MO824A/MC859A – Tópicos em Otimização Combinatória
Primeiro semestre de 2019

Atividade 5 (extra-classe)
Entrega: 25 de abril até meio-dia

Prof. Fábio Luiz Usberti (fusberti@ic.unicamp.br)
Prof. Celso Cavellucci (celsocv@ic.unicamp.br)

---

1 Objetivo

O objetivo desta atividade consiste na implementação (em grupos de **dois** ou **três** alunos) de uma metaheurística fundamentada em "Algoritmo Genético" (*Genetic Algorithm*) para a solução de um problema de maximização de uma função binária quadrática ("quadractic binary function" – QBF).

2 Problema MAX-QBF

Uma função binária quadrática (QBF) é uma função $f : \mathbb{B}^n \to \mathbb{R}$ que pode ser expressa como uma soma de termos quadráticos:

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \cdot x_i \cdot x_j$$

Onde $a_{ij} \in \mathbb{R} \ \ (i, j = 1, \ldots, n)$ são os coeficientes da função f. Em notação matricial, uma QBF pode ser expressa como:

$$f(\mathbf{x}) = \mathbf{x}' \cdot \mathbf{A} \cdot \mathbf{x}$$

Por exemplo:

$$\begin{aligned} f(\mathbf{x}) &= \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 2 \\ 0 & 0 & 3 \\ 0 & 0 & 4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ &= \begin{bmatrix} 0 & x_1 & (2x_1 + 3x_2 + 4x_3) \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ &= x_1x_2 + 2x_1x_3 + 3x_2x_3 + 4x_3^2 \end{aligned}$$

O problema de maximização de uma função binária quadrática (MAX-QBF) pode ser expresso como:

$$Z = \max_{\mathbf{x}} \ f(\mathbf{x}) \ ,$$

O MAX-QBF é um problema NP-difícil [1], mesmo que nenhuma restrição adicional seja imposta sobre as variáveis binárias $\mathbf{x}$. No entanto, se os coeficientes a_{ij} forem todos não-negativos, o problema torna-se trivial, uma vez que $x_i = 1 \ \ (i = 1, \ldots, n)$ é uma solução ótima.

3 Problema MAX-QBF com triplas proibidas

Uma variante do problema MAX-QBF é definida a seguir:

Problema MAX-QBFPT (“Maximum quadractic binary function with prohibited triples”): Considere o conjunto $\mathcal{T}$ de todas as triplas ordenadas, sem repetição, dos naturais de 1 a n, ou seja, $\mathcal{T} = \{(i, j, k) \in \mathbb{N}^3 : 1 \leqslant i < j < k \leqslant n\}$. No problema **MAX-QBFPT**, dado um conjunto de triplas proibidas $T \subseteq \mathcal{T}$, deseja-se maximizar uma função binária quadrática tal que x_i, x_j e x_k não podem ser todos iguais a 1, para todo $(i, j, k) \in T$. Este problema pode ser formulado da seguinte forma:

$$
\begin{array}{lll}
Max & Z = \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} \cdot x_i \cdot x_j & \\
s.a. & & \\
 & x_i + x_j + x_k \leqslant 2 & \forall (i, j, k) \in T \\
 & x_i \in \mathbb{B} & \forall i = \{1, \ldots, n\}
\end{array}
$$

Onde $a_{ij} \in \mathbb{R}$ $(i, j = 1, \ldots, n)$ são parâmetros do problema.

Para esta atividade, o conjunto de triplas proibidas T será definido da seguinte forma. Para cada natural $u \in [1, n]$ serão aplicadas duas funções $g, h : [1, n] \rightarrow [1, n]$ para gerar dois novos números que formarão uma tripla proibida. As funções g e h serão definidas tomando como base a seguinte função linear congruente l, tipicamente usada para a geração de números pseudo-aleatórios:

$$l(u) = \quad 1 + ((\pi_1 \cdot (u - 1) + \pi_2) \mod n)$$

Onde π_1 e π_2 são normalmente escolhidos como números primos.

Para impedir que $g(u) = u$, define-se a função g da seguinte forma:

$$g(u) = \begin{cases} l(u) & \text{se } l(u) \neq u \\ 1 + (l(u) \mod n) & \text{caso contrário} \end{cases}$$

Para impedir que $h(u) = u$ ou $h(u) = g(u)$, define-se a função h da seguinte forma:

$$h(u) = \begin{cases} l(u) & \text{se } l(u) \neq u \wedge l(u) \neq g(u) \\ 1 + (l(u) \mod n) & \text{se } (1 + (l(u) \mod n)) \neq u \wedge (1 + (l(u) \mod n)) \neq g(u) \\ 1 + ((l(u) + 1) \mod n) & \text{caso contrário} \end{cases}$$

Para esta atividade, os números primos utilizados para a função g são $\pi_1 = 131$ e $\pi_2 = 1031$; para a função h os números primos são $\pi_1 = 193$ e $\pi_2 = 1093$.

