GRASP DUPLAMENTE REATIVA PARA O PROBLEMA GENERALIZADO DE LOCALIZAÇÃO DE *P*-MEDIANAS

Caroline Nascimento Parajara
Universidade Estadual do Norte Fluminense
carolparajara@homail.com

Geraldo Galdino de Paula Junior
Universidade Estadual do Norte Fluminense
galdino@uenf.br

Resumo

Este trabalho relata a experiência de construir e testar um novo procedimento duplamente reativo para a GRASP, destinado à solução do problema generalizado de localização não-capacitado de p-medianas. São usados dois parâmetros de reação para controlar a busca de soluções na fase de construção da GRASP. O uso simultâneo dos dois parâmetros reativos permitiu criar uma disciplina para alocação dos clientes aos agrupamentos das medianas: clientes mais próximos são alocados primeiro. Foram feitos experimentos computacionais com o algoritmo desenvolvido e aqui apresentado, usando dados da biblioteca OR Beasley. Os resultados alcançados são interessantes, confirmam a boa reputação desta metaheurística e ampliam o espaço por ela ocupado na solução de problemas combinatórios de médio e grande portes que, com frequência, aparecem na economia.
Palavras-Chaves: Problema de localização; GRASP; p-medianas

Abstract

This paper reports the experience of building and testing a new procedure doubly reactive for the GRASP for the solution of the generalized uncapacited p-medians location problem. Two parameters of reaction are used to control the search of solutions in the construction phase of GRASP. The simultaneous use of two parameters allowed to create a discipline of the allocation of clients to groups of medians: customers closer are allocated first. Were made computational experiments with the algorithm developed and presented here using data from the library OR Beasley. The results obtained are interesting, confirm the reputation of this metaheuristic and expand the space it occupied in solving combinatorial problems large and medium sized companies that often appear in the economy.
Keywords: Location problem; GRASP; p-medians.

1. Introdução

Neste trabalho um novo procedimento duplamente reativo para a GRASP destinado à solução do problema generalizado de localização não-capacitado de *p*-medianas é descrito e testado. Introduzida por Feo & Resende (1995), a GRASP adquiriu boa reputação na solução de problemas combinatórios, com aplicações relevantes em vários setores da economia, conforme descrito na literatura, Festa & Resende (2009). Sua evolução vem ocorrendo pela via de esquemas reativos e recursos de religação de caminhos, Frinhani (2011), Boudia et al. (2007), Resende & Werneck (2002), Delmaire et al. (1999), Battiti (1996), que oferecem aprimoramentos na eficiência do método. Esquemas paralelos da GRASP também têm permitido bons avanços, Aiex et al. (2003), Martins et al. (2000) e Alvim & Ribeiro (1998).

2. O problema generalizado de localização de *p*-medianas

Modelos locacionais são usados em projetos que visam localizar armazéns, depósitos ou centros de distribuição numa cadeia de suprimento, buscando minimizar o custo ou tempo de atendimento aos clientes. Problemas de localização associam-se a um conjunto de instalações ou facilidades que minimizam o custo de atendimento a demandas e obedecem a um elenco de restrições. Os componentes principais que ajudam a descrever um problema locacional são: uma métrica para tratar as distâncias ou custos de transporte entre clientes e instalações; clientes, a respeito dos quais se assume que já estão localizados em pontos ou rotas; instalações, que podem ser armazéns, hubs, medianas, depósitos, centros de distribuição (cd's), etc., a serem localizadas; e um espaço físico ou região geográfica em que estão situados os clientes e os pontos candidatos para receber as instalações. Modelos locacionais são também aplicados a instalações indesejadas, Berman & Drezner (2000), que movimentam ou guardam produtos nocivos à natureza e à saúde humana, maximizando a distância entre essas instalações e o público em geral.

O problema generalizado de *p*-medianas procura identificar, no máximo, *p* locais onde construir as instalações relativas ao problema de localização. Este trabalho aborda a versão não-capacitada desse problema. Seu modelo é

$$\text{Minimizar} \sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij} + \sum_{i\in I} f_i y_i \tag{1}$$

$$\sum_{i\in I} x_{ij} = 1, \qquad \forall j \in J \tag{2}$$

$$x_{ij} \le y_i, \qquad \forall i \in I,\ \forall j \in J \tag{3}$$

$$\sum_{i\in I} y_i \le p \tag{4}$$

$$y_i \in \{0,1\}, \qquad \forall i \in I \tag{5}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in I,\ \forall j \in J \tag{6}$$

