos ficam mais lentos, e as áreas congestionadas crescem.

Segundo (FERREIRA, 2004), a média de automóveis por pessoa disparou a partir da década de 70, onde ocorreu o auge do "milagre econômico", chegando a 3,47 habitantes por automóvel em 1996 no Distrito Federal.

Em 1950 havia um automóvel para cada 259,5 pessoas no Brasil, distribuídos com certa disparidade (91,6 habitantes para cada automóvel no Rio de Janeiro, 129,5 em São Paulo, 162,5 no Rio Grande do Sul e 28.688,8 para o Acre). Em 1970, no auge do período conhecido como "milagre econômico" a média de pessoas por automóvel passa para 37,6 pessoas por automóvel. Neste mesmo período, no Distrito Federal a média estava em 16,9 habitantes por automóvel, 18,8 em São Paulo, 22,2 no Rio de Janeiro. Já em 1996, a densidade para o Brasil passa a 8,12 habitantes por automóvel, com destaque para o Distrito Federal que possui a maior média de veículos do País, com 3,47 habitantes por automóvel.

Com isso a necessidade de se criar soluções para uma melhoria de trafegabilidade cresce em decorrência do aumento da quantidade de automóveis nas vias. Dentre as soluções possíveis se encontra a solução emergente proposta no escopo do trabalho, utilizando os Algoritmos Genéticos para definição do melhor caminho a seguir em um determinado momento.

4.1.2 Fundamentação teórica

Rotas, vias, fluxos de tráfego e caminhos são elementos essenciais para definição da solução do problema de trafegabilidade. Para representação desses caminhos, o estudo de caso baseia-se na utilização da estrutura de grafos, por se tratar de um paradigma de representação de conhecimento que possibilita modelar diversos problemas.

O grafo é a forma de apresentação mais utilizada quando se quer representar um conjunto de pontos e os caminhos que levam um ponto ao outro. A estrutura de um mapa tem como objetivo essas características, portanto o grafo representará um mapa com valores pré-definidos e toda a pesquisa será baseada neste.

O grafo da pesquisa servirá de apoio para o desenvolvimento do simulador, pois conterá as informações necessárias, além de ilustrar perfeitamente a solução do problema proposto. Para definição do problema de previsão e otimização de tráfego

viário o grafo representará um mapa e possibilitará a visualização de locais e trechos definidos.

Seja o Grafo (E, A) , onde E é o conjunto de vértices ou locais e A o conjunto de arestas ou trechos. Conforme Figura 4.1, tem-se que:

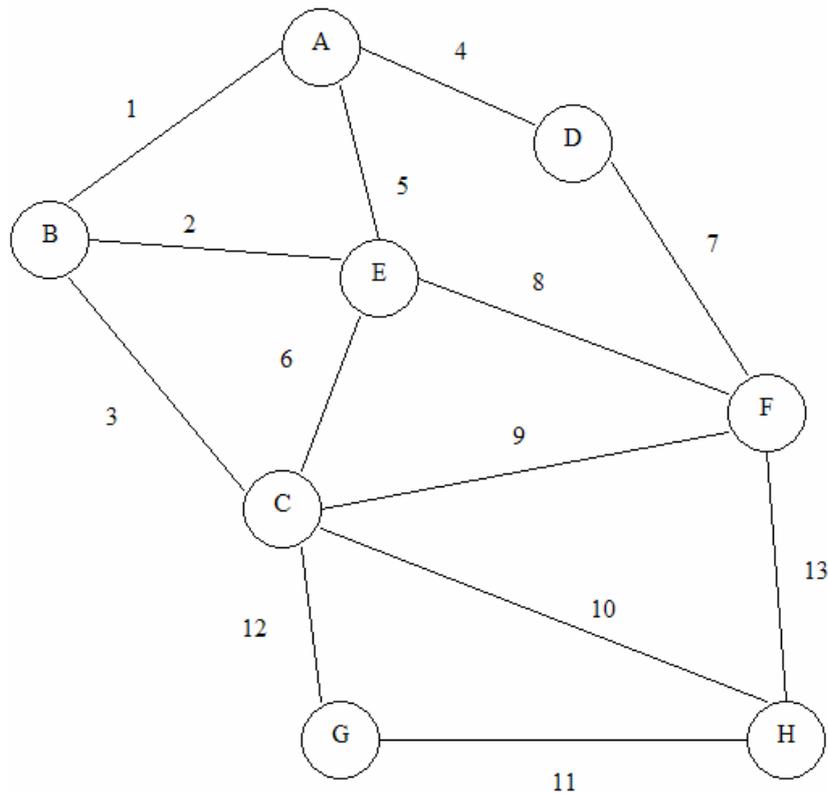


Figura 4.1: Grafo ilustrativo para um conjunto de locais e trechos.

O grafo possui como conjunto de vértices $E = \{A, B, C, D, E, F, G, H\}$ e conjunto de arestas $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$. Cada vértice representa um local que fará parte da escolha da origem e do destino (O-D). As arestas representam os trechos a se percorrer saindo de um local ao outro. Cada trecho possui informações, tais como, velocidade média, distância e tempo de percurso, necessárias para a escolha do melhor do caminho. O caminho é o conjunto de trechos possíveis para se utilizar em uma viagem de uma origem X a um destino Y.

Na solução do problema proposto o usuário informa os locais de origem e destino desejados. Após esta informação o software buscará as informações, em um banco de dados, que dizem respeito aos locais escolhidos e as combinações de trechos possíveis. A partir da seleção desses possíveis caminhos o Algoritmo

Genético será aplicado para definição do melhor caminho, utilizando as técnicas de reprodução, cruzamento e mutação, já estudados no tópico anterior.

