

## Um Algoritmo Evolutivo Baseado em Chaves Aleatórias Viciadas Aplicado ao Problema de Roteamento de Veículos com Janelas de Tempo

**Welton T. M. de Sousa\***      **Carlos A. Silva**

Instituto Federal de Minas Gerais - Departamento de Computação,  
34515-640, Sabará, MG  
E-mail: weltonthiago7@hotmail.com, carlos.silva@ifmg.edu.br.

### RESUMO

O Problema de Roteamento de Veículos (PRV), introduzido na literatura por [1], é um dos mais clássicos problemas da pesquisa operacional. A forma mais simples do PRV é o Problema de Roteamento de Veículos Capacitados (PRVC), onde uma frota de veículos, localizada inicialmente em um depósito, deve atender a um conjunto de consumidores com diferentes demandas de produtos a serem distribuídos por essa frota. Neste trabalho abordaremos o Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT), o qual é uma extensão do PRVC. Além da restrição de capacidade, são adicionadas restrições relacionadas ao horário em que cada consumidor exige ser atendido. Para cada consumidor  $i$ , é associado um intervalo de tempo ou janela de tempo  $[a_i, b_i]$  indicando o horário de início do atendimento, e um tempo de serviço  $s_i$ , determinando o período de tempo que o veículo deve aguardar a finalização das tarefas. O algoritmo evolutivo proposto para resolver este problema, consiste em determinar um conjunto de rotas com o menor custo possível, respeitando um determinado conjunto de restrições.

Neste trabalho propomos um método de solução para o PRVJT utilizando um algoritmo genético baseado em chaves aleatórias viciadas, denominado BRKGA (*Biased Random-key Genetic Algorithm*). Representamos a solução do problema por um vetor de  $n$  chaves aleatórias, sendo essas chaves números reais dentro do intervalo  $[0, 1)$ . Um decodificador é usado para mapear o vetor de chaves aleatórias e transformá-lo numa solução do PRVJT, ou seja, em um vetor de números inteiros, em que subsequências desses números correspondem a rotas formadas pelas cidades da instância utilizada. Inicialmente é gerada uma população de  $p$  vetores de chaves aleatórias e em seguida são selecionadas as melhores soluções. O conjunto de tamanho  $p_e$  contendo as melhores soluções é preservado para a próxima geração ou iteração do algoritmo. Um novo conjunto de tamanho  $p - p_e$  é adicionado às melhores soluções, compondo a nova população da próxima geração. Este novo conjunto de soluções é gerado a partir de combinações entre pais e filhos conforme uma dada distribuição probabilística.

De acordo com [2], um RKGA (*Random Key Genetic Algorithm*) evolui uma população, de  $p$  vetores de chaves aleatórias aplicando o princípio de Darwin, ou seja, há uma maior probabilidade de que os indivíduos mais aptos sobrevivam. Uma população inicial de  $p$  vetores de  $n$  chaves aleatórias é gerada de forma randômica. Na  $k$ -ésima geração, a população é particionada em dois conjuntos:  $p_e < p/2$  e  $p_{ne} = p - p_e$ , conjunto elite e não-elite, respectivamente. O conjunto  $p_e$  é constituído dos vetores que formam as melhores soluções, e conseqüentemente o conjunto  $p_{ne}$  é formado pelo restante da população. Para a  $k + 1$ -ésima geração, a nova população é formada pelos conjuntos  $p_e, p_m$  e  $p_r$ . O conjunto  $p_m$  é composto por vetores de chaves aleatórias, e, é denominado de mutante, pois desempenha o mesmo papel dos operadores de mutação nos algoritmos genéticos clássicos, ou seja, evita que a população estacione em um ótimo local. O conjunto  $p_r = p - p_e - p_m$  complementa a população. Os vetores deste último conjunto são gerados combinando pares de soluções da população da  $k$ -ésima geração,

---

\*Bolsista de Iniciação Científica PIBIC/IFMG

ambos escolhidos aleatoriamente, com a combinação uniforme seguindo os parâmetros de [4]. Sejam  $s = [0..s_i..0]$  e  $t = [0..t_i..0]$  dois vetores-pai e  $u = [0..u_i..0]$  um vetor-filho, onde 0 representa o depósito e  $s_i, t_i, u_i$  representam chaves aleatórias que indicam alguma cidade no PRVJT. O elemento  $u_i$  pode receber  $s_i$  ou  $t_i$  de acordo com as probabilidades  $\rho_s$  e  $\rho_t = 1 - \rho_s$ , respectivamente.

