

**Universidade Federal de São João del-Rei**  
**Heurísticas e Metaheurísticas**  
**Atividade Avaliativa 3**

**Felipe Francisco Rios de Melo**

## **1. INTRODUÇÃO**

A meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedure*) é um algoritmo comumente aplicado a problemas de otimização combinatória. Como diversos métodos construtivos, a aplicação do GRASP consiste em criar uma solução inicial e depois efetuar uma busca local para melhorar a qualidade da solução. Seu diferencial para outros métodos está na geração dessa solução inicial, baseada nas três primeiras iniciais de sua sigla em inglês: gulosa (*Greedy*), aleatória (*Randomized*) e adaptativa (*Adaptive*).

A estratégia de construção de uma solução no GRASP consiste na definição de um critério de avaliação dos elementos que podem ser inseridos em um conjunto que, ao final do processo, será uma solução para o problema de otimização que se pretende resolver. Esse critério adapta-se à solução já construída, de forma que a valoração dos elementos muda durante a construção da solução. Entretanto, esse critério não é tomado como referência absoluta para a decisão do próximo elemento a ser inserido, havendo uma escolha aleatória entre os melhores elementos a cada iteração.

Uma das formas de aperfeiçoar o GRASP é com a estratégia Path-Relinking. Com o Path-Relinking, após uma solução ter sido gerada pelo GRASP, ela é combinada com uma solução aleatoriamente selecionada em um conjunto de soluções elite. O objetivo do GRASP com o Path-Relinking é encontrar soluções intermediárias de melhor qualidade, que estejam entre duas boas soluções.

Este trabalho prático teve como objetivo, propor uma implementação do Path-Relinking à implementação do GRASP para o problema da mochila binária vistos nas aulas práticas.

## **2. IMPLEMENTAÇÃO**

### **2.1. Forma de integração**

O Path-Relinking foi integrado ao GRASP como intensificação entre um ótimo local e uma solução elite. Neste método de integração, o path-relinking é aplicado após a busca local é aplicado a cada iteração, onde que a solução inicial é a solução obtida

na busca local e a solução guia é uma solução aleatoriamente selecionada a partir do conjunto elite.

Este método é mais eficiente, em comparação ao método da pós-otimização, pois a cada iteração ele busca, através da vizinhança entre a solução inicial e solução guia, encontrar uma solução melhor que a solução obtida na busca local, proporcionando um rápido aperfeiçoamento da solução, iteração a iteração.

## **2.2. Conjunto Elite**

Foi estabelecido um tamanho fixo para o conjunto elite, limitado a 10 soluções.

Os critérios para a entrada de uma solução ao conjunto foram:

- se o conjunto de elite ainda não se preenche totalmente (tamanho atual < 10), a solução candidata entra.
- se o conjunto está completo, verifica se a solução candidata a entrar é melhor que a pior solução do conjunto elite, a solução candidata entra.

O critério de saída está relacionado ao segundo critério de entrada:

- se a solução candidata a entrar é melhor que a pior solução do conjunto elite, a pior solução do conjunto elite sai.

## **2.3. Direção**

A direção do caminho da solução corrente até a solução guia, é progressiva. Pois considerando que as soluções do conjunto elite são as mais bem avaliadas, o path-relinking implementado caminha em direção a melhor solução.

## **2.4. Características escolhidas para o GRASP nas duas abordagens**

O  $\alpha$  é um parâmetro de crucial importância para o desempenho do GRASP. Uma seleção criteriosa de seu valor proporciona um equilíbrio entre a diversificação e a qualidade da solução. Este parâmetro determina quais elementos serão colocados na LRC em cada iteração da fase construtiva. Como este problema é de maximização, o  $\alpha = 0$  corresponde a um algoritmo puramente guloso, enquanto  $\alpha = 1$  equivale a uma construção puramente aleatória.

Foi utilizado o  $\alpha = 0.7$ , e como será visto na seção de análises, foi obtido soluções de qualidade.

A lista restrita de candidatos (LRC) tem seus elementos escolhidos baseados em valores. É dado pela seguinte formula:

$$\{ j \mid c_j \geq c_{\max} - \alpha ( c_{\max} - c_{\min} ) \}$$

onde  $c_{\min}$  = custo do melhor elemento,  $c_{\max}$  = custo do pior elemento.

O algoritmo de busca local usado foi o VND (Variable Neighbourhood Descent), este é método de busca local que consiste em explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança, aceitando somente soluções de melhora da solução corrente e retornando à primeira estrutura quando uma solução melhor é encontrada.

### 3. ANÁLISES

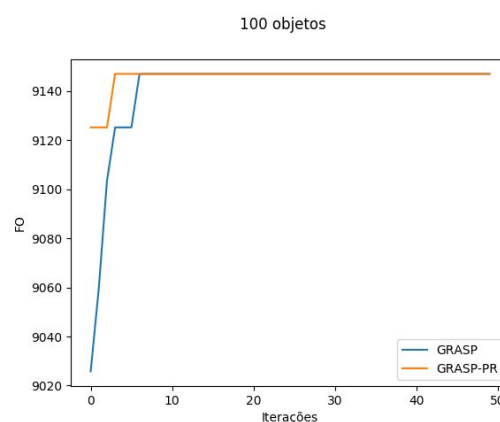
Os algoritmos GRASP e GRASP + PR (Path-Relinking), foram executados com as instâncias de 100 objetos, 200 objetos e 500 objetos do arquivo large\_scale (disponível no link: [http://artemisa.unicauca.edu.co/~johnnyortega/instances\\_01\\_KP/](http://artemisa.unicauca.edu.co/~johnnyortega/instances_01_KP/)).