A Tabela 4.1 determina o tempo (T_p) que se leva para percorrer cada trecho (T_c) definido no conjunto de arestas A.

Tabela 4.1: Tabela Trecho/Tempo

Trecho (T_c)	Tempo (T_p)
T_c 1	2,57 minutos
T_c 2	2,25 minutos
T_c 3	3,42 minutos
T_c 4	2,14 minutos
T_c 5	2,3 minutos
T_c 6	2,5 minutos
T_c 7	2,57 minutos
T_c 8	3 minutos
T_c 9	3,37 minutos
T_c 10	3,75 minutos
T_c 11	2,85 minutos
T_c 12	2,7 minutos
T_c 13	2,82 minutos

A fim de ilustrar o problema proposto de previsão e otimização de tráfego serão escolhidos a origem B e o destino H. A partir da escolha da O-D (origem - destino), as seguintes informações na Tabela 4.2 são encontradas:

Tabela 4.2: Conjunto de trechos possíveis da origem (B) ao destino (H) e o tempo total (TT) gasto para cada um dos caminhos.

Possibilidades P	Conjunto de trechos (Rotas)	Tempo total (TT)
P_1	(1,4,7,13)	10,1 min
P_2	(1,5,8,13)	10,69 min
P_3	(1,5,6,9,13)	13,56 min
P_4	(1,5,6,10)	11,12 min
P_5	(1,5,6,12,11)	12,92 min
P_6	(2,8,13)	8,07 min
P_7	(2,6,10)	8,5 min

P ₈	(2,6,9,13)	10,94 min
P ₉	(2,6,12,11)	10,3 min
P ₁₀	(3,6,8,13)	11,74 min
P ₁₁	(3,9,13)	9,61 min
P ₁₂	(3,10)	7,17 min
P ₁₃	(3,12,11)	8,97 min

Definido os caminhos possíveis para a origem e destino B-H, a estrutura de árvore da figura 4.2 é proposta como ilustração desses caminhos.

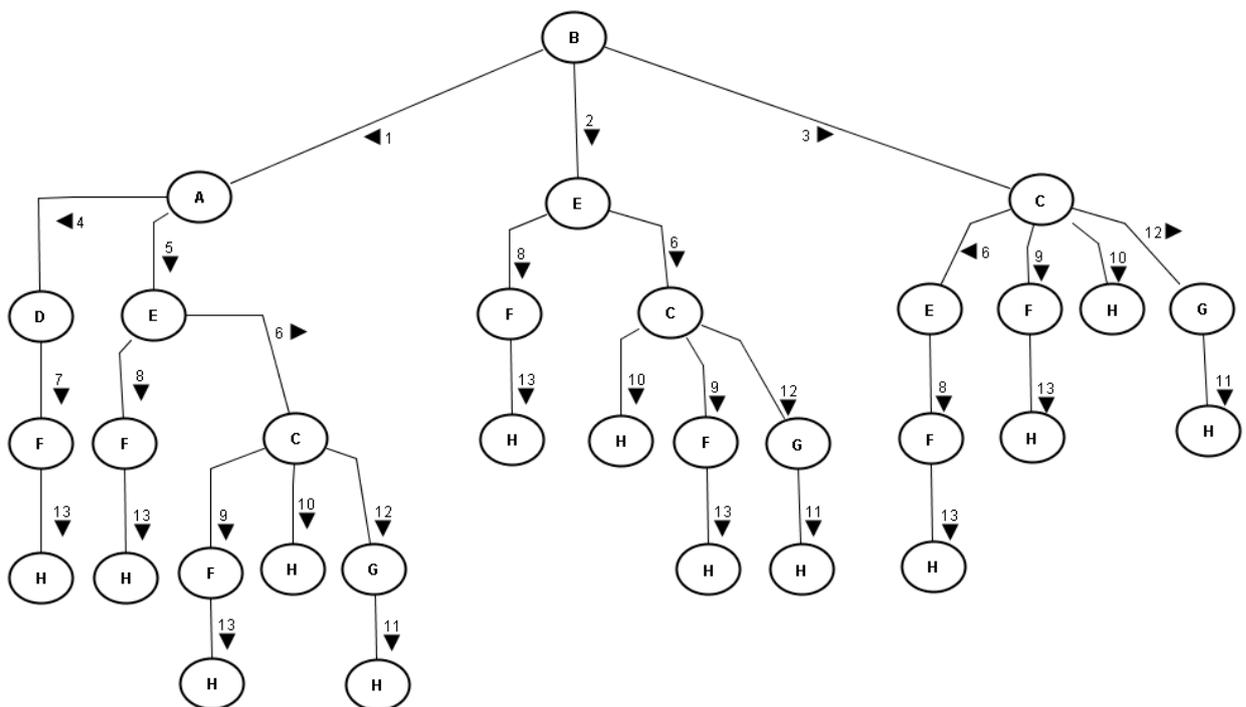


Figura 4.2: Árvore com caminhos possíveis para B-H.

Através da definição de cada rota possível a se tomar dada a origem e o destino B-H, o AG definirá a melhor dentre as encontradas. Em tese, a melhor rota é a P₁₂, onde o tempo total de percurso é o menor, 7.17 minutos, porém esse tempo está definido para um fluxo de trânsito livre.

Conforme a tabela 4.3 mostra, cada trecho possui uma velocidade média (V_n), onde seria a velocidade normal que os carros se encontrariam se o trânsito estiver em boas condições. Porém os trechos também possuem uma velocidade, chamada de velocidade crítica (V_c), que caracteriza se o trecho se encontra em congestionamento. A velocidade crítica é a velocidade normal da via sobre dois

($V_n/2$), o que torna o tempo de viagem maior e, portanto, esse trecho não pode ser considerado na escolha do melhor caminho.