Onde (1) é a função critério representada pela soma de uma parcela de custos (ou distâncias) de transporte entre os clientes e as facilidades, com outra parcela, sendo esta de custos fixos das instalações. (2) garante que cada cliente será atendido por uma única instalação. (3) representa a condição pela qual, clientes só serão atendidos a partir de instalações abertas, ($y_i = 0 \Rightarrow x_{ij} = 0$, ou, se a instalação *i* está fechada, não há fluxo de bens ou serviços de *i* para o cliente *j*). (4) é a restrição generalizada de *p*–medianas. Em (5), $y_i = 1$ se a *i*-ésima instalação é aberta, $y_i = 0$, caso contrário. Em (6), $x_{ij} = 1$ se a instalação $i \in I$

atende o cliente $j \in J$, e $x_{ij} = 0$, caso contrário.

3. A metaheurística GRASP

A metaheurística GRASP (Greedy Randomized Adaptive Search Procedure), Feo & Resende (1995), é um procedimento iterativo em que cada iteração consiste de duas fases: uma de construção que produz soluções viáveis, e outra de busca local (ou de melhoria) em que um ótimo local é procurado na vizinhança da solução construída. A melhor solução encontrada ao longo de todas as iterações realizadas é retornada e aceita como boa solução, porém, sem a garantia de ser ótima.

A fase de construção começa produzindo iterativamente uma solução viável, elemento a elemento. A cada iteração, os próximos elementos a serem incluídos na solução são colocados em uma lista C de candidatos, possivelmente, seguindo algum critério de ordenação. A seleção é baseada em uma função gulosa capaz de medir a conveniência de cada candidato para compor a solução. O lado probabilístico da GRASP se expressa pela seleção aleatória de cada elemento a partir de um agregado que contém os melhores candidatos, chamado de lista restrita de candidatos (LRC).

A fase de busca local procura melhorar a solução criada na fase de construção. Essa busca se dá através de trocas sucessivas em que se avalia a conveniência de substituir o pior elemento que compõe a solução pelo melhor que ainda não foi escolhido para compô-la.

A seguir, a metaheurística GRASP é apresentada no seu formato genérico através do Algoritmo 1 (GRASP), do Algoritmo 2 (fase de construção) e do Algoritmo 3 (fase de busca local ou melhoramento).

Algoritmo 1 GRASP

Requer: GRASP (ϕ (□), g(□), V (□), MAX$_{GRASP}$, s)

1. $\phi^* \leftarrow \infty$
2. $s^* \leftarrow s$
3. **Para** (iter = 1, ... , MAX$_{GRASP}$) **faça**
4. Construção (g (□), α , s)
5. Busca Local (ϕ (□), V (□), s)
6. **Se** $(\phi(s) < \phi^*)$ **então**
7. $s^* \leftarrow s$
8. $\phi^* \leftarrow \phi(s)$
9. **Fim Se**
10. **Fim Para**
11. $s \leftarrow s^*$
12. Retornar s
13. **Fim** GRASP

Algoritmo 2 Construção

Requer: GRASP-CONSTRUÇÃO (g (□), α , s)

$s^* \leftarrow \varnothing$
Inicializar o conjunto C de candidatos
Enquanto $C \neq \varnothing$ **faça**
    $g_{\min}(\theta) = \min\{g(\theta) : \theta \in C\}$
    $g_{\max}(\theta) = \max\{g(\theta) : \theta \in C\}$
    $LRC = \{\theta \in C : g(\theta) \le g_{\min}(\theta) + \alpha(g_{\max}(\theta) - g_{\min}(\theta))\}$
    Selecionar, aleatoriamente, um elemento $\theta \in LRC$
    $s \leftarrow s \cup \{\theta\}$
    Atualizar o conjunto C de candidatos.
Fim Enquanto
Retornar s
Fim GRASP-CONSTRUÇÃO

Algoritmo 3 Melhoria

Requer: GRASP-MELHORIA $(\phi$ (□), V (□), $s)$

$s^* \leftarrow s$ { s^* é a melhor solução encontrada até o momento}
$H = \{s' \in V(s) : \phi(s') < \phi(s)\}$
Enquanto $|H| > 0$ **faça**
    Selecionar $s \in V$
    Se $\phi(s) < \phi(s^*)$ **então**
        $s^* \leftarrow s$
        $H = \{s' \in V(s) : \phi(s') < \phi(s)\}$
    Fim Se
    $s \leftarrow s^*$
Fim Enquanto
Retornar s
Fim GRASP-MELHORIA

Cada especialização da GRASP para classes específicas de problemas tem a sua forma, que não é única, de projetar as fases de construção e busca local. Neste trabalho a GRASP é especializada para resolver o problema de localização, de acordo com o modelo definido pelas relações de (1) a (6). Além disso, um novo mecanismo duplamente reativo é descrito, juntamente com resultados de experimentos computacionais que procuram revelar possível aprimoramento dessa metaheurística.

