

GRASP DUPLAMENTE REATIVA COM *PATH-RELINKING* PARA O PROBLEMA DE *P*-MEDIANAS APLICADO À FORMAÇÃO DE CLUSTERS EM LOGÍSTICA DE LOCALIZAÇÃO

DOUBLY REACTIVE GRASP WITH *PATH-RELINKING* TO *P*-MEDIAN PROBLEM APPLIED TO CLUSTERS FORMATION IN LOCATION LOGISTICS

Caroline Nascimento Parajara* E-mail: carolparajara@hotmail.com

Geraldo Galdino de Paula Junior* E-mail: galdino@uenf.br

*Universidade Estadual do Norte Fluminense (UENF), Campos dos Goytacazes, RJ, Brasil

Resumo: Este trabalho relata a experiência de formular, construir e testar um procedimento duplamente reativo para a meta-heurística GRASP com *path-relinking* destinado à solução do problema generalizado de *p*-medianas aplicado à logística de localização. 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. Para implementação do *path-relinking*, um conjunto elite de soluções foi criado através de um conceito denominado "índice vocação mediana" que estabelece o quanto cada cidade está próxima das demais cidades do grupo. Para testar a qualidade das soluções, foi utilizado um problema envolvendo as cidades brasileiras de médio e grande porte.

Palavras-chave: Problema de localização. GRASP com *path-relinking* e índice vocação mediana.

Abstract: This paper reports the experience in formulating, building and testing a procedure for a doubly reactive GRASP with *path-relinking* for the solution of the general problem of *p*-median applied to location logistic. Two reaction parameters were used to control the search for solutions in the construction phase of the GRASP. The simultaneous use of two reactive parameters allowed creating a disciplined allocation of clients to groups of medians: customers closer are allocated first. To implement the *path-relinking*, a set of elite solutions was created through a concept called "median vocation index" that establishes how much each city is close to the other cities of the group. To test the quality of solutions, a problem involving medium and large Brazilian cities was employed.

Keywords: Location problem. GRASP with *path-relinking* and median vocation index.

1 INTRODUÇÃO

Localização de instalações tem sido assunto de interesse da Engenharia de Produção, Engenharia de Transporte, de administradores, de executivos ligados à logística, etc. O problema pode ser abordado de forma qualitativa ou quantitativa, ou ainda adotando uma abordagem híbrida dos dois enfoques.

Problemas de localização associam-se a um conjunto de instalações que minimizam o custo de satisfazer a demandas e obedecem a um elenco de restrições.

Modelos e decisões locacionais são de natureza econômica e visam organizar o atendimento eficiente a algum tipo de demanda. Modelos locacionais são usados em aplicações que têm como meta e ênfase otimizar o custo ou tempo de atendimento, entre outros itens de interesse.

Especializando o tema para a indústria e a distribuição de seus produtos, o que se procura determinar é “onde” tais facilidades devem ser localizadas, “quais” fornecedores devem ser utilizados e com “quantos” centros de distribuição a empresa deve operar. Estudos que antecedem e sucedem os projetos locacionais procuram determinar o tipo de cliente ou de zonas de mercado que devem ser supridos a partir de cada centro de distribuição.

Em geral, não se pode garantir que problemas de localização sejam resolvidos de maneira exata por algoritmos em tempo polinomial determinístico. Portanto, suas soluções podem ficar comprometidas quanto a velocidade de emissão dos resultados. Este fato motiva a utilização de algoritmos heurísticos para proverem soluções viáveis tão boas quanto possíveis, eventualmente ótimas, para os problemas.

A meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedure*) (ver em Feo & Resende (1995)) vem produzindo excelentes resultados na solução de problemas combinatórios importantes. Variações da GRASP como RGRASP (GRASP REATIVA), conforme García (2001), e GRASP com religamento de caminhos (*path-relinking*), de acordo com Resende & Werneck (2002a), Boudia et al. (2007), Caires et al. (2010) e Frinhani (2011) têm produzido resultados interessantes neste contexto.