A tabela 4.3 ilustra as informações contidas em cada trecho do grafo da figura 4.1 e suas respectivas velocidades críticas.

Tabela 4.3: Dados pertencentes a cada trecho (T_c).

Trecho (T_c)	Extensão (metros)	Gene	V_n (km/h)	V_c (km/h) $V_n/2$	Tempo (T_p) (minutos)
T_c 1	3000	0001	70	35	2,57 minutos
T_c 2	3000	0010	80	40	2,25 minutos
T_c 3	4000	0011	70	35	3,42 minutos
T_c 4	2500	0100	70	35	2,14 minutos
T_c 5	2300	0101	60	30	2,3 minutos
T_c 6	2500	0110	60	30	2,5 minutos
T_c 7	3000	0111	70	35	2,57 minutos
T_c 8	4000	1000	80	40	3 minutos
T_c 9	4500	1001	80	40	3,37 minutos
T_c 10	5000	1010	80	40	3,75 minutos
T_c 11	3800	1011	80	40	2,85 minutos
T_c 12	2700	1100	60	30	2,7 minutos
T_c 13	3300	1101	70	35	2,82 minutos

A velocidade crítica torna a solução mais real, principalmente pelo fato de que os trechos que possuirão a V_c naquele momento são escolhidos randomicamente. Dessa forma, qualquer um dos caminhos propostos na tabela 4.2 pode, em algum momento, se tornar a solução ótima.

A solução para o problema de otimização de fluxo de tráfego será, portanto, a obtenção do tempo mínimo de percurso no referido trecho. O Algoritmo Genético gerará informações classificadas por tempo mínimo de viagem em cada rota para ser utilizada como sugestões direcionadas ao usuário do sistema afim de buscar o equilíbrio da rede viária bem como sua eficiência, maximizando a sua utilização (SILVA, 2003).

4.1.3 Modelagem da solução

Para o desenvolvimento do sistema de simulação de otimização de tráfego viário, implementado através do estudo dos Algoritmos Genéticos, foram utilizadas técnicas de orientação a objetos (OO) para modelagem do sistema.

A UML é uma linguagem-padrão para a elaboração da estrutura de projetos de softwares. Uma linguagem de modelagem, como a UML, é uma linguagem cujo vocabulário e regras têm seu foco voltado para a representação conceitual e física de um sistema. A modelagem permite a compreensão do sistema como um todo ou em partes.

Por que fazer a modelagem? Porque a construção desses modelos servirá para melhor compreensão do sistema que está sendo desenvolvido, alcançando quatro objetivos (BOOCH, 2005):

1. Os modelos ajudam a visualizar o sistema como ele é ou como se deseja que seja;
2. Os modelos permitem especificar a estrutura ou o comportamento de um sistema;
3. Os modelos proporcionam um guia para a construção do sistema;
4. E, os modelos documentam as decisões tomadas.

Dessa forma, compreende-se a necessidade da modelagem de um sistema. O principal objetivo para utilização da técnica trata-se da contribuição desta para a apresentação do comportamento e estrutura do sistema.

4.1.3.1 Especificação

Inicialmente, a primeira modelagem produzida para o software simulador deste trabalho foi o Diagrama de classes (Figura 4.3). (MARTINS, 2004):

Os diagramas de classes são utilizados para representar a estrutura estática do sistema, composta pelas classes de negócio, classes de interface ..., e as classes de controle, responsáveis pelo controle das transações.