A terminologia aqui adotada usará os termos, mediana, hub ou cd (centro de distribuição) para os pontos ou nós que conterão algum tipo de instalação. Cidades ou clientes serão os pontos ou nós que formarão os agrupamentos a serem atendidos pelas medianas, hubs ou cd's.

4. Cálculo da função gulosa da GRASP para *p*-medianas

Neste trabalho, a função gulosa necessária na fase de construção da GRASP, relativa à $i-$ésima mediana, $i \le p$, é definida por

$$g_i(C_i) = g^i_{\min} + \alpha(g^i_{\max} - g^i_{\min}) \tag{7}$$

onde $g^i_{\min}$ é escolhido de tal maneira que $g^i_{\min} < g^i_{\max}$ e

$$g^i_{\max}(C_i) = \frac{\sum_{j \in C_i} \Delta_{ij}}{|C_i| - 1}, \quad |C_i| > 1 \tag{8}$$

ou

$$g^i_{\max}(C_i) = \frac{f_i + \sum_{j \in C_i} \Delta_{ij}}{|C_i| - 1}, \quad |C_i| > 1 \tag{9}$$

Tanto na função (7) quanto nas funções (8) e (9), C_i representa um possível agrupamento da mediana i. Δ_{ij} é a diferença entre a distância da mediana i à cidade de menor distância, e a distância da mediana i à cidade j. No caso da função (9), f_i é o custo fixo da mediana i. Em qualquer dos dois casos, o agrupamento (com sua mediana) será tão melhor quanto menor for a média dos custos representada pelas funções.

Nas fórmulas (8) e (9), o -1 no denominador se refere ao nó (cidade) mais próxima da mediana, que não participa do somatório. Para ela - cidade mais próxima - a distância adicional, em relação a distância mínima, é zero $(\Delta_{ij} = 0)$.

Para ilustrar o uso da função gulosa, considera-se uma matriz de distâncias para 10 cidades, de acordo com a Tabela 1. Aleatoriamente a cidade 8 é escolhida candidata a mediana.

A cidade mais próxima é a 3 que está a uma distância 5 da cidade 8. As demais cidades para serem atendidas por 8, percorrerão esses 5 mais uma distância adicional: a cidade 1 está distante 12: 5+7. A cidade 2 está distante 18: 5+13, e assim por diante. A seguir são feitos os outros cálculos com auxílio da Tabela 1.

- $g^8_{\max} = \dfrac{\sum_{j \in C_8} \Delta_{8j}}{|C_8| - 1} = \dfrac{7+13+1+9+4+5+3+35}{8} = \dfrac{77}{8} = 9{,}625 \rightarrow 10$
- $g^8_{\min} = 0$
- $\alpha = 0{,}7$
- $g^8 = g^8_{\min} + 0{,}7 \times \left(g^8_{\max} - g^8_{\min}\right) = 0{,}7 \times 10 = 7$

Escolha dos clientes da mediana candidata 8, com $\alpha = 0{,}7$. Em cada linha usa-se $\dfrac{\sum_{j \in C_i} \Delta_{ij}}{|C_i| - 1}$.

TABELA 1- Matriz de distâncias para ilustrar o cálculo da função gulosa, com 10 cidades

	1	2	$\boxed{3}$	4	5	6	7	$\boxed{8}$	9	10
1	0	*	*	*	*	*	*	*	*	*
2	*	0	*	*	*	*	*	*	*	*
3	*	*	0	*	*	*	*	*	*	*
4	*	*	*	0	*	*	*	*	*	*
5	*	*	*	*	0	*	*	*	*	*
6	*	*	*	*	*	0	*	*	*	*

7	*	*	*	*	*	*	0	*	*	*
→ $\boxed{8}$	12	18	$\boxed{5}$	6	14	9	10	0	8	40
	5+7	5+13	5+0	5+1	5+9	5+4	5+5	mediana	5+3	5+35
9	*	*	*	*	*	*	*	*	0	*
10	*	*	*	*	*	*	*	*	*	0

- $\frac{1}{1} = 1 < 7$, escolhe o 4
- $\frac{1+3}{2} = 2 < 7$, escolhe o 9
- $\frac{1+3+4}{3} = 2,\ldots < 7$, escolhe o 6
- $\frac{1+3+4+5}{4} = 3,\ldots < 7$, escolhe o 7
- $\frac{1+3+4+5+7}{5} = 4 < 7$, escolhe o 1
- $\frac{1+3+4+5+7+9}{6} = 5,\ldots < 7$, escolhe o 5
- $\frac{1+3+4+5+7+9+13}{7} = 6 < 7$, escolhe o 2
- $\frac{1+3+4+5+7+9+13+35}{8} = 9{,}625 > 7$, NÃO escolhe o 10

Neste caso, o agrupamento da mediana 8 é {4, 9 , 6, 7, 1, 5, 2}.

