Exercícios referentes à AULA 06

As duas formas do algoritmo ACO foram implementadas em MATHCAD-2001® e para Problema do Caixeiro Viajante e estão disponíveis (4 programas!) para *download* na *homepage* do curso. Para quem não tem o software MATHCAD-2001®, estão disponíveis 2 arquivos AVI referentes a animações do algoritmo. Além disto há um programa executável codificado em DELPHI® e desenvolvido por ***Marcelo Kaminski Lenzi*** com o Manual do Usuário em documento também disponível, que permite *rodar* à vontade o PCV!

Divirtam-se!

Roberta & Evaristo

1. Rode[1] o exemplo do PCV dado em aula (com 16 cidades dispostas simetricamente!) variando o número de formigas [m>2], e os parâmetros α, β, ρ, τ_0 e Q. Faça depois um relatório circunstanciado em que se avalia o efeito de cada um destes parâmetros na convergência do método à solução exata [L_{min} =160].

2. Sugira uma nova forma de selecionar a próxima cidade utilizando no lugar do procedimento de roleta utilizado em aula a seleção por torneio (como mostrado na página 3 da Aula4). Refaça o PCV dado em aula com este novo procedimento.

3. Uma nova versão do ACO pode ser feita na versão original, reforçando, ao final de cada iteração, duas vezes a melhor rota. Com esta modificação refaça o problema do Exercício 1. Analise e comente seus novos resultados.

4. Uma outra aplicação interessante do ACO é na otimização dinâmica de processos. Um exemplo deste problema é a programação de temperatura em um reator químico operando em batelada onde é conduzida a reação : **A➔B➔C**. Sendo C_A a concentração

[1] Note que há duas possibilidades de *rodar* o exemplo: com os programas em MATHCAD disponibilizados ou com o programa do LENZI já disponível como programa executável!

adimensional do reagente **A** e C_B a concentração adimensional do produto intermediário **B**, a dinâmica do reator é descrita pelas equações diferenciais ordinária:

$$\begin{cases} \dfrac{dC_A(t)}{dt} = -k_1(T)\cdot C_A(t) \\ \dfrac{dC_B(t)}{dt} = k_1(T)\cdot C_A(t) - k_2(T)\cdot C_B(t) \end{cases}$$

onde : t: tempo adimensional entre 0 e 1;

$k_i(T) = k_i^{(0)} \cdot \exp\left(-\dfrac{E_i}{R_g \cdot T}\right)$ para i= 1 e 2.

Em t=0 tem-se: $C_A(0)=1$ e $C_B(0)=0$.

Deseja-se determinar o valor de T a cada tempo ao longo da batelada de modo que no final da batelada ($t_{final}=1$) o valor de x_2 seja o maior possível. Uma maneira de resolver este problema é reescrevê-lo na forma:

$$\begin{cases} \dfrac{dC_A(t)}{dt} = -u(t)\cdot C_A(t) \\ \dfrac{dC_B(t)}{dt} = u(t)\cdot C_A(t) - \alpha \cdot [u(t)]^{\beta} \cdot C_B(t) \end{cases} \quad \text{para } 0 < t \le 1, \text{ com } C_A(0)=1 \text{ e } C_B(0)=0;$$

$$\text{onde: } u(t) = k_1[T(t)] \Rightarrow u_{min} \le u \le u_{MAX}\,;\ \beta = \frac{E_2}{E_1} \text{ e } \alpha = \frac{k_2^{(0)}}{\left(k_1^{(0)}\right)^{\beta}}.$$

Considerando a seguir a discretização em t: $t_i = \dfrac{i}{N}$ para $i = 0, 1, \cdots, N$ e o valor de u(t) no intervalo i [i= 1, ...N], quando $t_{i-1} < t \le t_i$, como constante, isto é:

$u(t) = u^{(i)} \Rightarrow$ constante, para $t_{i-1} < t \le t_i$ sendo o valor de $u^{(i)}$ selecionado entre os (M+1) valores discretos de u: $u_k = u_{min} + \dfrac{k}{M}\cdot\left(u_{MAX} - u_{min}\right)$ para $k = 0, 1, \cdots, M$.

Mostre como este problema discreto de otimização poderia ser resolvido pelo algoritmo ACO.