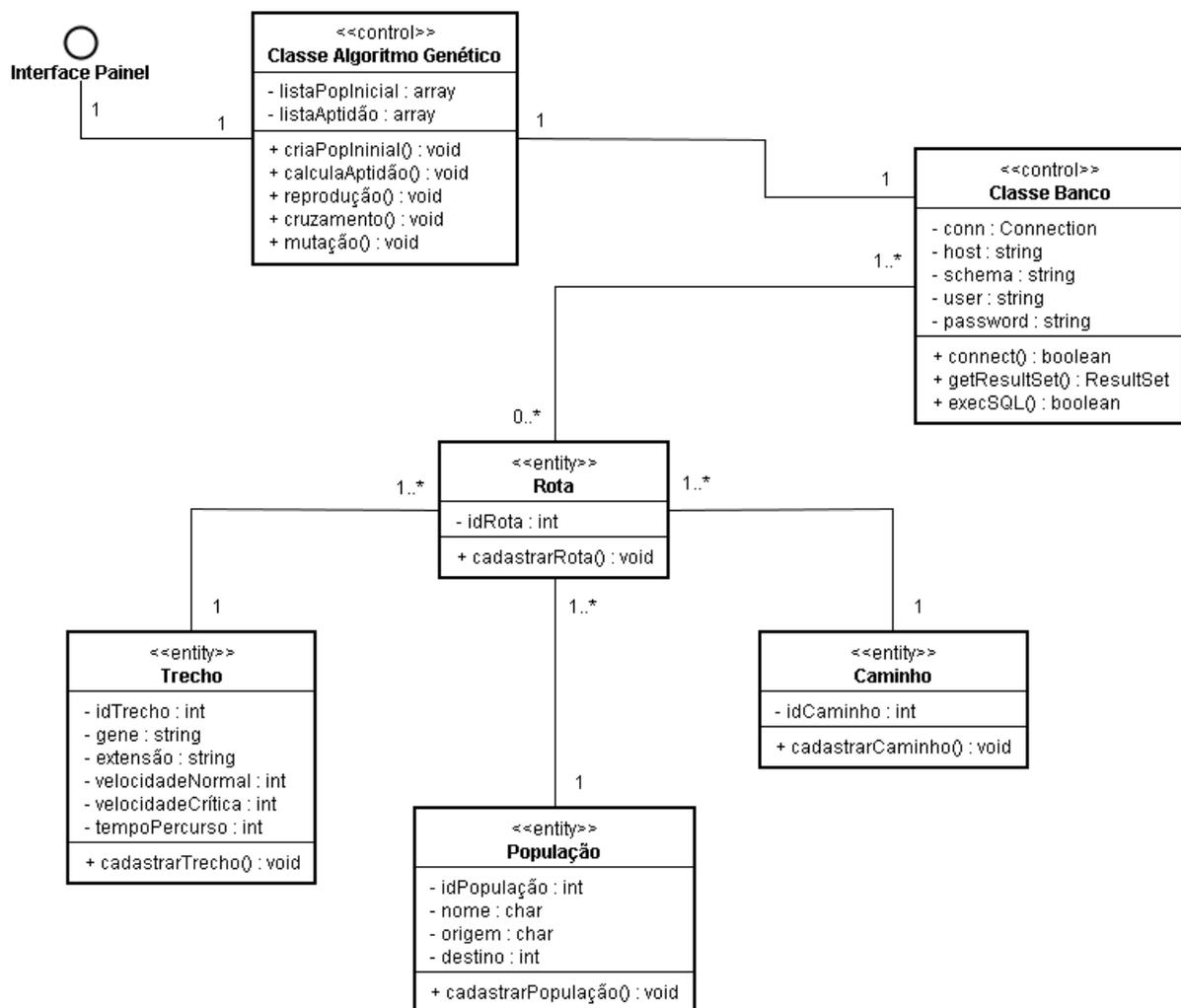


Figura 4.3: Diagrama de Classe de Domínio para o Sistema Simulador de Otimização de Tráfego Viário.

Estão representadas neste Diagrama de classes, as entidades, as classes de controle e a interface, existentes no sistema.

As classes de entidade representam os objetos ou coisas que se deseja armazenar informações no banco de dados. Trecho, por exemplo, é o objeto que representa as vias (ruas) e possui como um dado a distância em metros que é chamada de extensão.

As classes de controle, como o próprio nome sugere, controlam essas entidades. A Classe Banco, por exemplo, é a classe que fará todas as transações com o banco de dados. Dentre essas transações se encontra conexão, consulta, cadastro, alteração, etc.

A classe de interface desse diagrama de classe apenas representa a forma como se inicia a utilização do sistema.

Para modelagem das funcionalidades do sistema, foi produzido o Diagrama de Casos de Uso (Figura 4.4).

Segundo, (BOOCH, 2005):

Um diagrama de caso de uso mostra um conjunto de casos de uso e atores e seus relacionamentos. [...] são importantes principalmente para a organização e modelagem dos comportamentos de um sistema.

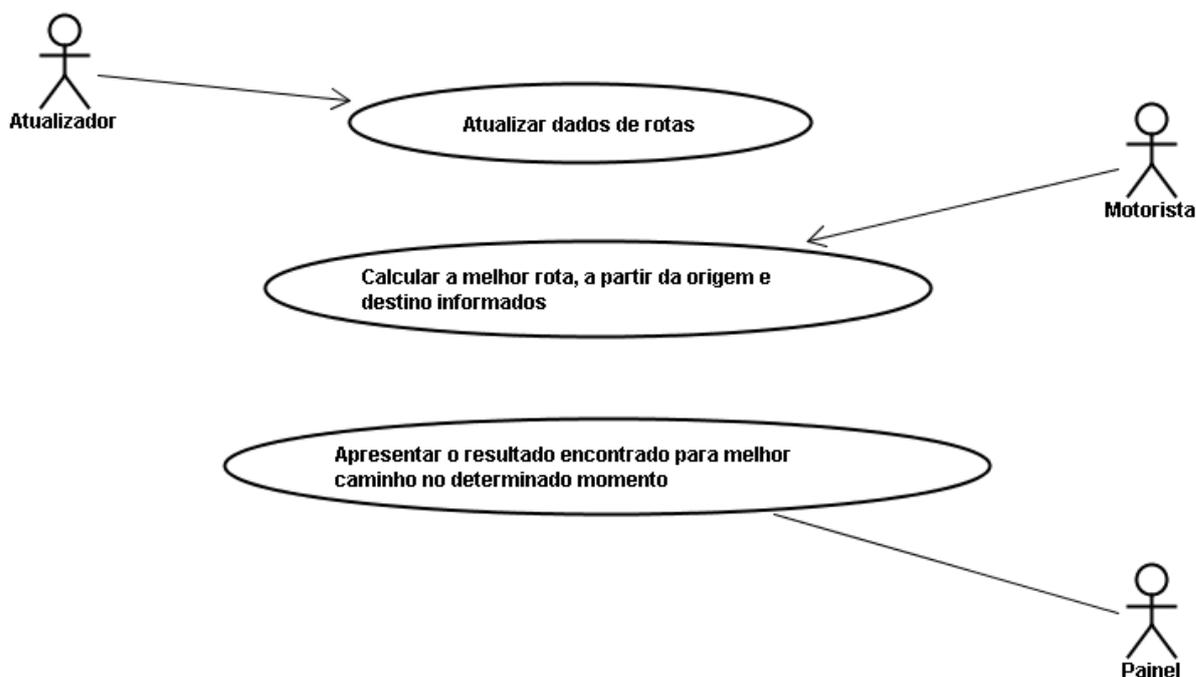


Figura 4.4: Diagrama de caso de uso para o Sistema Simulador de Otimização de Tráfego Viário.