A partir das informações acima, torna-se possível definir o conjunto de triplas proibidas desta atividade como $T = \{(i, j, k) \in \mathcal{T} : \forall u \in [1, n], (i, j, k) = sort(\{u, g(u), h(u)\})\}$.

4 Requisitos da atividade

Esta atividade envolve a implementação de uma metaheurística fundamentada em Algoritmo Genético como um método de solução para o MAX-QBFPT. Para esta atividade você pode utilizar como base o Framework de Algoritmo Genético em Java, disponível no ensino aberto, desenvolvido pelos docentes desta disciplina.

Para esta atividade é necessário a implementação de pelo menos duas *estratégias evolutivas alternativas* discutidas em aula:

1. *Latin hypercube*
2. *Stochastic universal selection*
3. *Uniform crossover*
4. *Adaptive mutation*
5. *Steady-state*
6. *Diversity maintenance*

A atividade exige a entrega do código fonte e de um relatório (até 5 páginas) descrevendo brevemente as seguintes informações sobre a metaheurística desenvolvida:

- Descrição do problema: variáveis de decisão e modelo matemático.
- Metodologia: codificação de uma solução, geração da população inicial, método de seleção de indivíduos, método de recombinação ("crossover"), método de mutação, critérios de parada, estratégias evolutivas alternativas.
- Resultados: tabela de resultados e discussão.

Devem ser avaliados dois tamanhos de população, duas taxas de mutação e três estratégias evolutivas (padrão e duas alternativas). Desse modo, uma sugestão de possíveis configurações são:

1. `PADRÃO`: Algoritmo Genético com tamanho de população P_1, taxa de mutação M_1 e construção aleatória da população.
2. `PADRÃO+POP`: Algoritmo Genético `PADRÃO` mas com tamanho de população P_2.
3. `PADRÃO+MUT`: Algoritmo Genético `PADRÃO` mas com taxa de mutação M_2.
4. `PADRÃO+EVOL1`: Algoritmo Genético `PADRÃO` mas com estratégia evolutiva alternativa 1.
5. `PADRÃO+EVOL2`: Algoritmo Genético `PADRÃO` mas com estratégia evolutiva alternativa 2.

Procure organizar os resultados em uma tabela, avaliando qual a estratégia obteve o melhor desempenho.

5 Instâncias

Testes computacionais devem ser realizados com um conjunto de sete instâncias disponíveis no ambiente ensino aberto. Recomenda-se um tempo de execução para cada instância de pelo menos 30 minutos. Os nomes das instâncias, suas dimensões e os valores das soluções ótimas (quando conhecidos) são fornecidos a seguir:

Instância	$\|\mathbf{x}\|$	MAX-QBF (Z^*)	MAX-QBFPT (Z^*)
qbf020	20	**151**	**125**
qbf040	40	**429**	**366**
qbf060	60	**576**	$[508, 576]$
qbf080	80	**1000**	$[843, 1000]$
qbf100	100	$[1468, 1539]$	$[1263, 1539]$
qbf200	200	$[5385, 5826]$	$[3813, 5826]$
qbf400	400	$[14826, 16625]$	$[9645, 16625]$

Para fins de conferência, seguem as triplas proibidas para a instância qbf020:

```
[1, 12, 14]
[2, 3, 7]
[3, 14, 20]
[4, 5, 13]
[5, 6, 16]
[6, 7, 19]
[7, 12, 18]
[5, 8, 9]
[9, 18, 20]
[10, 11, 12]
[2, 4, 11]
[12, 13, 17]
[4, 10, 13]
[3, 14, 15]
[6, 15, 16]
[9, 16, 17]
[2, 8, 17]
[15, 18, 19]
[8, 10, 19]
[1, 2, 20]
```

6 Referências

1. Kochenberger, et al. The unconstrained binary quadratic programming problem: a survey. **J Comb Optim** (2014). 28:58–81. DOI:10.1007/s10878-014-9734-0.
2. Colin R. Reeves. Genetic Algorithms. In: M. Gendreau, J.-Y. Potvin (eds.), **Handbook of Metaheuristics**, International Series in Operations Research & Management Science 146, DOI: 10.1007/978-1-4419-1665-5.