2 GRASP

Proposta por Feo & Resende (1995), a meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedure*) é um método iterativo que consiste de duas fases: uma fase de construção, na qual uma solução gulosa e aleatória é gerada, elemento a elemento; e uma fase de busca local, na qual um ótimo local na vizinhança da solução construída é pesquisado. A melhor solução encontrada ao longo de todas as iterações realizadas é retornada como resultado da GRASP.

A seguir, a GRASP é apresentada através do Algoritmo 1.

Algoritmo 1 GRASP

Requer: $GRASP (f (\square), g (\square), N (\square), GRASP_{\max}, s)$

1. $f^* \leftarrow \infty$
2. $s^* \leftarrow s$
3. **Para** (iter = 1, ... , $GRASP_{\max}$) **faça**
4. Construção ($g (\square), \alpha , s$)
5. Busca Local ($f (\square), N (\square), s$)
6. **Se** ($f(s) < f^*$) **então**
7. $s^* \leftarrow s$
8. $f^* \leftarrow f(s)$
9. **Fim Se**
10. **Fim Para**
11. $s \leftarrow s^*$
12. Retornar s
13. **Fim GRASP**

2.1 Fase de construção da GRASP

A fase de construção da GRASP começa com uma solução viável sendo construída iterativamente, elemento por elemento. A cada iteração, os próximos elementos a serem incluídos na solução são colocados em uma lista C de candidatos seguindo um critério de ordenação pré-determinado. A seleção é baseada em uma função gulosa $g : C \rightarrow \mathfrak{R}$, que mede o benefício da seleção de cada candidato a compor a solução. A componente probabilística da GRASP vem do fato de que cada elemento é selecionado de forma aleatória a partir de uma lista que contém os melhores elementos que compõem a lista de candidatos, chamada lista restrita de candidatos (LRC). Esta técnica de escolha permite que em cada iteração diferentes soluções sejam geradas.

O pseudocódigo da Fase de Construção é apresentado pelo algoritmo 2.

O tamanho da LRC é controlado por um parâmetro α , onde $0 \leq \alpha \leq 1$. O α pode ser, então, considerado como o principal parâmetro da GRASP, pois a qualidade das soluções depende de seu ajuste

Algoritmo 2 Construção**Requer:** GRASP-CONSTRUÇÃO ($g(\square)$, α , s)

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

2.2 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, $i \leq p$, é definida por

$$g_i(C_i) = g_{\min}^i + \alpha(g_{\max}^i - g_{\min}^i) \quad (1)$$

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

$$g_{\max}^i(C_i) = \frac{\sum_{j \in C_i} \Delta_{ij}}{|C_i| - 1}, \quad |C_i| > 1 \quad (2)$$

ou

$$g_{\max}^i(C_i) = \frac{f_i + \sum_{j \in C_i} \Delta_{ij}}{|C_i| - 1}, \quad |C_i| > 1 \quad (3)$$

Tanto na função (1) quanto nas funções (2) e (3), 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 (3), 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 (1) e (2), o termo -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.

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

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

	1	2	3	4	5	6	7	↓ 8	9	10
1	0	*	*	*	*	*	*	*	*	*
2	*	0	*	*	*	*	*	*	*	*
3	*	*	0	*	*	*	*	*	*	*
4	*	*	*	0	*	*	*	*	*	*
5	*	*	*	*	0	*	*	*	*	*
6	*	*	*	*	*	0	*	*	*	*
7	*	*	*	*	*	*	0	*	*	*
→ 8	12	18	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

Escolha dos clientes da mediana candidata 8, com $\alpha = 0,7$. Em cada linha

usa-se $\frac{\sum_{j \in C_i} \Delta_{ij}}{|C_i|-1}$.

