XXXV SBPO
A pesquisa Operacional e os Recursos Renováveis

4 a 7 de novembro de 2003, Natal-RN

UMA HEURÍSTICA GRASP PARA O PROBLEMA DA MOCHILA QUADRÁTICA 0-1

Rodrigo Tavares Nogueira, DSc.
Universidade Estácio de Sá - Campos dos Goytacazes, Av. 28 de Março, 423
nogueirart@bol.com.br

Geraldo Galdino de Paula Jr., DSc.
Universidade Estadual do Norte Fluminense, Av. Alberto Lamego, 2000
galdino@uenf.br

Carlos Leonardo Ramos Póvoa, MSc.
Universidade Estadual do Norte Fluminense, Av. Alberto Lamego, 2000
l_povoa@zipmail.com.br

Resumo

Este trabalho considera o problema da mochila quadrática 0-1 (QKP). Por tratar-se de um problema NP-Hard sugere-se uma heurística GRASP para resolvê-lo em sua formulação QKP1. O algoritmo sugerido (HGQKP1) de complexidade $O(n^2)$, foi bem testado com a utilização de instâncias geradas aleatoriamente. Os bons resultados obtidos pelo algoritmo o tornam passível de utilização como "solver" de problemas de grande porte ou como fornecedor de "bounds" para métodos exatos enumerativos
Palavras-chave: mochila quadrática 0-1, heurística GRASP.

Abstract

This paper considers the 0-1 quadratic knapsack problem (QKP). Because the problem is NP-hard a GRASP heuristic is suggested to solve it in its formulation QKP1. The suggested algorithm, with complexity $O(n^2)$, was well tested with the utilization of randomly generated instances. The good outcomes observed make it suitable to be utilized as a solver in huge real problems or as a supplier of bounds for exact enumerative methods.
Keywords: 0-1 quadratic knapsack, GRASP heuristic.

1. Introdução.

A existência de um hiato entre pesquisa e prática, no campo das ciências administrativas, tem sido objeto de muitos estudos. Em programação inteira e otimização combinatória, esse hiato tem resultado, em sua maioria, das diferentes ênfases que "pesquisadores" e "práticos" têm colocado em métodos exatos de otimização e métodos heurísticos de solução. Os termos entre aspas não têm sentido pejorativo. Enquanto os pesquisadores, ao longo do tempo, concentraram seus esforços em métodos exatos de otimização, os práticos foram forçados a recorrer a heurísticas para a obtenção de soluções viáveis para problemas reais, em geral de grande porte. Felizmente, esse hiato tem diminuído sensivelmente, principalmente com o advento da teoria da complexidade ou da moderna teoria da ciência de computação (Karp,1986). Os práticos passaram a levar mais a sério os métodos exatos de otimização, principalmente porque estes melhoraram seus desempenhos até o ponto em que podem ser bem sucedidos na solução de problemas reais. Analogamente, os pesquisadores passaram a olhar com mais respeito os métodos heurísticos, desde que se cristalizou a impressão de que os problemas de otimização combinatória, em sua maioria, são extremamente difíceis de serem resolvidos (Cook, 1971) e, por isso mesmo, suas soluções para determinadas instâncias requerem a utilização de heurísticas.

Inúmeros problemas reais podem ser formulados como problemas de programação inteira, como exemplos, entre outros, problemas de empacotamento, alocação de tarefas, orçamento ou alocação de recursos financeiros e seleção de portfólios. Como problemas de programação inteira são, de modo geral, computacionalmente intratáveis, ou seja, não podem ser resolvidos de maneira exata por algoritmos em tempo polinomial determinístico, suas soluções ficam comprometidas quanto ao resultado. Este fato motiva a utilização de algoritmos heurísticos para proverem soluções viáveis tão boas quanto possíveis, eventualmente ótimas, para os referidos problemas.

Um problema de programação inteira que tem aplicações em finanças, tanto em alocação de recursos quanto em seleção de carteira de ações é o problema da mochila quadrática 0-1 (QKP). Uma característica deste problema é que ele é NP-Hard (Garey e Johnson, 1979) e, conseqüentemente, para problemas reais de maior porte, soluções exatas são computacionalmente inviáveis. Para a obtenção de soluções viáveis que sejam boas, no sentido de aproximarem-se do ótimo, faz-se necessária a utilização de heurísticas.