Escolha dos clientes da mediana candidata 8, com $\alpha = 0{,}3$. Neste caso, $g^8 = g^8_{\min} + 0{,}3 \times \left(g^8_{\max} - g^8_{\min} \right) = 0{,}3 \times 10 = 3$.

- $\frac{1}{1} = 1 < 3$, escolhe o 4
- $\frac{1+3}{2} = 2 < 3$, escolhe o 9
- $\frac{1+3+4}{3} = 2,\ldots < 3$, escolhe o 6
- $\frac{1+3+4+5}{4} = 3,\ldots > 3$, NÃO escolhe o 7

Neste caso, o agrupamento da mediana 8 é {4, 9, 6}. Observa-se que quanto menor for o α, mais se restringe a escolha de cidades clientes da mediana candidata.

Os primeiros projetos de metaheurística GRASP trabalhavam com um α, $0 \le \alpha \le 1$, que era calibrado até se tornar um parâmetro fixo do método. Com a evolução da pesquisa pertinente, foram adotados esquemas reativos em que é permitido o α sofrer variações durante as sucessivas interações do algoritmo.

5. Novo procedimento reativo para a GRASP - a metáfora da peneira

O procedimento reativo para a GRASP é conduzido provocando uma variação do α, $0 \le \alpha \le 1$, durante o processo iterativo, na fase de construção do método.

5.1. Variando os orifícios da tela da peneira

Uma das possibilidades da GRASP reativa consiste em começar com o α bem pequeno visando com isso escolher/alocar para cada mediana os nós (clientes) mais próximos. Em seguida o α é aumentado progressivamente, permitindo que clientes mais distantes de cada mediana sejam alocados. O α funcionaria como uma espécie de controlador dos orifícios da tela da peneira que é iniciada com uma tela bem fina. No início, o material peneirado é constituído de grãos muito pequenos - ou finos (que representam os clientes mais próximos a serem alocados). A tela da peneira é progressivamente substituída por outra tela mais grossa, ou seja, com espaços maiores entre os fios da tela. O material peneirado vai se tornando progressivamente granulado com grãos maiores (representando os clientes mais distantes a serem alocados por último). Cria-se com isso uma disciplina de alocação: clientes mais próximo são alocados primeiro.

5.2. Variando o tamanho da peneira

Simultaneamente ao processo de aumentar progressivamente os orifícios da tela da peneira, fazendo o α crescer no intervalo [0,1], o tamanho da peneira, em si, é também controlado de acordo com $g = g_{\min} + \alpha(g_{\max} - g_{\min})$, onde

$$\begin{aligned} g_{\min} &= g_{\min}(\gamma, g_{\max}) \\ &= \gamma \cdot g_{\max}, \qquad 0 \le \gamma < 1 \end{aligned}$$

Com a alteração iterativa do parâmetro γ, o valor da diferença $g_{\max} - g_{\min}$ se altera também de forma iterativa. Cria-se com isso um processo GRASP_2 REATIVA, em que a reação se dá, simultaneamente, pelos dois procedimentos descritos.

5.3. A fase de melhoria ou busca local da GRASP

Existe uma variedade de procedimentos possíveis para conduzir esta fase da metaheurística. Alguns desses esquemas de melhoria podem ser vistos em García (2001). Na melhoria das soluções da GRASP, adotada neste trabalho, o algoritmo compara o custo de alocação ao agrupamento original - vindo da fase de construção - de cada cidade cliente com o custo de sua alocação aos demais agrupamentos. Se outro agrupamento oferece um custo menor de atendimento, ou seja, uma proximidade melhor da mediana, a cidade cliente é retirada do agrupamento em que ela está no momento, e alocada ao agrupamento de menor custo de alocação. Ao se mover um cliente de um agrupamento para outro, ele deixa de incorrer em um custo para o agrupamento antigo e passa a incorrer em um custo para o novo agrupamento. Para haver uma troca, a diferença entre o que se ganha e o que se perde com a troca, deve diminuir o custo de alocação do cliente em questão.