- $\frac{1}{1} = 1 < 7$, escolhe o 4;
- $\frac{1+3}{2} = 2 < 7$, escolhe o 9;

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

$$g^8 = g_{\min}^8 + 0,3 \times (g_{\max}^8 - g_{\min}^8) = 0,3 \times 10 = 3.$$

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

Aqui, 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 da GRASP trabalhavam com um α , $0 \leq \alpha \leq 1$, que era calibrado até se tornar um parâmetro fixo do método. Com a evolução da pesquisa, foram adotados esquemas reativos em que é permitido ao α sofrer variações durante as sucessivas iterações do algoritmo, buscando um valor ideal para ele.

2.2 GRASP Reativa

Segundo García (2001), uma das possíveis desvantagens dos algoritmos baseados em meta-heurísticas é a falta de robustez, no sentido de que sua eficácia pode ser muito dependente da escolha de valores dos parâmetros. Isso geralmente

implica em uma tarefa considerável para “ajustar” esses valores, a fim de obter maior desempenho. Isto levou ao estudo dos chamados métodos reativos, que, em essência, são algoritmos baseados em meta-heurísticas de auto-ajuste.

2.2.1 Novo procedimento reativo para a GRASP - a metáfora da peneira

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

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 granuloso 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.

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

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

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 GRASP₂ REATIVA, em que a reação se dá, simultaneamente, pelos dois procedimentos descritos.

2.3 A fase de melhoria ou busca local da GRASP

Existe uma variedade de procedimentos possíveis para conduzir esta fase da meta-heurística. Alguns desses esquemas podem ser vistos em García (2001). Na fase de 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 hipotética 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. 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.

Segundo Rangel et al. (2000), a fase de melhoria é necessária pelo fato das soluções iniciais da GRASP não serem necessariamente ótimos locais. O objetivo da fase de melhoria, então, é melhorar a solução criada na fase construtiva. Essa busca por melhores soluções se dá consultando a estrutura de vizinhança, *Viz(s)*, da solução *s* construída, através de sucessivas trocas da solução corrente, sempre que uma melhor solução é encontrada. Várias são as trocas: pode-se fechar uma planta e alocar seus clientes à outras plantas; pode-se fechar uma planta e abrir uma que estava fechada; mudar um cliente de uma planta para outra; trocar de planta dois clientes, etc.

A seguir, o algoritmo 3 apresenta o pseudocódigo da fase de melhoria ou busca local.

Algoritmo 3 Melhoria

Executa: GRASP-MELHORIA ($f(\square)$, $N(\square)$, s)

1. $s^* \leftarrow s$ { s^* é a melhor solução encontrada até o momento }
2. $V = \{s' \in N(s) : f(s') < f(s)\}$
3. **enquanto** $|V| > 0$ **faça**
4. Selecionar $s \in V$
5. **se** $f(s) < f(s^*)$ **então**
6. $s^* \leftarrow s$
7. $V = \{s' \in N(s) : f(s') < f(s)\}$
8. **Fim se**
9. $s \leftarrow s^*$
10. **Fim enquanto**
11. Retornar s
13. **Fim** GRASP-MELHORIA

2.4 GRASP modificada: inclusão dos parâmetros de reação

Os algoritmos 4 e 5, que são 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 $\{0,0.1,\dots,0.9\}$, no decorrer das iterações. Do mesmo modo, $0 \leq \gamma < 1$ assume os valores de $\{0,0.1,\dots,0.9\}$.

3 PATH-RELINKING

O *path-relinking* foi sugerido como enfoque para integrar estratégias de intensificação e diversificação num esquema de busca. O enfoque pode ser visto como uma instância – altamente focada – de uma estratégia que procura incorporar atributos de soluções de alta qualidade (soluções elite), procurando contemplar ou favorecer esses atributos nos movimentos selecionados (GLOVER et al., 2003, *apud* GLOVER & KOCHENBERGER, 2003).