Para cada instância, foi executado 10 vezes o algoritmo, tanto para o GRASP quanto para o GRASP + PR.

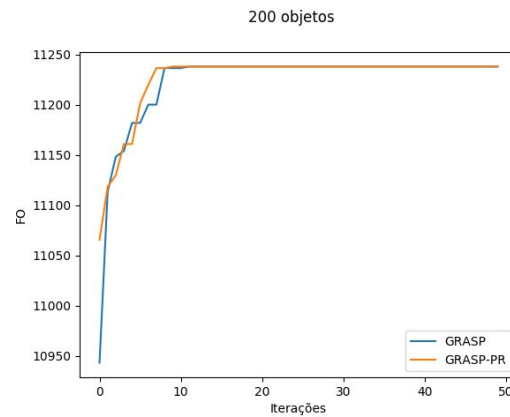
A leitura e tratamento do arquivo de saída e geração dos gráficos, foi realizado de maneira automatizada pelo arquivo *generateAnalysis.py* presente na pasta *analysis*.

A seguir temos as análises dos dados obtidos destas execuções:

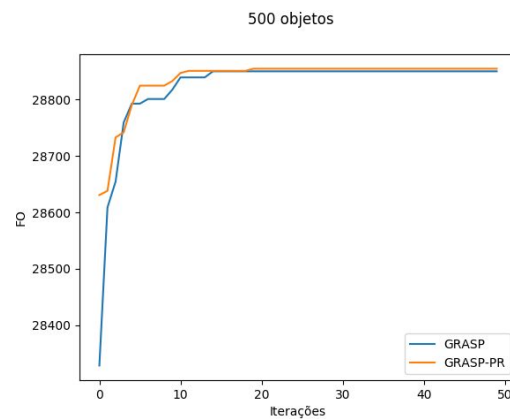
**GRÁFICO 1.** Comparação GRASP x GRASP + PR com instância de 100 objetos.



**GRÁFICO 2.** Comparação GRASP x GRASP + PR com instância de 200 objetos.



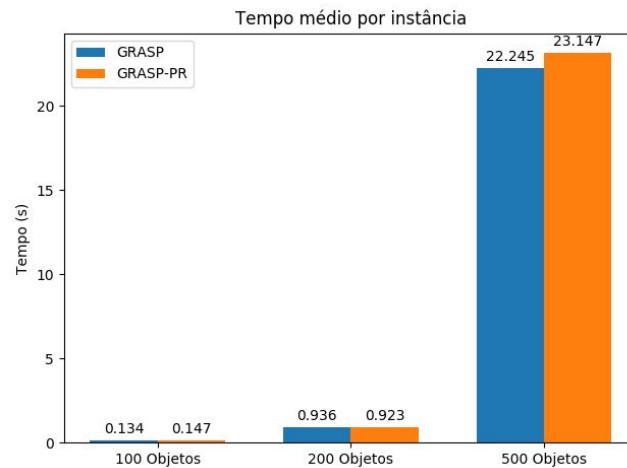
**GRÁFICO 3.** Comparação GRASP x GRASP + PR com instância de 500 objetos.



Em todas as instâncias, observamos que já na primeira iteração, o Path-Relinking consegue melhorar consideravelmente a solução encontrada, o que contribui para acelerar a convergência da função objetivo.

Analisando de modo geral os gráficos, conseguimos comprovar este fato ao observar a taxa de convergência do GRASP com Path-Relinking em relação ao GRASP sem Path-Relinking. Ele chega à melhor solução com um menor número de iterações.

**GRÁFICO 4.** Tempo computacional médio por instância



Em relação ao tempo computacional, é possível notar que para as instâncias testadas, a diferença de tempo foi pequena. O que nos leva a concluir, que o uso do GRASP com Path-Relinking para estas instâncias é uma escolha bem viável.

Não foi realizado muitos testes com 10 execuções (como foi feito para as outras instâncias) com instâncias acima de 1000 objetos, devido ao tempo de execução gastos pelos mesmos. Porém, realizando algumas execuções esporádicas com uma instância de 1000 objetos, a diferença no tempo entre o GRASP e o GRASP + PR, foi de aproximadamente 20s, em um tempo total de execução de mais ou menos 8 minutos.

A ordem de complexidade da função Path-Relinking implementada é  $O(n^2)$ . Como visto anteriormente, o Path-Relinking não influencia muito no tempo computacional a medida que o número de objetos aumenta. O grande “vilão” neste caso é o algoritmo de busca local VND.

#### 4. CONCLUSÃO

Foi possível neste trabalho, implementar e analisar, em relação a qualidade da solução e tempo computacional gasto, o método Path-Relinking para aperfeiçoar a meta-heurística GRASP. Bem como compreender a influência de certos parâmetros para encontrar a melhor solução de maneira satisfatória.

## **5. REFERÊNCIAS BIBLIOGRÁFICAS**

Frinhani, R. D. M. D. (2011). GRASP com Path-Relinking para agrupamento de dados biológicos.

GRASP, Wikipedia. Disponível em: <<https://pt.wikipedia.org/wiki/GRASP>>

MARTINS, Alexandre Xavier et al. Uma nova estratégia de geração de rotas e um algoritmo VND aplicado ao problema de roteirização e atribuição de comprimentos de onda.