6. GRASP modificada: Inclusão dos parâmetros de reação

Os algoritmos 4 e 5, que são, respectivamente, os algoritmos 1 e 2 alterados, incorporam as modificações que tornam a GRASP duplamente reativa. Esta modificação afeta somente a parte inicial do algoritmo e a fase de construção. Na implementação o intervalo

[0,1] é discretizado para fornecer valores numéricos aos parâmetros de reação. Por exemplo, $0 \leq \alpha \leq 1$ assume os valores do conjunto $\{0, 0.1, \ldots, 0.9\}$, no decorrer das iterações. Do mesmo modo, $0 \leq \gamma < 1$ assume os valores de $\{0, 0.1, \ldots, 0.9\}$.

O procedimento reativo também aconteceu de forma que, primeiramente, apenas 10% das cidades candidatas a mediana eram utilizadas. Em seguida, 20% das mesmas. E assim sucessivamente até serem utilizadas 100% das cidades candidatas a mediana. No final, o algoritmo mostrou com qual porcentagem de cidades candidatas a mediana foi encontrada a melhor solução. Esse procedimento permitiu gerar uma maior variedade de soluções, tendo assim, uma maior possibilidade de se alcançar a solução ótima dos problemas.

Diga-se, $\beta\%$ × Total de candidatas a mediana, para o primeiro problema, $2\beta\%$ × Total de candidatas a mediana para o segundo problema, e assim por diante, até que $k \times \beta\%$ se tornasse igual a 100%, ou sejam, todos os supostos candidatos a medianas. Cuidado foi tomado para que em nenhum dos problemas menores a quantidade de medianas candidatas ficasse inferior ao total máximo de medianas que se desejava localizar. A solução de todos esses problemas menores produziu um aglomerado de medianas, do qual participaram todas as medianas escolhidas como solução dos problemas menores.

Algoritmo 4 GRASP

Requer: GRASP (ϕ (□), g(□), V (□), MAX_{GRASP}, s)

$\phi^* \leftarrow \infty$
$s^* \leftarrow s$
Para (iter = 1, ..., MAX_{GRASP}) **faça**
    Construção (g (□), $0 \leq \alpha \leq 1, 0 \leq \gamma < 1$, s)
    Busca Local (ϕ (□), V (□), s)
    Se $(\phi(s) < \phi^*)$ **então**
        $s^* \leftarrow s$
        $\phi^* \leftarrow \phi(s)$
    Fim Se
Fim Para
$s \leftarrow s^*$
Retornar s
Fim GRASP

Algoritmo 5 Construção

Requer: GRASP-CONSTRUÇÃO (g (□), $0 \le \alpha \le 1$, $0 \le \gamma < 1$, s)

1. $s^* \leftarrow \varnothing$
2. Inicializar o conjunto C de candidatos
3. **Enquanto** $C \neq \varnothing$ **faça**
4. $\quad g_{\max}(\theta) = \max\{ g(\theta) : \theta \in C\}$
5. $\quad g_{\min}(\theta) = \gamma \cdot g_{\max}(\theta), \theta \in C\}$
6. $\quad LRC = \{\theta \in C : g(\theta) \le g_{\min}(\theta) + \alpha(g_{\max}(\theta) - g_{\min}(\theta))\}$
7. Selecionar, aleatoriamente, um elemento $\theta \in LRC$
8. $\quad s \leftarrow s \cup \{\theta\}$
9. Atualizar o conjunto C de candidatos.
10. **Fim Enquanto**
11. Retornar s
13. **Fim** GRASP-CONSTRUÇÃO

7. Experimento computacional usando a biblioteca OR-Beasley

O experimento computacional foi desenvolvido no ambiente C++ do Visual Studio 2010, usando computador baseado em processador de múltiplos núcleos.

No experimento projetado o β foi considerado igual a 10. Isto significa que o primeiro problema menor foi resolvido com 10% das cidades com menor distância total às demais 111 cidades, como candidatas a mediana. O segundo problema menor foi resolvido com 20% dessas cidades, e assim por diante, até que 100% das cidades tornaram-se candidatas a mediana na solução do problema de localização em foco. Portanto, foram resolvidos 9 instâncias menores mais a instância original, isto é, 10 instâncias. No final, o algoritmo mostra com qual porcentagem foi encontrada a melhor solução.

Foi usada a biblioteca de problemas teste de OR-Beasley, Beasley (1990), onde cada instância é fornecida com um número de ligações entre cidades que não corresponde ao total de ligações possíveis para aquela instância. A matriz de distância foi então completada para cada instância usando o algoritmo de Floyd-Warshall, usado para encontrar o caminho de menor distância entre todas as cidades da rede.