O presente trabalho aborda o problema da mochila quadrática 0-1 e sugere uma heurística GRASP ("greedy randomized adaptive search procedure"), inédita, para resolvê-lo. A escolha de um algoritmo GRASP fundamenta-se, principalmente, em sua simplicidade e engenhosidade de idéias e facilidade de codificação, fatores estes que, em absoluto, não diminuem sua performance.

O Capítulo2 apresenta o problema da mochila quadrática 0-1 (QKP), com suas características e formulações.

O Capítulo3 apresenta idéias, conceitos essenciais e referências de utilização das heurísticas GRASP.

O Capítulo 4 propõe um algoritmo GRASP para a formulação QKP1 do problema da mochila quadrática 0-1 e expõe sua complexidade.

O Capítulo 5 apresenta e analisa resultados computacionais do algoritmo proposto, testado para diversas instâncias geradas aleatoriamente.

Finalmente, no Capítulo 6 (Conclusão) são destacados os principais aspectos do trabalho.

2. O problema da mochila quadrática 0-1.

Problemas de mochila 0-1 são problemas de programação inteira, onde as variáveis são binárias. O nome "mochila" decorre da semelhança com o que se faz na prática quando se deseja acomodar em uma mochila um certo número de itens (roupas, acessórios, etc.). Problemas de mochila quadrática 0-1 consistem na otimização de funções objetivo quadráticas, sujeitas a restrições de mochila. Como referido anteriormente, estes problemas têm aplicações em alocações de recursos financeiros, seleção de projetos independentes e seleção de carteira de ações. Existem duas formulações para o problema da mochila quadrática binária: i) a mais tradicional (QKP1) consiste na maximização de uma função objetivo quadrática (Z), sujeita a uma restrição do tipo menor ou igual a uma capacidade; ii) a outra formulação (QKP2) consiste na minimização de uma função objetivo quadrática (Z), sujeita a uma restrição do tipo maior ou igual a uma capacidade.

Em sua formulação QKP1 o problema, sob forma de expressão, pode ser visto como:

$$\text{(QKP1) Maximizar } X^T\mathbf{Q}X$$
$$\text{s.a.:}$$
$$\mathbf{a}^T X \leq \mathbf{b}$$
$$X \in \{0, 1\}^n$$

onde $\mathbf{Q} = (\mathbf{q_{ij}}) \in \Re^{nxn}$ é uma matriz quadrada, suposta definida positiva e seus elementos correspondem a coeficientes de "lucro", $\mathbf{a} = (\mathbf{a_i}) \in \Re_+^{\,n}$ é o vetor dos coeficientes "técnicos" ou de "capacidades" e $\mathbf{b} \in \Re^+$ corresponde à capacidade da mochila.

O problema da mochila quadrática 0-1, principalmente em sua formulação QKP1, tem motivado o aparecimento de inúmeros algoritmos exatos para resolvê-lo. Morin e Marsten (1976) utilizaram um algoritmo recursivo baseado em programação dinâmica para gerar uma família completa de soluções viáveis não dominadas. Gallo, Hammer e Simeone (1980) propuseram algoritmos exatos, onde limites superiores ("upper bounds") são computados pela utilização de "upper planes" (planos superiores), que são funções lineares majoradas das variáveis binárias na função objetivo, no conjunto das soluções viáveis. Mathur, Salkin e Morito (1983) utilizaram procedimento "branch and search" que incluía um eficiente algoritmo para encontrar a solução relaxada (contínua) e uma regra de redução que computava limites "apertados" ("tight bounds") , tanto superiores quanto inferiores, para as variáveis inteiras. Hammer e Rader (1996) propuseram um algoritmo heurístico, baseado em linearização da função quadrática, e um algoritmo "branch and bound" que determinavam, a cada passo do procedimento, variáveis que poderiam ser fixadas em valores tanto viáveis quanto ótimos para certos subproblemas, com base em conclusões derivadas de relaxação Lagrangeana, relações de ordem e emparelhamento de restrições. Billionet, Faye e Soutif (1999) desenvolveram algoritmo "branch and bound" em que um bom limite superior foi obtido por relaxação Lagrangeana. Também, nesta linha, Caprara, Pisinger e Toth (1999) desenvolveram um algoritmo "branch and bound" que utiliza limites superiores apertados, dentro da faixa de 1% do valor ótimo, obtidos por relaxação Lagrangeana, e que reporta soluções exatas, efetivamente verificadas, para problemas de até duzentas variáveis, o que caracteriza um porte superior aos normalmente encontrados na literatura.