Algoritmo 4 GRASP

Executa: GRASP ($f(\square)$, $g(\square)$, $N(\square)$, $GRASP_{\max}$, s)

1. $f^* \leftarrow \infty$
2. $s^* \leftarrow s$
3. **para** (iter = 1, ..., $GRASP_{\max}$) **faça**
4. Construção ($g(\square)$, $0 \leq \alpha \leq 1$, $0 \leq \gamma < 1$, s)
5. Busca Local ($f(\square)$, $N(\square)$, s)
6. **se** ($f(s) < f^*$) **então**
7. $s^* \leftarrow s$
8. $f^* \leftarrow f(s)$
9. **fim se**
10. **fim para**
11. $s \leftarrow s^*$
12. Retornar s
13. **fim** GRASP

Algoritmo 5 Construção

Executa: GRASP-CONSTRUÇÃO ($g(\square)$, $0 \leq \alpha \leq 1$, $0 \leq \gamma < 1$, s)

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

Um procedimento equivalente seria construir caminhos usando uma combinação de várias soluções até que todas fossem examinadas.

O *path-relinking* se associa à ideia de reinterpretar as combinações lineares de pontos no Espaço Euclidiano como caminhos entre e além das soluções no âmbito da vizinhança. O caminho entre duas soluções no espaço de busca pode produzir soluções que compartilham atributos comuns a soluções previamente existentes. A sequência de soluções vizinhas no espaço de decisão é gerada a partir da solução inicial, indo na direção da solução alvo. A melhor solução encontrada na sequência é retornada.

Intensificação e diversificação em qualquer método de busca devem ser tratadas com a ideia de se manter um equilíbrio entre elas durante o processo. A

proposta do *path-relinking* foi feita como um instrumento capaz de garantir, ou pelo menos permitir a integração equilibrada dessas estratégias de intensificação e diversificação.

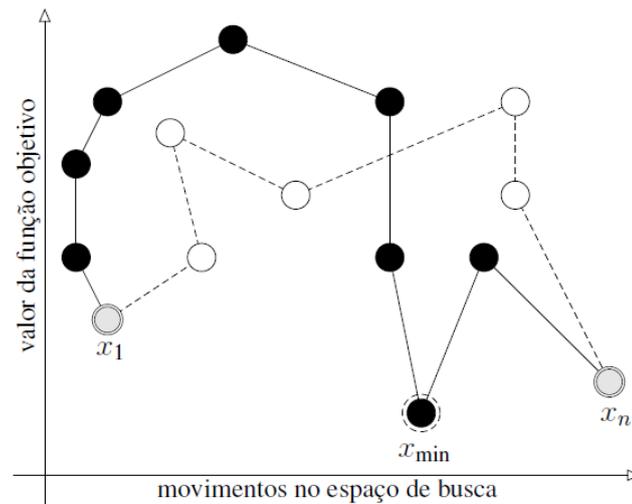
O *path-relinking* é um enfoque que pode ser usado na fase terminal de uma meta-heurística “quase como uma pós-otimização, ou pós-melhoria no caso da GRASP” para gerar novas soluções através do exame de trajetórias que ligam soluções de alta qualidade – novas ou já existentes – a partir da solução inicial.

Desse ponto inicial em diante o procedimento explora o espaço de busca gerando um caminho que leva a outras soluções, chamadas de soluções guia. A seleção dos movimentos nesta fase é feita explorando os atributos de soluções já conhecidas, em particular, aqueles das soluções guia. Enfim, o *path-relinking* pode ser visto como uma estratégia que atua no sentido de incorporar atributos de soluções de alta qualidade, tentando privilegiar esses atributos nos movimentos que são progressivamente selecionados.