O sistema possui três funcionalidades. A primeira está relacionada aos dados armazenados no banco de dados, onde um ator, nomeado Atualizador, atualiza as informações referentes as rotas, trechos, etc. A segunda funcionalidade, e mais importante, é onde o sistema recebe a origem e o destino informados pelo motorista e processa o melhor caminho a percorrer naquele momento. E a terceira funcionalidade apenas informa no painel o melhor caminho encontrado pelo sistema.

Para representação dos fluxos do sistema, foi construído o Diagrama de Atividades (Figura 4.5). (BOOCH, 2005):

Um diagrama de atividades mostra o fluxo de uma atividade para outra em um sistema. Uma atividade mostra um conjunto de atividades, o fluxo seqüencial ou ramificado de uma atividade para outra e os objetos que realizam ou sofrem uma atividade. Esses diagramas são importantes principalmente para se fazer a modelagem da função de um sistema.

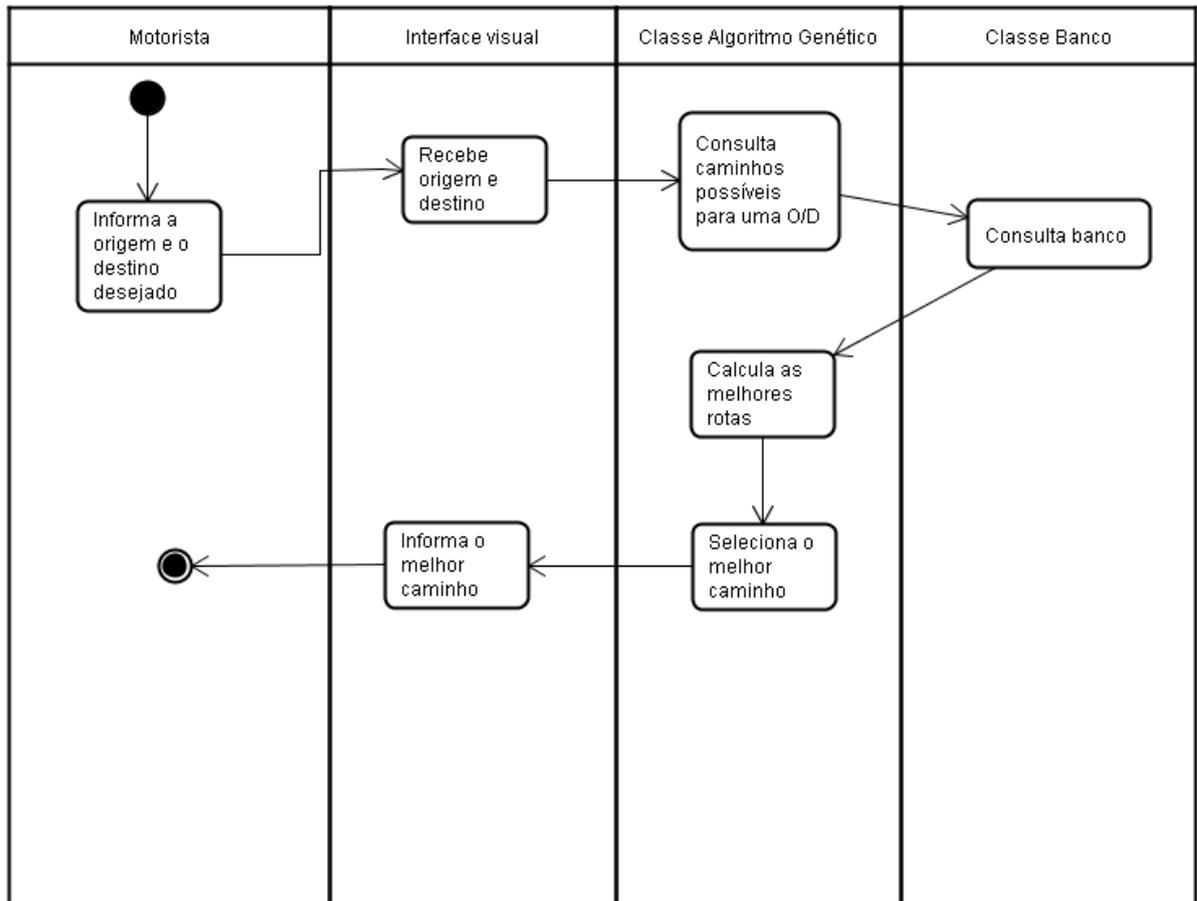


Figura 4.5: Diagrama de Atividades Fluxo de Processamento de Informação dada pelo usuário (motorista).

A fim de detalhar a funcionalidade de “Calcular a melhor rota”, foi criado o diagrama de atividades da Figura 4.6.

O diagrama “Fluxo de Cálculo de Rotas” (Figura 4.6), apresenta o fluxo que ocorrerá para determinar a melhor rota. Esse fluxo é determinado pela implementação do Algoritmo Genético.