3. Heurísticas GRASP.

Apesar dos esforços dos pesquisadores, QKP1 continua sem solução exata para instâncias maiores. Por isso, reforça-se a idéia da utilização de algoritmos heurísticos para a obtenção de soluções viáveis que sejam próximas do valor ótimo ou, eventualmente, ótimas.Em função da intratabilidade computacional do problema da mochila quadrática 0-1, diversas heurísticas têm sido apresentadas para solucioná-lo em tempo polinomial, fornecendo boas soluções viáveis, eventualmente ótimas, para instâncias maiores. Como abordagens mais tradicionais destacam-se os algoritmos aproximativos e os gulosos ou míopes.Entre os primeiros podem ser citados, entre outros, os trabalhos de Hochbaum (1995), apresentando um esquema de aproximação totalmente polinomial para solução do problema, de Maurício e Maculan (1997) com um método de aproximação por região de confiança e de Rader e Woeginger (2002) que também apresentaram um esquema de aproximação totalmente polinomial, derivado de um algoritmo de programação dinâmica de complexidade pseudo-polinomial. As heurísticas gulosas ou míopes baseiam-se em uma idéia simples, escolhendo a cada passo do algoritmo a melhor decisão local. Muito embora elas sejam consideradas arbitrariamente ruins para o problema QKP1(Maurício, 1991), por sua simplicidade e por fornecerem boas soluções continuam sendo bastante utilizadas.

Não obstante a ampla utilização das heurísticas clássicas mencionadas, sabe-se que elas produzem, a cada rodada do algoritmo, uma única solução viável, que pode ser ótima ou não.As heurísticas GRASP ("Greedy Randomized Adaptive Search Procedure"), cujas idéias foram originalmente desenvolvidas por Feo e Resende (1995) são meta-heurísticas bastante adequadas para implementação em processadores paralelos e surgiram procurando conciliar a simplicidade conceitual das heurísticas gulosas com a limitação destas gerarem a mesma e única solução cada vez que acionadas. A solução GRASP passa por duas fases: *uma fase de construção*, baseada em critérios míopes de decisão, relaxados através de um parâmetro (α), de modo a produzirem um conjunto de candidatos a integrarem o conjunto solução, e de escolhas aleatórias desses candidatos, e, de *uma fase de melhoria* que consiste em uma busca local, que objetiva melhorar a solução míope aleatória construída. Assim, pela aleatoriedade das escolhas, várias rodadas do algoritmo resultam em várias soluções, possibilitando a escolha da melhor delas como a solução heurística do problema, o que implica maior probabilidade de se achar a solução ótima. Esta intuitiva e consistente simplicidade de idéias tem sido responsável pela ampla utilização de algoritmos GRASP em inúmeros problemas de otimização combinatória. Heurísticas GRASP têm sido bastante usadas em problemas de programação inteira, tanto em sua concepção original quanto em algoritmos GRASP híbridos. Um grande número de aplicações GRASP pode ser visto em bibliografia GRASP, mantida e freqüentemente atualizada por Resende. Mais recentemente, Ribeiro, Uchoa e Werneck (2001) trataram do problema de Steiner em grafos, utilizando uma GRASP híbrida, com perturbações ponderadas, em que a fase de construção GRASP é substituída pela utilização de diversas heurísticas de construção. Resende, Aiex e Ribeiro (2001) estudaram as distribuições de probabilidade do tempo de solução até um valor sub-ótimo alvo em cinco algoritmos GRASP que apareceram na literatura, para os quais estavam disponíveis os códigos fonte.