O algoritmo BRKGA-PRVJT proposto, está sendo implementado em linguagem C++, utilizando o compilador NetBeans IDE 7.31, e executado em um computador Intel(R) Core i3 2.5 GHz, com 3 GB de memória RAM, sob plataforma Windows 7 Professional 32 bits. Estão sendo utilizadas, para teste, as instâncias introduzidas por [3], usadas amplamente como referência de desempenho de algoritmos para o PRVJT. Para a implementação do BRKGA-PRVJT foi utilizado o modelo matemático apresentado em [5]. A função de avaliação foi definida como:

$$f(s) = \mu n_v(s) + \alpha e_c(s) + \beta v_r(s) + \sum_{(r,s) \in \mathcal{A}} d_{rs}, \quad (1)$$

onde, para cada solução  $s$ , o número de veículos é representado por  $n_v$ ; a soma de excessos de capacidade de todos os veículos é indicada por  $e_c$  e a soma das violações referentes às restrições de janela de tempo é representada por  $v_r$ . Além disso, a variável  $d_{rs}$  representa a distância entre os consumidores  $r$  e  $s$ , e  $\mathcal{A}$  é o conjunto dos arcos da solução. O Algoritmo 1 apresenta o pseudocódigo do BRKGA aplicado ao PRVJT.

---

#### Algoritmo 1 BRKGA-PRVJT

---

```

1  MelhorSolução ← PiorSoluçãoPossível;
2  enquanto critério de parada não for satisfeito faça
3    Gere a população  $P$  com vetores de  $n$  chaves aleatórias e a avalie usando (1);
4    Particione  $P$ ;
5    Aplique o cruzamento e mutação através das chaves aleatórias usando as probabilidades  $\rho_s$  e  $\rho_t$ ;
6    Atualize a população e usando (1) encontre SoluçãoGerada;
7    Aplique Busca Local na SoluçãoGerada;
8    se  $f(\textit{SoluçãoGerada}) < f(\textit{MelhorSolução})$ 
9      MelhorSolução ← SoluçãoGerada;
10   fim se
11  fim enquanto
12  Retorne MelhorSolução.

```

---

Esta pesquisa se encontra em fase de simulação. A metodologia de chaves aleatórias pode proporcionar uma nova forma de explorar o espaço de soluções para o PRVJT, além de que, as características probabilísticas facilitariam a não estagnação das soluções em ótimos locais.

**Palavras-chave:** *Problemas de Roteamento de Veículos, Algoritmo Evolutivo, Chaves Aleatórias.*

## Referências

- [1] G.B. Dantzig e J.H. Ramser, The truck dispatching problem, *Management Science*, (1959) 80-91.
- [2] M.G.C. Resende, Introdução aos algoritmos genéticos de chaves aleatórias viciadas, em “Simpósio Brasileiro de Pesquisa Operacional”, pp. 16-19, 2013.
- [3] M.M. Solomon, Algorithms for vehicle routing and scheduling problems with time windows constraints, *European Journal of Operational Research*, 35 (1987) 254-266.
- [4] W.M. Spears e K.A. DeJong, On the virtues of parameterized uniform crossover, em “Proceedings of the Fourth International Conference on Genetic Algorithms”, pp. 230-236, 1991.
- [5] K.C. Tan; L.H. Lee; Q.L. Zhu e K. Ou, Heuristic methods for vehicle routing problem with time windows, *Artificial Intelligence in Engineering*, 15 (2001) 281-295.