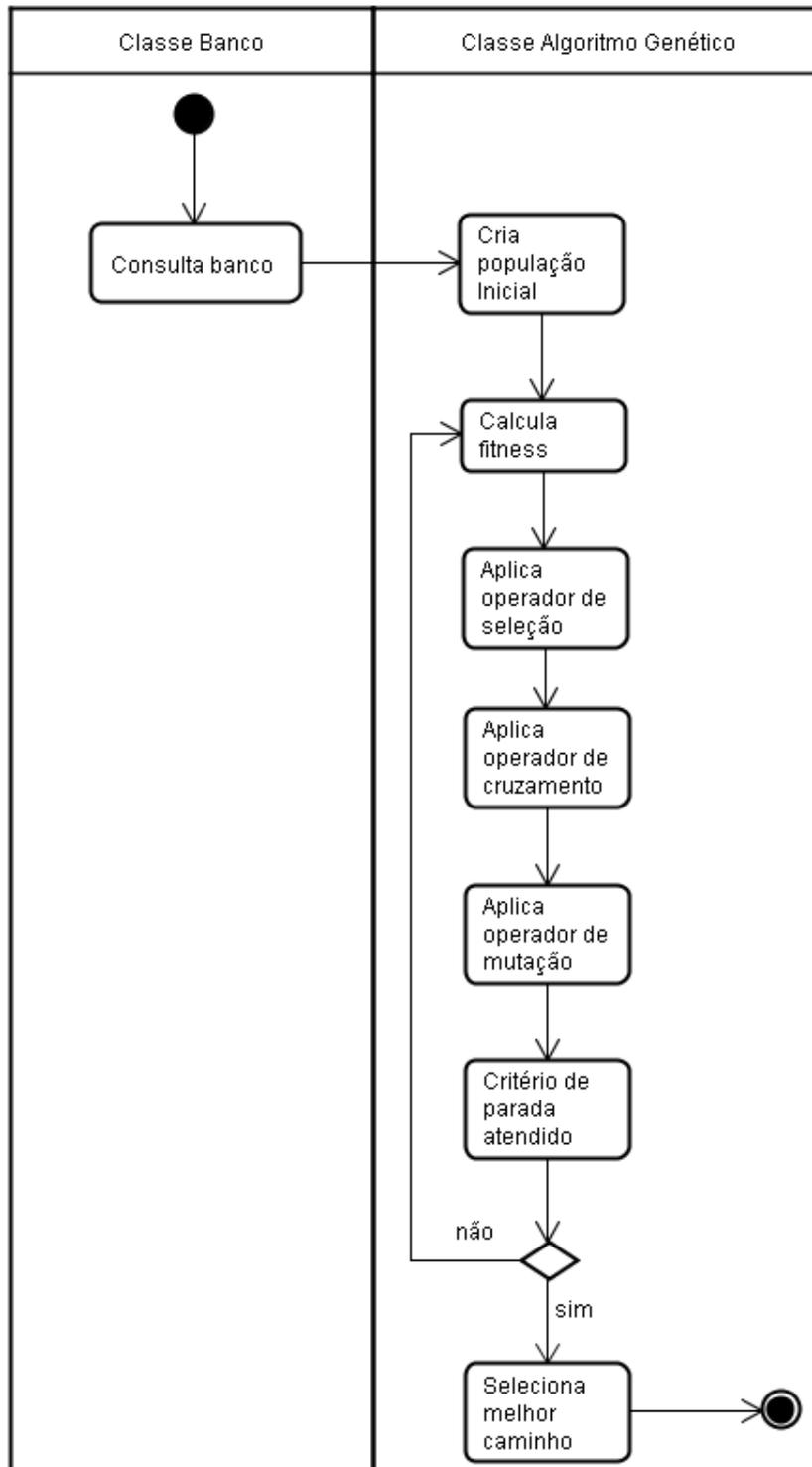


Figura 4.6: Diagrama de Atividade Fluxo de Cálculo de Rotas.

5 – CONCLUSÃO E PERSPECTIVAS

Nesta pesquisa foram realizados estudos a cerca dos Algoritmos Evolutivos cujo objetivo foi a solução de problemas de otimização de tráfego viário. Dentre os algoritmos estudados se encontram: Sistema Imunológico Artificial, Otimização por Colônia de Partículas, Otimização por Colônia de Formigas e Algoritmos Genéticos.

Com relação ao Sistema Imunológico Artificial, vê-se que pode ser amplamente aplicado na indústria de software de segurança, tais como antivírus.

O estudo dos algoritmos de Colônia de partículas e de formigas, ainda está amadurecendo, tendo ainda pouca aplicabilidade. Já as pesquisas com os Algoritmos Genéticos estão bem mais avançadas, com aplicações em diversas áreas da IA, tais como problemas de otimização complexos, gerenciamento de redes e muitas outras aplicações.

Após o estudo e fundamentação desses algoritmos, ficou evidente que o Algoritmo Genético apresenta melhor aplicabilidade para a solução do problema proposto.

Os Algoritmos Genéticos pertencem a uma classe particular dos Algoritmos Evolutivos e, portanto, possuem grande aplicação a problemas de otimização e busca.

O problema de otimização de tráfego viário envolve um espaço de busca e solução muito grande para que um algoritmo determinístico possa resolver em pouco tempo. Através de uma busca estocástica (probabilística), o Algoritmo Genético consegue eliminar as soluções menos ótimas, e sempre trazer as melhores soluções utilizando processos evolutivos (seleção, cruzamento, mutação). Dessa forma, o Algoritmo Genético é capaz de resolver problemas complexos em menos tempo, diminuindo a complexidade computacional.

O problema de engarrafamento atinge, hoje, grande parte das cidades do Brasil. Cidades grandes, como São Paulo, enfrentam congestionamentos com mais de 120 km de distância em horário de pico. O problema é agravante e precisa ser refletido. Contudo, se vê muitas pesquisas na área que buscam soluções de melhoria de trafegabilidade.

Como proposta desse trabalho, a implementação do Algoritmo Genético para busca de melhor caminho a seguir no momento da pesquisa pelo motorista propõe

dar o conforto ao motorista de saber que a rota informada para seguir está em ótimas condições de fazer uma viagem tranqüila, sem desconforto com congestionamentos.