O procedimento GRASP pode ser sintetizado como abaixo:

Para a t-ésima rodada, t = 1, 2, ...,T
Início
Ler (instância, α);
Enquanto (critério de parada não satisfeito) faça
Procedimento construir solução;
Procedimento melhorar solução;
  Fim-enquanto;
  Escrever (melhor solução);
Fim.

onde, $0 \le \alpha \le 1$ é o parâmetro de relaxação do conjunto das variáveis candidatas à solução e também regulador do quão aleatória ou gulosa será a solução: valores extremos de α implicam, dependendo da formulação do problema em estudo, soluções totalmente aleatórias ou soluções totalmente míopes ou gulosas.

Observe-se que a solução escolhida para o problema será a melhor entre as T diferentes soluções.

4. Uma heurística GRASP para QKP1 (HGQKP1)

Nesta seção propõe-se um algoritmo GRASP, inédito, para obtenção de boas soluções viáveis para o problema da mochila quadrática 0-1 em sua formulação QKP1 (HGQKP1). No que concerne ao critério míope de escolha, levando-se em conta a formulação QKP1, percebe-se que incrementos na função objetivo ocorrem devidos tanto aos coeficientes de "lucro" da matriz **Q** quanto aos coeficientes "técnicos" de **a** (maior número de itens na mochila aumenta o valor da função objetivo). Com estas observações em mente, adotou-se o seguinte critério míope de decisão: escolher para candidato a integrar a mochila ou, equivalentemente, a solução, o elemento que apresentar a maior razão entre os coeficientes de "lucro" e os coeficientes "técnicos".O algoritmo proposto será apresentado sob forma de pseudocódigo, o que facilita sobremaneira sua compreensão.

Heurística GRASP para QKP1 (HGQKP1)

Inicialização

Ler (n, Q, a, b, α); // entrada de dados da instância e α
N: = { 1, 2,..., n}; // conj.^{to} de índices das variáveis fora da solução
I: = $\varnothing$; // conj.^{to} de índices das variáveis da solução
FO: = 0; VIAB: = 0; // variáveis função objetivo e viabilidade
Para j = 1, 2, ..., n faça $x_j^t = 0$; // vetor solução

```
Para t = 1, 2,...,T  faça                                   // loop externo das T rodadas

    Procedimento construir solução

  Enquanto ( N ≠ ∅ ) ∩ ( VIAB < b ) faça                  // critérios de parada

Início
    c(i)_{i∈N} : = q_ii + Σ_{j∈I} q_ij + Σ_{j∈I} q_ji ;       // determinação do "lucro"
    β: = max { c (i ) / a_i : i ∈ N };                      // critério míope de escolha
    R:= { i ∈ N : αβ ≤ c ( i ) / a_i ≤ β;                   // conjunto das candidatas
    k: = Random ( R );                                     // escolha aleatória da variável
    x_k^t : = 1;                                           // variável escolhida para solução
    VIAB: = VIAB + a_k;                                    // atualização da viabilidade
    FO: = FO + c (k);                                      // atualização da função objetivo
    I: = I ∪ {k};  N: = N – {k};                           // atualização conjuntos de índices
 Fim-enquanto;

Escrever ( x^t, FO, I, N, VIAB, b );                       // solução míope aleatória construída

    Procedimento melhorar solução
 Ler ( x^t, FO, I, N, VIAB, b );                            // solução construída
 SW: = Verdadeiro;                                          // variável "switch" do tipo Booleana
 Enquanto (I ≠ ∅ ) ∩ ( SW ) faça                            // critério de parada
 Início
    cg ( i )_{i∈I} : = Σ_{j∈I} q_ij + Σ_{j∈I} q_ji - q_ii ;      // determinação dos "lucros" individuais
                                                            das variáveis da solução
    cg ( l ):= min_{i∈I} { cg (i ) };                        // variável candidata a sair da solução
    S : = I – { l };                                        // conjunto auxiliar de índices
    cn ( i )_{i∈N} : = q_ii +  Σ_{j∈S} q_ij + Σ_{j∈S} q_ji ;    // determinação dos "lucros"
                                                            das variáveis fora da solução
    Z: = N;                                                 // conjunto auxiliar de índices
    Enquanto ( Z≠ ∅ ) ∩ ( SW ) faça                         // critério de parada
    Início
        cn ( r ): = max_{i∈Z} { cn ( i ) };                 // variável candidata a entrar na solução
         Se ( cn ( r ) > cg ( l )) faça                     // teste de atratividade
               Se ( a_r - a_l  + VIAB ≤ b ) faça            // teste de viabilidade
               Início
                 x_r^t : = 1; x_l^t : = 0                   // troca de variáveis
              FO: = FO – cg ( l ) + cn ( r );               // atualização da função objetivo
              SW: = Falso;                                  // atualização de variável
               Fim-se;
                 Z: = Z – { r };                            // atualização conjunto auxiliar
         Fim-se;

  Fim-enquanto;

   I: = I – {l};  N: = N – {r};                             // atualização conjuntos de índices
Fim-enquanto;
Escrever ( X^t , FO );                                      //  solução melhorada
```