A Figura 1 mostra dois caminhos possíveis ligando a solução x_1 = “início” à solução x_n = “alvo”, para ilustrar o religamento entre soluções. O caminho original é representado por segmentos de linha interrompida e decorre da “operação normal” de um algoritmo que produz seu resultado na forma de uma série de movimentos que conduzem do ponto x_1 ao ponto x_n , isto é, gera a sequência (“início” = $x_1, x_2, x_3, \dots, x_n$ = “alvo”) enquanto que o caminho de religamento é representado pelos segmentos de linha sólida.

Os caminhos são gerados com base na seleção de movimentos e adotam uma trajetória em que os passos intermediários são determinados por alguma função de avaliação. A Figura 1 mostra uma solução (o nó escuro associado a x_{\min}) alcançada pelo caminho de linha sólida. Além disso, o caminho religado pode encontrar soluções possivelmente melhores do que a solução guia inicial — que, nesta ilustração, pode ser x_n — ou provê “entradas férteis” para alcançar outras soluções, de alguma forma, melhores. Em geral, as soluções guia podem ser constituídas a partir de uma seleção de soluções anteriores chamadas de “soluções elite”.

Figura 1 - Dois caminhos: o original e o *path-relinking* – em linhas sólidas



O algoritmo 6 apresenta o pseudocódigo do *path-relinking*.

Algoritmo 6 GRASP COM *PATH-RELINKING*

Executa: GRASP COM *PATH-RELINKING* ($s_k, MAXk_{PATHRELINKING},$

$X_1, \dots, X_{MAXk_{PATHRELINKING}}, Y, S_{PATHRELINKING}$)

1. // X_k para $k = 1, \dots, MAXk_{PATHRELINKING}$, contém dados originais do //problema de localização como candidatos a medianas
2. // Y acumula as soluções elite como dados para o *path-relinking*
3. $Y \leftarrow \emptyset$
4. **para** ($k = 1, \dots, MAXk_{PATHRELINKING}$) **faça**
5. // X_k recebe, a cada iteração, uma parte (porcentagem)
6. // de vértices como candidatos a medianas, até o total de vértices
7. $X_k \leftarrow \left(\frac{k}{MAXk_{PATHRELINKING}} \times \text{DadosORIGINALIS} \right)$
8. $s \leftarrow s_k$
9. GRASP(X_k)
10. $Y \leftarrow Y \cup s_k$
11. **fim para**
12. GRASP(Y)
13. $S_{PATHRELINKING} \leftarrow s$
14. Retorna $S_{PATHRELINKING}$
15. **fim** GRASP(Y)

3.1 A integração do *path-relinking* com a GRASP

Considera-se a seguinte meta-heurística híbrida que integra o conceito de *path-relinking* com a GRASP. A GRASP mantém um conjunto de soluções elite encontradas durante as iterações da meta-heurística. A cada iteração da GRASP, o *path-relinking* é aplicado, a partir da nova solução gerada pela GRASP, a uma dada solução dentre as soluções elite. A solução alvo pode ser escolhida aleatoriamente a partir do conjunto elite. Como experiência advinda da literatura, pode-se afirmar que os papéis da solução inicial e da solução guia são intercambiáveis. Cada solução pode ser induzida a possibilitar movimentos simultâneos na direção da outra como forma de gerar combinações. Mais detalhes em Laguna & Martí (1999) e Talbi (2009).

3.2 O enfoque de *path-relinking* sugerido e adotado neste trabalho

O procedimento de *path-relinking* construído para este trabalho será descrito a seguir com destaque para novidade que ele introduz em função das características da GRASP e do problema de localização em que ele foi aplicado. Em sequência, uma aplicação foi feita através do ato de simular a formação de *clusters* com as médias e grandes cidades brasileiras, produzindo um ensaio sobre como utilizar esses recursos na solução de problemas de localização de p -medianas.