Assim, o presente estudo apresenta melhorias na solução do equilíbrio dinâmico de tráfego, pois soluciona o problema em tempo real, consegue enorme vantagem no uso das rotas apropriadas evitando pontos de excessivos congestionamentos, trazendo satisfação da comunidade usuária da rede viária por minimizar os problemas de percurso, entre outros.

Portanto, este trabalho contribui para a solução de problemas de congestionamento da rede viária através da orientação aos usuários do sistema. Reforça o uso do AG como técnica de IA, utilizada na solução de problemas complexos e não lineares.

GLOSSÁRIO

Aresta	Elemento que compõe um grafo, também chamado de reta.
Bit	Representa um componente de uma cadeia de bits ou de cromossomos. Na biologia é chamado de gene.
Cromossomo	Representa uma solução para o problema, composto por uma cadeia de bits ou genes.
Crossover	Cruzamento que ocorre entre dois cromossomos pais para formar um cromossomo filho.
Fitness	Definição dada para representar as qualidades de um indivíduo, pode ser entendido também por adequabilidade. Também chamado por função de aptidão.
Gbest	Melhor valor da função de aptidão da colônia, denominado versão global.
Gene	Elemento que compõe uma cadeia de cromossomos. Na IA é também chamado de bit.
Grafo	É representado por um conjunto de arestas e vértices. Cada vértice está ligado a uma ou mais arestas.
Híbrido	Qualidade de tudo o que resulta de elementos de natureza distinta.
Linfócitos	São células constituintes dos leucócitos, ou células branca, do sangue. Tem uma função importante na defesa do corpo contra microorganismos.
Mutação	Fato que ocorre em um ou vários cromossomos, modificando um elemento de sua cadeia de bits.
Pbest	Melhor valor da função de aptidão da partícula, denominado versão local.
Pseudocódigo	Forma genérica de escrever um algoritmo, utilizando uma linguagem simples (no caso o português) sem necessidade de conhecer a sintaxe de nenhuma linguagem de programação.
Randômico	Aleatório.

Reprodução	Fato que seleciona o conjunto de cromossomos possíveis para solução do problema.
Robustez	Qualidade de robusto; que tem força, potente, resistente.
UML	Linguagem de Modelagem Unificada que utiliza conceitos da orientação a objetos.
Vértice	Elemento que compõe um grafo, também chamado de ponto.

REFERÊNCIA

BEIGHTLER, C. S.; PHILLIPS, D. T.; WILDE, D. J. *Foundations of optimization*. 2. ed. Englewood Cliffs, 1979. *Lógica: desenvolvimento e aplicação de ferramentas computacionais inspiradas em sistemas imunológicos artificiais*, Tese de doutorado, Faculdade de Engenharia Elétrica.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. *UML – Guia do Usuário*. Tradução de Fábio Freitas da Silva e Cristina de Amorin Machado. Rio de Janeiro: Elsevier, 2005.

COELHO, Leandro dos Santos. *Fundamentos, Potencialidades e Aplicações de Algoritmos Evolutivos*, Notas em Matemática Aplicada: 2. São Paulo: 2003.

GOLDBERG, David. *Genetic Algorithms in search, optimization and machine learning*. USA: Addison-Wesley Publishing Company, INC, 1989.

HOLLAND, J. *Adaptation in Natural and artificial systems*, University of Michigan Press, Ann Arbor. 1975.

KENNEDY, J. and EBERHART, R. C. *Particle swarm optimization*. Proc. IEEE int'l conf. on neural networks Vol. IV, pp. 1942-1948. IEEE service center, Piscataway, NJ, 1995.

LACERDA, E. G. M.; CARVALHO A. C. P. L. F. *Introdução aos algoritmos genéticos*. Em: Galvão, C. O. e Valença, M. J. S. (orgs) *Sistemas Inteligentes: aplicações a recursos hídricos e sistemas ambientes*. Ed. Universidade. UFRGS, Porto Alegre, 1999.

MARTINS, José Carlos Cordeiro. *Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML*. Rio de Janeiro: Brasport, 2004.

MICHALEWICZ, Z.; VIGNAUX, G. A.; HOBBS, M.; *A non-standard genetic algorithm for the nonlinear transportation problem*, ORSA Journal on Computing, v. 3, n. 4, pp. 307-316, 1991.

MIRANDA, Marcio Nunes de. *Algoritmos genéticos: fundamentos e aplicações*, 2000. Disponível em: <<http://www.gta.ufrj.br/~marcio/genetic.html>>. Acesso em: 27 fev 2006.

PONCE, André Carlos de Leon. *Algoritmos genéticos*. Disponível em: <<http://www.icmc.usp.br/~andre/research/genetic/index.htm>>. Acesso em: 27 fev 2006.

ROCHA, Jair do Prado e SARAMAGO, Sezimária F. Pereira. *Otimização por Colônia de Partículas*. UFU – Universidade Federal de Uberlândia, Uberlândia – MG, 2005.

SARAMAGO, Sezimária F. Pereira. *Métodos de Otimização Randômica: Algoritmos Genéticos e "Simulated Annealing"*, Notas em Matemática Aplicada: 6, São Carlos: 2003.

SILVA, Leandro Nunes de Castro. *Engenharia imunológica: Desenvolvimento e aplicação de ferramentas computacionais inspiradas em Sistemas Imunológicos Artificiais*. UNICAMP, Campinas, 2001. Disponível em: <http://www.dca.fee.unicamp.br/~vonzuben/research/lnunes_dout/index.html>. Acesso em: 20 nov 2005.