Finalização

Escolher $\{X^t\}$ de maior FO; // escolha da mellhor das T soluções

Fim-para;

Fim.

Destacando, no algoritmo, I representa o conjunto de índices das variáveis ativas que estão na solução e N é o conjunto inicial de índices das variáveis inativas ou que não estão na solução. Os conjuntos de índices S e Z são apenas conjuntos auxiliares. O parâmetro α, como se sabe, é o parâmetro de relaxação do conjunto de variáveis candidatas à solução.

A análise da complexidade de HGQKP1, a seguir, baseia-se nas idéias e conceitos apresentados em Campello e Maculan (1994).

Proposição 1. A complexidade assintótica de HGQKP1 é de ordem O (n^2).

Prova: A complexidade assintótica do algoritmo é determinada, preponderantemente, pela verificação das complexidades dos procedimentos *construir solução* e *melhorar solução*, pois é nesses procedimentos que ocorre um maior número de instruções.

A complexidade do procedimento *construir solução* é influenciada em maior grau pela determinação de $c(i)_{i\in N} := q_{ii} + \sum_{j\in I} q_{ij} + \sum_{j\in I} q_{ji}$ e pela determinação de $\beta := \max \{ c(i) / a_i : i \in N \}$. É sabido que o máximo de n números pode ser obtido por algoritmo de ordem O (n). A determinação de $c(i)_{i\in N}$ envolve uma adição + 2m somas (onde $m \leq n$ é a cardinalidade de I) + 1 atribuição. No pior caso a mochila estará cheia. Portanto, simplificadamente, o número de operações elementares será de $n.(n+1)/2 + 2.n^2/2 + 2n$, ou $3n^2/2 + 5n/2$. Considerando-se as constantes k =4 e $n_0 = 1$, verifica-se que $3n^2/2 + 5n/2 \leq 4n^2$, $\forall\ n \geq 1$ o que, pela definição de ordem de complexidade, permite concluir-se que o procedimento *construir solução* tem complexidade de ordem O (n^2).

A complexidade do procedimento *melhorar solução*, de modo análogo, é influenciada em maior grau pela determinação de $cg(i)_{i\in I} := \sum_{j\in I} q_{ij} + \sum_{j\in I} q_{ji} - q_{ii}$. No pior caso (mochila cheia), a determinação de $cg(i)_{i\in I}$ envolverá n.(2n + 1) operações de adição (subtração) mais n atribuições. Simplificadamente, $2n^2 + 2n$ operações. Tomando-se as constantes k = 4 e $n_0 = 1$, verifica-se que $2n^2 + 2n \leq 4n^2$, $\forall\ n \geq 1$ e, portanto, por definição, o procedimento melhorar solução tem complexidade de ordem O (n^2).

A complexidade assintótica de HGQKP1, completando a prova, pela *regra das somas*, será de ordem max (O (n^2), O (n^2)), portanto de ordem O (n^2).