A parcela de custos fixos da função objetivo não foi utilizada nos testes - os dados da biblioteca OR-Beasley não fornecem essa informação. As Figuras 1 e 2, em escalas diferentes para preservar a razão de aspecto, mostram os dois casos extremos dos experimentos.

A Figura 1 mostra os resultados do experimento que executa uma única iteração com cada uma das 40 instâncias de OR-Beasley.

A Figura 2 mostra os resultados do experimento com a execução de 50000 iterações da GRASP sobre cada uma das mesmas instâncias. Outras avaliações foram feitas com número de iterações abaixo e acima de 50000 iterações.

Nas Figuras 1 e 2 a barra mais à esquerda nos gráficos - relativa a cada instância - mostra o desempenho da GRASP. A Tabela 2 mostra os resultados numéricos, obtidos dos experimentos com uma única iteração e com 50000 iterações.

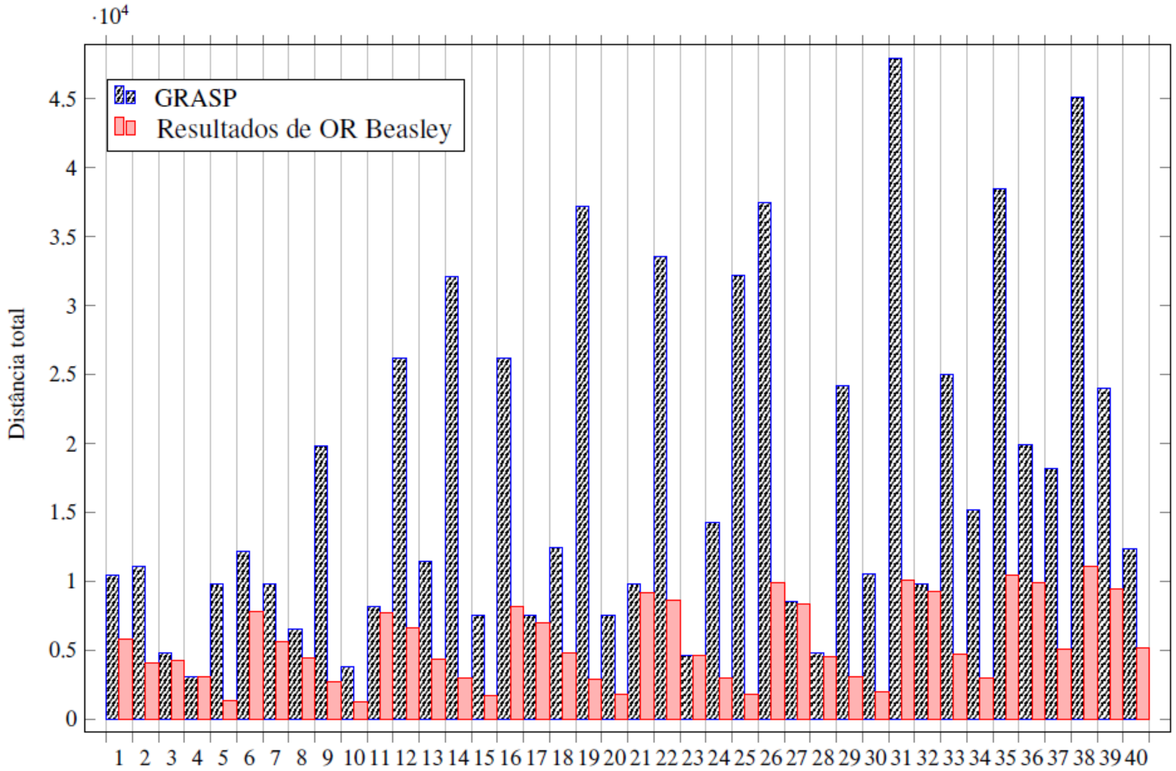

FIGURA 1 – Experimento: comparação do valor distância total com Beasley – uma única iteração GRASP