SILVA, Osmar Quirino; WEIGANG, Li; YAMASHITA, Yaeko; SILVA, Paulo César Marques; Andrew Maclver; Jörg Denzinger. *Otimização, Previsão e Orientação de tráfego viário utilizando Algoritmo Genético*. Rio de Janeiro: XVII Congresso de Pesquisa e Ensino em Transportes, Novembro, 10-14, 2003.

TELES, Wesley Martins. *Sistema adaptativo para web sites baseado no comportamento da formiga*. Tese de mestrado, Departamento de Ciência da Computação, Instituto de Ciências Exatas, UNB, Brasília: 2003.

REFERÊNCIA BIBLIOGRÁFICA

ARROYO, José Elias C. e ARMENTANO, Vinícius A. *Algoritmo Genético para o Problema do Caixeiro Viajante Multiobjetivo*. XXIII Congresso da Sociedade Brasileira de Computação. JAIA. Proceedins Anais -agosto de 2003, Campinas, SP - Brasil.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. *UML – Guia do Usuário*. Tradução de Fábio Freitas da Silva. Rio de Janeiro: Elsevier, 2000, 12ª reimpressão.

CÂMARA, Gilberto, CARVALHO, Marília Sá. *Análise Espacial de Eventos*. Publicações INPE, 2002.

CARVALHO, Maria Cecília M. de. *Construindo o saber – Metodologia científica: fundamentos e técnicas*. 7. ed. Campinas: Papirus Editora, 1998.

CARVALHO, Marília Sá. *Análise Espacial e Geoprocessamento*. Publicações INPE, 2002.

DARWIN, Charles Robert. *A origem das espécies*. Ed. Hemus. SP, 1979.

FURTADO, Antonio Luz. *Teoria dos grafos: algoritmos*. Editora Livros Técnicos e Científicos. SP, 1973.

GOODRICH, Michel T.; TAMASSIA, Roberto. *Estruturas de dados e algoritmos em Java*. Ed. Bookman. Porto Alegre, 2002.

GUEDES, Gilleanes T. A. *UML: uma abordagem prática*. Ed. Novatec, 2004.

GUO-GUANG, He; SHOU-FENG, Ma. *AI-based Dynamic Route Guidance Strategy and Its Simulation and A traffic flow forecast supported system based multi-agent*. Intelligent Transportation System Conference Proceedings – Oaklang(CA), USA – August 25 – 29,2001.

HOROWITS, Ellis, Sartaj Sahmi, Sanguthevar Rajasekaran. *Computer Algorithms*. Computer Sciences Press, ,New York ,1998.

KALIDAS, Ashok. *Estimation and Prediction of Time-Dependent Oringin-Destination Flows*. Massachussets Institute of Technology.1996.

MA, Tao and Baher Abdulhai. *Genetic Algorithm-Based Combinatorial Parametric Optimization for the Calibration of Microscopic Traffic Simulation Models*. IEEE2001, Intelligent Transportation Systems Conference Proceedins – Oaklang(CA), USA – August 25-29,2001.

NETTO, Alvim Antônio de Oliveira. *Metodologia da Pesquisa Científica: guia prático para a Apresentação de Trabalhos Acadêmicos*. Florianópolis: VisualBooks, 2005.

PEDRON, Ademar João. *Metodologia científica: auxiliar do Estudo, da Leitura e da Pesquisa*. Edição do autor, 3.ed. Brasília: Gráfica Redentorista, 2001.

PEDROSA, Bianca Maria; CÂMARA, Gilberto. *Modelagem dinâmica e geoprocessamento*. Publicações INPE, 2002.

PETERS, James F. ; PEDRYCZ, Witold. *Engenharia de software: teoria e prática*. RJ. Ed. Campus, 2001.

REZENDE, Solange Oliveira. *Sistemas Inteligentes: fundamentos e aplicações*. Ed. Manole. 1^a. Ed, 2003.

SHOU-FENG, Ma; GUO-GUANG, He and Shi-tong Wang. *A traffic flow forecast supported system based multi-agent*. *IEEE2001, Intelligent Transportation Systems Conference Proceedins – Oaklang(CA), USA – August 25-29,2001*.

SU, Chwen-Tzeng. *Locations and vehicle routing designs of physical distribution systems*. *Production Planning & control*, 1998. VOL. 9. No. 7. 650-659. Department of Industrial Management, National Yunlin Institute of Technology, 123 University Road, Sec. 3, Touliu, Taiwan, ROC.

VIGNAUX and Z. MICHALEWICZ, *A genetic algorithm for the linear transportation problem*, *IEEE Trans. Syst., Mans. & Cybern.*, Vol.21, no.3, pp.445-452, 1991.

WEIGANG, Li; YAMASHITA, Yaeko; SILVA, Osmar Quirino da; XIJUN, Deng; PRAZERES, Miguel; OLIVEIRA, Daniel Cristino Silva de. *Implementação do sistema de mapeamento de uma linha de ônibus para um sistema de transporte inteligente*. Seminário Integrado de Software e Hardware (SEMISH), Anais do XXI Congresso da Sociedade Brasileira de Computação, pp 72 – 85 30 de julho – 3 de agosto de 2001. Fortaleza – Brasil.

WEIGANG, Li; YAMASHITA, Yaeko; KOENDJBIHARE, Marlon Winston; JUCÁ, Ricardo Cezar de Moura; XIJUN, Deng; SILVA, Osmar Quirino da; MACIVER, Andrew. *The development and implementation of the system and data bank for the Intelligent Transportation System*. 9th World ITS Congress, October 14-18, 2002. Chicago, USA.