5. Testes de validação de HGQKP1.

A heurística HGQKP1 foi submetida a uma bateria de testes de validação, com a utilização de um computador pessoal, com processador Pentium III de 933 MHz e 128 Mb de memória. As soluções heurísticas, expressas em termos de valores da função objetivo (Z^H), foram comparadas, *quando possível*, com os valores ótimos (exatos) da função objetivo (Z^*). Como fator de avaliação, definiu-se um "erro" relativo ε como sendo igual ao quociente da diferença, em valor absoluto, entre os valores exatos e heurísticos da função objetivo, dividida pelo valor exato da função objetivo. Sob forma de relação, tem-se: $\varepsilon = | Z^* - Z^H | / Z^*$. Não se considerou o tempo de processamento (t), em segundos, como fator de comparação, embora o mesmo tenha sido registrado nos testes de HGQKP1. Nas situações onde não foi possível a comparação *resultado heurístico x resultado exato*, de qualquer modo, foram obtidas soluções heurísticas que se pretendem boas, vistos os resultados comparáveis.

A opção pela comparação com soluções exatas deveu-se a dois fatores principais: primeiro, pela intenção de avaliar a qualidade da solução GRASP, em termos de solução ótima e segundo, pelo desconhecimento da disponibilidade de outras meta-heurísticas formuladas especificamente para resolver QKP1.

Os testes foram realizados para cinco instâncias geradas aleatoriamente, por grupos de 50, 100, 150, 200, 250 e 500 variáveis binárias, considerando-se, matrizes quadradas (Q), com densidades (Δ) de 25%, 50%, 75% e 100%, e valores do parâmetro de relaxação (α) iguais a 0,4, 0,6 e 0,8. Como se vê, os testes foram significativos em quantidade.

O algoritmo HGQKP1 foi bem testado, até o limite de 200 variáveis binárias, graças à possibilidade de utilização de potente algoritmo (exato) "branch and bound", desenvolvido por Caprara, Pisinger e Toth (1999), já referido anteriormente, cujo código foi tornado disponível por Pisinger. O limite de 200 variáveis é imposto por construção do algoritmo exato. Cumpre observar que a saída deste algoritmo não informa o tempo de processamento. Para fins de simplificação, considerou-se insolúvel a instância que levava mais de trinta e cinco minutos (relógio) para apresentar solução. Para números de variáveis superiores a 200, somente resultados GRASP foram obtidos.

Na geração das instâncias utilizou-se um gerador de instâncias aleatórias integrado ao algoritmo "branch and bound" e que também foi utilizado por HGQKP1. O gerador assume, para os coeficientes $\mathbf{a_i}$ do vetor $\mathbf{a}$, distribuição uniforme em determinado intervalo [1; l], onde l é o limite superior do intervalo. Os valores $\mathbf{q_{ij}}$ da matriz $\mathbf{Q}$ também são uniformemente distribuídos no intervalo, com probabilidade igual à densidade Δ da matriz. A capacidade **b** da mochila é calculada como uma "média" dos elementos $\mathbf{a_i}$. Como se vê, as instâncias geradas dessa maneira são correlacionadas, o que aumenta significativamente o grau de dificuldade de obtenção de soluções exatas.Em todos os casos, os dados de entrada são valores inteiros não negativos (mochila supermodular). As densidades da matriz também devem ser informadas. A utilização de instâncias aleatórias, em tese, garante a representatividade dos testes.

Considerando-se todos os resultados obtidos, para cada valor de α, tomaram-se, tanto para o erro relativo ε quanto para o tempo de processamento t, os respectivos valores médios. A Tabela 1 mostra o desempenho de HGQKP1, até 200 variáveis, considerando-se valores médios para ε (ε_M) e para o tempo de processamento (t_M), em segundos, para cada valor de α utilizado. Aqui, cabe a observação de que não foram encontradas soluções exatas para três instâncias de 200 variáveis testadas, duas delas com densidade de 25 % e uma com densidade de 50 %. Os valores médios, para estes casos, somente consideram os resultados realmente obtidos.