A primeira providência foi criar um conjunto de soluções elite através de um conceito que aqui foi denominado de “vocação mediana”. Este conceito, que será detalhado no decorrer desta descrição, fundamenta-se em um índice — “índice_{vm}” — que estabelece o quanto cada cidade, suposta candidata a mediana, está próxima das demais cidades do grupo. Antes de construir esse índice, o instrumento previamente utilizado foi soma das distâncias de cada cidade às demais. Posteriormente, as cidades, supostas candidatas a mediana, foram ordenadas, em ordem crescente, de acordo com a distância total de cada uma delas às demais. Vários problemas menores foram construídos e resolvidos, considerando somente uma porcentagem do total de candidatas a medianas. Diga-se, $\beta\% \times (\text{Total de candidatas a mediana})$, para o primeiro problema, $2\beta\% \times (\text{Total de candidatas a mediana para o segundo problema})$, e assim por diante, até que $k\beta\%$ se tornasse

igual a 100%, isto é, 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. Esse aglomerado de medianas foi aceito, empiricamente, como um conjunto de medianas associado às soluções elite do problema de localização. Tal aceitação empírica da condição “elite” das soluções assim obtidas, ficou aguardando que experimentos comprovassem a presença de tal atributo. O passo final foi aplicar a esse conjunto de soluções elite um procedimento de *path-relinking* que explorou a intensificação e a diversificação do processo de busca. Ou seja, um conjunto de novas soluções foi gerado nas várias iterações da GRASP para produzir a solução final que se desejava melhor do que as soluções encontradas antes da intensificação do *path-relinking*.

4 O CÓDIGO COMPUTACIONAL

O código computacional foi feito em C++. Foram usados alguns componentes da biblioteca STL, o que permitiu desenvolver um código estável, com alto grau de generalidade e baseado em estruturas de dados bastante testadas.

O código foi projetado e construído com base em uma classe de membros, dados e funções, que permitiram modular as várias operações envolvidas na solução do problema. As informações que alimentavam o código foram constituídas do número $|N|=n$ de pontos, cidades ou vértices envolvidos e que formam o conjunto N . Além desse número, os dados continham a matriz $n \times n$ simétrica de distâncias, e o número máximo de medianas que a instância pretendia localizar. Um determinado número de ligações compunha algumas entradas da matriz de distâncias. Posteriormente, as demais entradas foram calculadas usando uma rotina que implementa o algoritmo — de programação dinâmica, detalhes em Nemhauser (1966) — de Floyd-Warshall, para encontrar as distâncias mínimas entre todos os pares de pontos ou vértices que se associam às entradas da matriz.

5 RESULTADOS

Para mostrar o funcionamento deste modelo de *path-relinking* foi feita uma aplicação do problema generalizado de *p*-medianas, o que passa a ser feito através da formação de *clusters* com as cidades brasileiras, médias e grandes, que aparecem na matriz de distâncias do DNIT.

A Tabela 2 mostra as 112 cidades brasileiras que participaram do experimento ordenadas pelo índice de vacação mediana (índice_{vm}).

Tabela 2 - Índice de vocação mediana das 112 cidades (continua)