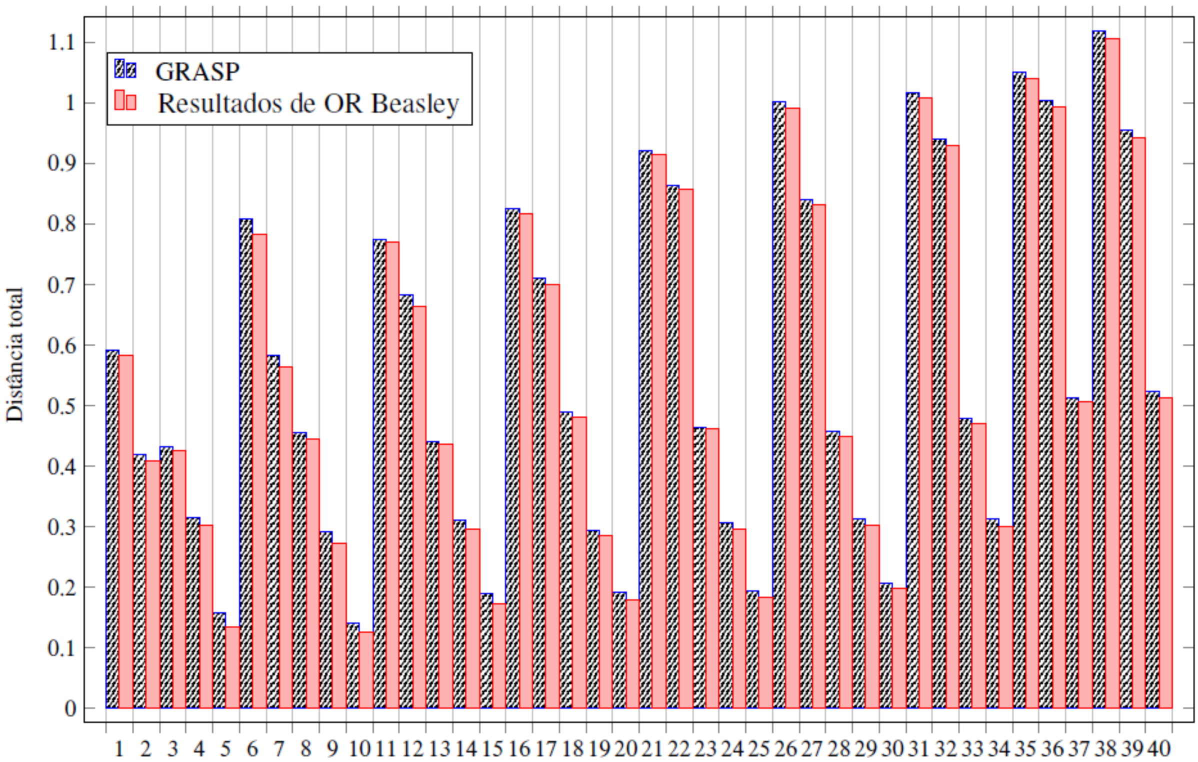

FIGURA 2 – Experimento: comparação do valor distância total com Beasley – 50000 iterações GRASP

TABELA 2 – Resultados do experimento para avaliar a GRASP com instâncias da biblioteca OR-Beasley

Instância	Número de Cidades	Número Máximo de Medianas	Trocas Para Uma Melhor Solução, em 10000 Iterações GRASP	Distância Total Para Uma Única Iteração GRASP	Distância Total Para 50000 Iterações GRASP	Tempo Médio Por Iteração GRASP (seg)	Solução Ótima OR Beasley
1	100	10	16	10475	5854	0.025	5819
2	100	10	12	11075	4093	0.036999999	4093
3	100	20	11	4750	4320	0.059999999	4250
4	100	33	18	3098	3064	0.11	3034
5	100	5	12	9795	1355	0.032	1355
6	200	10	5	12175	7827	0.07	7824
7	200	20	13	9812	5698	0.149999999	5631
8	200	40	6	6511	4505	0.31	4445
9	200	5	9	19795	2734	0.047	2734
10	200	67	7	3816	1306	0.41999999	1255
11	300	100	9	8169	7743	1.02	7696
12	300	10	10	26210	6684	0.13999999	6634
13	300	30	15	11478	4416	0.36	4374
14	300	5	12	32103	2968	0.16	2968
15	300	60	5	7512	1807	0.58	1729
16	400	10	14	26153	8198	0.27999999	8162
17	400	133	5	7565	7114	1.68999999	6999
18	400	40	10	12143	4892	0.61	4809
19	400	5	18	37145	2845	0.14999999	2845
20	400	80	9	7515	1912	1.14000004	1789
21	500	100	4	9800	9200	1.69	9138
22	500	10	15	33495	8627	0.35	8579
23	500	167	2	4650	4648	2.79999999	4619
24	500	50	12	14295	3078	1.01	2961
25	500	5	11	31234	1828	0.32	1828
26	600	10	15	37412	10023	0.3999999	9917
27	600	120	3	8497	8395	2.71	8307
28	600	200	4	4759	4579	3.0000009	4498
29	600	5	12	24152	3100	0.41	3033
30	600	60	9	10495	2075	1.43999999	1989
31	700	10	19	47192	10154	0.62999999	10086
32	700	140	7	9783	9389	3.50999998	9297
33	700	5	10	25015	4758	0.54	4700
34	700	70	11	15145	3122	1.87999997	3013
35	800	10	15	38416	10489	0.82	10400
36	800	5	9	19917	10015	0.63999999	9934
37	800	80	14	18194	5126	2.55	5057
38	900	10	16	45123	11151	1.03	11060
39	900	5	9	24022	9473	0.86999998	9423
40	900	90	12	12331	5245	2.31999997	5128

8. Considerações Finais

A metaheurística GRASP duplamente reativa descrita neste trabalho obteve bons resultados para as 40 instâncias de problemas testes OR-Beasley.