Conforme se vê, os resultados médios obtidos foram bastante satisfatórios, em termos de valores de função objetivo, destacando-se os alcançados pela utilização do parâmetro de relaxação α com valor igual a 0,8, o qual apresentou erros médios (ε_M) inferiores a 1% do valor ótimo, exceto no caso particular de uma instância de 100 variáveis binárias, com densidade Δ igual a 75%, que apresentou erro ε igual a 0,0575, o que subiu o erro médio ε_M para cerca de 0,0130 ou 1,3% do valor ótimo.

De qualquer modo, os resultados para α igual a 0,8 foram superiores aos obtidos pelos outros valores de α. Como valores de α mais próximos de 1 implicam, para HGQKP1, soluções próximas de uma solução essencialmente míope, os bons resultados para α = 0,8 atestam o bom critério míope adotado.

De fato, o critério míope de escolha (max { c (i) / a_i : i $\in$ N }, onde $c(i)_{i\in N} := q_{ii} + \sum_{j\in I} q_{ij} + \sum_{j\in I} q_{ji}$) talvez seja o mais adequado por considerar incrementos na função objetivo tanto como função dos coeficientes de **Q** quanto dos coeficientes de **a.**

Tabela 1: Resultados médios de HGQKP1 para instâncias com até 200 variáveis.

		α = 0,4		α = 0,6		α = 0,8	
Δ (%)	n	ε_M	t_M (s)	ε_M	t_M (s)	ε_M	t_M (s)
25	50	0,0048	5	0,0008	5,2	O,0000	5,2
	100	0,0078	32	0,0029	31,8	0,0003	32,4
	150	0,0133	109	0,0061	109,2	0,0013	110
	200	0,0049	302	0,0037	304,7	0,0010	307,3
50	50	0,0105	5	0,0010	5	0,0000	5
	100	0,0169	30,8	0,0033	30,8	0,0011	31
	150	0,0081	99,8	0,0035	101,2	0,0007	101,6
	200	0,0089	277,5	0,0048	279	0,0007	280,8

75	50	0,0189	4,6	0,0003	4,8	0,0000	4,8
	100	0,0176	28,2	0,0131	28,6	0,0130	28,8
	150	0,0124	101,2	0,0042	102,2	0,0018	103,2
	200	0,0131	242,8	0,0077	249,6	0,0038	252,2
100	50	0,0052	4,6	0,0031	4,6	0,0000	4,8
	100	0,0036	23,8	0,0021	24,4	0,0007	24,8
	150	0,0183	92	0,0015	92,4	0,0011	92,4
	200	0,0085	262,4	0,0062	266,2	0,0024	271,2

A Tabela 2 a seguir mostra os resultados médios de HGQKP1, tanto em valor de função objetivo (Z^H_M) quanto em tempo de processamento (t_M), em segundos, para cada valor de α utilizado, nos casos de 250 e 500 variáveis binárias.

Tabela 2: Resultados médios de HGQKP1 para instâncias de 250 e 500 variáveis.

Δ (%)	n	$\alpha = 0,4$		$\alpha = 0,6$		$\alpha = 0,8$	
		Z^H_M	t_M (s)	Z^H_M	t_M (s)	Z^H_M	t_M (s)
25	250	46.368	651	46.353	654	46.419	657
	500	835.125	4.993	835.262	5.067	835.755	5.070
50	250	331.574	634	333.170	636	333.677	638
	500	967.289	5.406	968.605	5.467	969.239	5.514
75	250	358.549	639	358.715	642	359898	646
	500	968.625	5.317	972.071	5.347	972.908	5.378
100	250	375.827	645	376.705	649	377.637	652
	500	1.196.349	5.286	1.215.341	5.315	1.216.984	5.318

Os melhores resultados obtidos para as instâncias de 250 e 500 variáveis binárias deveram-se, também, à fixação de α igual a 0,8.

Uma observação interessante é a de que o tempo de processamento, embora não tenha sido considerado como relevante na avaliação de HGQKP1, foi maior para valor de densidade Δ de matriz igual a 25%. Variações nos valores de α também implicam modificações, pequenas, a propósito, no tempo de processamento.