Cidade	Distância		Cidade	Distância	
	Total	Índice _{vm}		Total	Índice _{vm}
Ribeirão Preto	147471	1,328	Barbacena	342656	3,086
Campinas	150832	1,358	Sobral	342933	3,089
Uberlândia	150823	1,358	Caldas Novas	343551	3,095
São Paulo	151522	1,365	Petrópolis	347597	3,313
Belo Horizonte	153667	1,384	Anápolis	349736	3,150
S. José dos Campos	154141	1,388	Sta. Maria da Vitória	350263	3,155
Goiânia	157662	1,419	Volta Redonda	351591	3,167
Brasília	158872	1,413	Itumbiara	359269	3,236
Santos	159368	1,435	Franca	359849	3,241
Juiz de Fora	161184	1,452	Guaratinguetá	360363	3,246
Londrina	164132	1,478	Poços de Caldas	361194	3,254
Rio de Janeiro	167323	1,507	Uberaba	361764	3,259
Curitiba	169685	1,528	Parnaíba	362049	3,261
Vitória	180256	1,623	Barretos	362978	3,270
Joinville	181321	1,633	Taubaté	363215	3,272
Campo Grande	190504	1,716	Limeira	368502	3,319
Florianópolis	195533	1,761	Araraquara	370828	3,340
Feira de Santana	207691	1,871	Porto Velho	371620	3,347
Caxias do Sul	217509	1,959	Rio Verde	371832	3,349
Salvador	217617	1,960	São Carlos	372180	3,352
Palmas	218699	1,970	Americana	373571	3,365
Cuiabá	221652	1,996	Sorocaba	378166	3,406
Porto Alegre	228249	2,056	Imperatriz	379122	3,415
Aracaju	232642	2,095	S. José do Rio Preto	379612	3,419
Pelotas	254483	2,292	Bauru	382036	3,441
Paulo Afonso	256007	2,306	Araçatuba	384912	3,467
Maceió	257475	2,319	Marília	388102	3,496
Guaranhuns	260734	2,348	Ourinhos	391800	3,529
Petrolina	268474	2,418	Presidente Prudente	396563	3,572
Itaparica	270544	2,437	Rondonópolis	410760	3,700
Caruaru	2701607	2,446	Paranaguá	413393	3,724
Teresina	272819	2,457	Maringá	415095	3,739
Itabuna	273897	2,467	Ponta Grossa	416291	3,750
Recife	273953	2,468	Tucuruí	419672	3,780
V. da Conquista	276382	2,489	Itajaí	429899	3,872
Valença	280867	2,530	Rio Branco	429826	3,872
João Pessoa	284830	2,566	Blumenau	434125	3,911
Campina Grande	285137	2,568	Cascavel	441668	3,978
Juazeiro do Norte	294508	2,653	Lages	445211	4,010
Porto Seguro	295473	2,661	Dourados	448614	4,041
São Luís	297497	2,680	Tubarão	452019	4,072
Picos	298725	2,691	Ponta Porã	452165	4,073
Natal	301164	2,713	Foz do Iguaçu	456646	4,113
Barreiras	301376	2,715	Criciúma	458262	4,128

Tabela 2 - Índice de vocação mediana das 112 cidades (conclusão)

Cidade	Distância Total	Índice _{vm}	Cidade	Distância Total	Índice _{vm}
Ilhéus	305398	2,751	Passo Fundo	464060	4,180
Fortaleza	308267	2,777	Corumbá	468708	4,222
Gov. Valadares	308359	2,778	Manaus	470779	4,241
Belém	311650	2,807	Novo Hamburgo	475959	4,287
Montes Claros	312953	2,819	Sinop	486850	4,386
Ipatinga	314507	2,833	Santa Maria	494601	4,445
Mossoró	317272	2,858	Vilhena	515323	4,642
Itabira	320016	2,883	Bagé	520224	4,686
Cachoeiro de Itap.	333516	3,004	Uruguaiana	524785	4,727
Patos de Minas	334335	3,012	S. do Livramento	531354	4,786
São Lourenço	339087	3,054	Chuí	535793	4,826
Campos dos Goyt.	341316	3,074	Boa Vista	556965	5,017

Pela Tabela 2, vê-se que a cidade brasileira mais próxima das demais — cidades médias e grandes — é Ribeirão Preto-SP, com índice_{vm} igual a 1,328. A mais distante das demais é Boa Vista-RR, com índice_{vm} igual a 5,017.

O índice_{vm} foi construído de acordo com

$$\text{índice}_{vm}(\text{"cidade"}) = \frac{\sum (\text{distâncias da "cidade" às outras "cidades"})}{\text{Número total das cidades} \times 1000}$$

Por exemplo,

$$\text{índice}_{vm}(\text{"Manaus"}) = \frac{\sum (\text{distâncias de Manaus às outras "cidades"})}{\text{Número total das cidades} \times 1000} = 4.241$$

$$\text{índice}_{vm}(\text{"São Paulo"}) = \frac{\sum (\text{distâncias de São Paulo às outras "cidades"})}{\text{Número total das cidades} \times 1000} = 1.365$$

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 resolvidas 9 instâncias menores mais a instância original.