Como eram de se esperar, os resultados com uma única iteração GRASP ficaram piores em relação ao da solução ótima dos problemas, afinal, a primeira solução fornecida por esssa metaheurística é aleatória. Mas houve uma melhora significativa nos resultados com 50 mil iterações, sendo que em 6 instâncias, a GRASP encontrou a solução ótima do problema.

Nas outras 34 instâncias, os resultados obtidos ficaram bem próximos da solução ótima OR-Beasley.

Como metaheurísticas são procedimentos com capacidade de produzir soluções aproximadas de boa qualidade, sendo que é frágil o compromisso desses procedimentos em alcançar o ótimo local, pode-se considerar que os resultados obtidos pela GRASP duplamente reativa neste trabalho são plenamente satisfatórios.

Outra característica positiva foi o tempo médio de cada iteração. Em poucas instâncias o tempo médio ficou acima de 1 segundo, confirmando, assim, que o tempo médio por iteração foi bom.

Com o aumento de escala dos problemas de localização, as metaheurísticas ganharam um espaço generoso junto aos métodos de solução de problemas combinatórios. O crescimento de porte dos casos práticos que surgem na economia, continua a exigir esforço de pesquisa para aumentar a capacidade e eficiência dos métodos de solução. O que se procurou aqui foi caminhar nessa direção.

Referências

AIEX, R. M.; BINATO, S. e RESENDE, M. G. G. Parallel grasp with path-relinking for job shop scheduling. *Parallel Computing - Special issue: Parallel computing in numerical optimization,* 29(4), 393 – 430, 2003.

ALVIM, A.; RIBEIRO, C. *Load balancing in the parallelization of the metaheuristic GRASP*. Technical report, Department of Computer Science, Catholic University of Rio de Janeiro, Rio de Janeiro, RJ 22453-900 Brazil.1998.

BATTITI, R. *Reactive search: towards self-tuning heuristics*. In V. J. Rayward- Smith, I. H. Osman, C. R. Reeves, G. D. Smith, editors, Modern Heuristic Search Methods. 1996.

BEASLEY, J. E. OR – library: distribution test problems by electronic mail. *Journal of Operational Research Society*, 41, 1069–1072, 1990.

BERMAN, O.; DREZNER, Z. A note on the location of an obnoxious facility on a network. *European Journal of Operational Research*, (120), 215–217, 2000.

BOUDIA, M.; LOULY, M.A.O., e PRINS, C. A reactive GRASP and path-relinking for a combined production–distribution problem. *Computers & Operations Research* 34 , 3402 – 3419, 2007.

DELMAIRE, H.; DÍAZ, J.; FERNÁNDEZ, E. e ORTEGA, M. Reactive grasp and tabu search based heuristics for the single source capacitated plant location problem. *Information Systems and Operational Research. Special Issue: Metaheuristics for location and routing problems*, 37(3), 194–225, 1999.

FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6, 109 – 133, 1995.

FESTA, P.; RESENDE, M. G. C. An annotated bibliography of grasp-part II: Applications. *International transactions in Operational Research*, 16, 131–172, 2009.

FRINHANI, R. M. D. *Grasp com path-relinking para agrupamento de dados biológicos*. (Dissertação de Mestrado) - Instituto de Ciências Exatas - Universidade Federal de Minas Gerais, Belo Horizonte, 2011.

GARCÍA, J. A. D. *Algorithmic approaches for the single source capacitated plant location problem.* (Tese de Doutorado) - Dept. d'Estatística i Investigación Operativa - Universitat Politécnica de Catalunya, Barcelona, 2001.

MARTINS, S.; RESENDE, M.; RIBEIRO, C. e PARDALOS, P. A parallel GRASP for the Steiner tree problem in graphs using a hybrid local search strategy. *Journal of Global Optimization*, 17, 267–283, 2000.

RESENDE, M. G. C.; WERNECK, R. F. A grasp with path-relinking for the p-median problem. *AT&T Labs Research Technical Report* TD-5E53XL, 2002.