6. Conclusão.

O trabalho apresentado aborda a importância do problema da mochila quadrática 0-1 (QKP) que em sua formulação QKP1 tem aplicações práticas de cunho econômico-financeiro, principalmente em alocação de recursos e seleção de projetos independentes. Por ser NP-Hard, e, portanto, computacionalmente, de difícil solução exata, propõe-se uma heurística GRASP, HGQKP1, inédita, para resolvê-lo.

Heurísticas GRASP são meta-heurísticas de filosofia simples e de fácil assimilação que se baseiam, em sua primeira fase, em critérios míopes de escolha aleatória das variáveis que farão parte da solução construída. Em uma segunda fase, a solução construída passa por um procedimento de melhoria (busca local). Por seu caráter aleatório, heurísticas GRASP são adequadas ao processamento em paralelo e permitem soluções diferentes a cada rodada do algoritmo.

O algoritmo HGQKP1 foi submetido a um número considerável de testes, os quais revelaram resultados bem favoráveis em termos de valores de funções objetivo,comparativamente aos resultados ótimos obtidos pela utilização de um algoritmo exato "branch and bound". Destacam-se os resultados obtidos para valor do parâmetro α de relaxação fixado como igual a 0,8, em que todos eles, à exceção de um, apresentaram valores heurísticos de função objetivo a menos de 1% do valor ótimo, o que confirma a hipótese de que quanto melhor o critério guloso adotado, mais próximas são as soluções ótima e míope ($\alpha = 1$). Tais resultados lhe conferem a possibilidade de utilização tanto como "solver" de problemas reais quanto como fornecedor de "bounds" para métodos exatos de enumeração.

Referências Bibliográficas

BILLIONET, A., FAYE, A., SOUTIF, E. (1999). A new upper-bound for the 0-1 quadratic knapsack problem. *European Journal of Operations Research*, 112: 664-672.

CAMPELLO, R. E., MACULAN, N. (1994). *Algoritmos e heurísticas:* desenvolvimento e avaliação de performance. Niterói: EDUFF.

CAPRARA, A., PISINGER, D., TOTH, P. (1999). Exact solution of the quadratic knapsack problem. *Journal of Computing*, 11, (2): 125-137.

COOK, S. A. (1971). The complexity of theorem – proving procedures. *Proc. 3rd Annual ACM Symposium on the theory of computing*. p. 151-158.

FEO, T., RESENDE, M. (1995). Greedy randomized adaptive search procedure. *Journal of Global Optimization*, 16: 1-24.

GALL0, G., HAMMER, P. L., SIMEONE, B. (1980). Quadratic knapsack problems.

Mathematical Programming, 12: 132-149.

GAREY, M. R., JOHNSON, D. S. (1979). *Computers and intractability*: a guide to the theory of NP – completeness. San Francisco, CA.: Freeman.

HAMMER, P. L., RADER, D. J. (1997). Efficient methods for solving quadratic 0-1 knapsack problems. *INFOR*, 35:170-182.

HOCHBAUM, D. (1995). A nonlinear knapsack problem. *Operations Research Letters*, 17: 103-110.

KARP, R. M. (1975). On the computational complexity of combinatorial problems. *Networks*, 5: 45-68.

MATHUR, K., SALKIN, H. M., MORITO, S. (1983). A branch and search algorithm for a class of nonlinear knapsack problems. *Operations Research Letters*, 2: 155- 160.

MAURICIO, D., MACULAN, N. (1997). A trust region method for zero-one nonlinear programming. *Operations Research*, 31: 331-341.

MAURICIO, D. (1991). *Métodos de resolução para mochila quadrática*. Dissertação de mestrado, COPPE, UFRJ.

MORIN, T. L., MARSTEN, R. E. (1976). An algorithm for nonlinear knapsack problems. Management Science, 22: 1147-1157.

RESENDE, M., AIEX, R., RIBEIRO, C. (2001). Probability distribution of solution time in GRASP: an experimental investigation. www.optimization-online.org.

RADER, D. J., WOEGINGER, G. J.(2002).The quadratic 0-1 knapsack problem with series-parallel support. *Operations Research Letters*, 30: 159-166.

RIBEIRO, C., UCHOA, E., WERNECK, R. (2001). An hybrid GRASP with Perturbations for the Steiner problem in graphs. www.optimization-online.org.