9

Referências Bibliográficas

1 DORIGO, M.; MANIEZZO, V.; COLORNI, A. **Positive feedback as a search strategy**. Milão: Dipartimento di Elettronica e Informatica, Politecnico di Milano, Itália, 1991. Relatório Técnico

2 DORIGO, M.; GAMBARDELLA, L. M. **Ant Colonies for the Travelling Salesman Problem**. Biosystems, Vol. 43, No. 2. pp. 73-81. Julho 1997.

3 BULLNHEIMER, B.; HARTL, R. F.; STRAUSS, C. **Applying the Ant System to the Vehicle Routing Problem**. In: 2nd International Conference on Metaheuristics MIC-97, Sophia-Antipolis, França, 21-24 de julho de 1997. Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization, 1999, pp. 285-296.

4 GAMBARDELLA, L. M.; TAILLARD, É.; AGAZZI, G. **MACS-VRPTW**: A Multiple Ant Colony System for vehicle routing problems with time windows. In: New Ideas in Optimization. Londres: McGraw-Hill, 1999. pp. 63-76.

5 LAWLER, E. L. et al. **The Traveling Salesman Problem**: A Guided Tour of Combinatorial Optimization. 1. ed. New York: John Wiley & Sons, 1985. 476p.

6 COOK, W. J. et al. **Combinatorial Optimization**. 1. ed. New York: John Wiley & Sons, 1998. 368p.

7 BARÁN, B.; SCHAERER, M. **A multiobjective Ant Colony System for vehicle routing problem with time windows**. In: Proceedings of the 21st IASTED International Conference of Aplied Informatics, Innsbruck, Austria, 10-13 de fevereiro de 2003.

8 CLARKE, G.; WRIGHT, J. **Scheduling of vehicles from a central depot to a number of delivery points**. Operations Research, Vol. 12, No.4. pp.568-581. Jul-Ago 1964.

9 PELLEGRINI, P. **Application of Two Nearest Neighbor Approaches to a Rich Vehicle Routing Problem**. Bruxelas: IRIDIA, Université Libre de Bruxelles, 2005. Relatório Técnico.

10 Benchmarks - Vehicle Routing and Travelling Salesperson Problems. http://www.top.sintef.no/vrp/benchmarks.html. Acesso em 22 de novembro de 2006.

11 DORIGO, M.; DI CARO, G.; GAMBARDELLA, L. M. **Ant Algorithms for Discrete Optimization**. Artificial Life, Vol. 5, No. 3, pp. 137-172. Abril 1999.

12 BULLNHEIMER, B.; HARTL, R. F.; STRAUSS, C. **An improved Ant System algorithm for the Vehicle Routing Problem**. Annals of Operations Research, Vol 89, No. 0. pp. 319-328. Janeiro 1999.

13 BRÄYSY O.; GENDREAU, M. **Metaheuristics for the Vehicle Routing Problem with Time Windows**. Oslo: SINTEF Applied Mathematics, Research Council of Norway, Noruega, 2001. Relatório Técnico.

14 NEVES, T. A.; SOUZA, M. J. F.; MARTINS, A. X. **Resolução do Problema de Roteamento de Veículos com Frota Heterogênea e Janelas de Tempo**. Ouro Preto: Departamento de Computação, UFOP, 2004. Relatório Técnico.

15 GAMBARDELLA, L. M. et al. **Ant Colony Optimization for vehicle routing in advanced logistics systems**. In: Proceedings of MAS 2003 – International Workshop on Modelling and Applied Simulation, Bergeggi, Itália, outubro de 2003.

10
Apêndice I

Algumas funções básicas do Matlab 6.0 foram utilizadas ao longo desta dissertação nas rotinas criadas e simuladas. A compreensão do funcionamento destas funções é condição necessária para que os algoritmos criados sejam entendidos em maior nível de detalhamento. Sendo assim, este Apêndice I se propõe a explicar de forma sucinta e objetiva as principais funções e comandos do Matlab que tenham sido utilizados ou referenciados ao longo deste trabalho.

Abaixo segue a listagem das funções básicas utilizadas, juntamente com suas aplicações:

- zeros(x, y) – cria uma matriz de zeros de tamanho x por y
- zeros(x) – cria uma matriz quadrada de zeros de tamanho x por x
- min(vetor) – retorna o menor elemento de "vetor"
- length(vetor) – informa o número de elementos de "vetor"
- max(x, y) – retorna o elemento maior, x ou y
- max(vetor) – retorna o maior elemento de "vetor"
- ismember(x, vetor) – retorna 0 se x não for membro de "vetor" e 1 se o for
- ones(x, y) – cria uma matriz de elementos 1 de tamanho x por y
- ones(x) – cria uma matriz quadrada de elementos 1 de tamanho x por x
- return – termina a execução da função imediatamente
- fix(x) – arredonda o número x em direção a zero (ou seja, para baixo se x > 0 e para cima se x < 0)
- rand(1) – gera um número aleatório entre 0 e 1