A Tabela 3 mostra um conjunto de 26 cidades candidatas a mediana, definido durante a solução das 10 instâncias, conforme já descrito. A Tabela 4 mostra um resumo das soluções de cada uma das 10 instâncias com 2000 iterações. A melhor solução sem o *path-relinking* ocorreu na quinta instância, ou seja, 20% das cidades candidatas a mediana, com distância total igual 68551, entre as medianas e as

idades que a elas se ligam. O resultado com *path-relinking* — Tabela 6 — teve distância total igual 62393. A diferença entre os dois resultados se torna expressiva se for considerado que os dois passaram por um processo de “otimização no sentido das meta-heurísticas”. Em outras palavras, o *path-relinking* trabalha sobre um resultado já otimizado, ou o tem como referência.

Tabela 3 - 26 cidades elite, candidatas a medianas na fase de *path-relinking*

Aracajú-SE	Campinas-SP	Campos Grande-MS	Caxias do Sul-RS
Cuiabá-MT	Curitiba-PR	Feira de Santana-BA	Florianópolis-SC
Fortaleza-CE	Goiânia-Go	João Pessoa-PB	Juiz de Fora-MG
Londrina-PR	Maceió-Al	Natal-RN	Pelotas-RS
Porto Alegre-RS	Porto Velho-RO	Ribeirão Preto-SP	Rondonópolis-MT
Salvador-BA	Santos-SP	S. J. dos Campos-SP	São Paulo-SP
Teresina-PI	Uberlândia-MG		

Tabela 4 - Resumo dos resultados com 112 cidades, 5 medianas, 2000 iterações

Instância	Trocas para melhor solução	Distância total	Tempo médio por iteração(seg)
1	8	108493	0,03250000
2	9	68551	0,06450000
3	3	68613	0,09950000
4	8	68899	0,13349999
5	7	70795	0,16850001
6	4	71048	0,20299999
7	5	71546	0,24050000
8	2	73940	0,28650001
9	8	72485	0,33600000
10	4	75445	0,38350001

Experimento semelhante, agora com 20000 iterações para resolver os problemas menores, tem o resultado resumido na Tabela 5. Neste caso, a melhor solução sem o *path-relinking* é a distância total igual a 66086. Com o *path-relinking* o valor melhora para 62393. Os valores da distância total sem o *path-relinking* são diferentes, mostrando que um número maior de iterações influenciou o resultado. Entretanto, a distância total com *path-relinking* foi o mesmo nos dois casos apresentados. O caso de 20000 iterações para resolver cada um dos problemas menores, gerou uma elite de 22 cidades, diferente do caso anterior, de 2000 iterações, que gerou uma elite de 26 cidades. Nos dois casos as cidades elites foram candidatas a mediana na fase de intensificação do *path-relinking*.

Com fator de intensificação do *path-relinking* igual a 10 a solução foi a mesma, e a única diferença foi a ordem em que o algoritmo definiu as medianas com

os *clusters* associados, de acordo com a Tabela 7. Suspeita-se que esta seja uma solução ótima para este número de medianas.

Tabela 5 - Resumo dos resultados com 112 cidades, 5 medianas, 20000 iterações

Instância	Trocas para melhor solução	Distância total	Tempo médio por iteração(seg)
1	3	108493	0,03335000
2	6	68551	0,06450000
3	9	66362	0.10200000
4	6	66086	0,14135000
5	13	68300	0,18560000
6	10	68631	0,22485000
7	19	67489	0,27375001
8	8	67329	0,37114999
9	14	70217	0,41319999
10	13	69277	0,47815001

Tabela 6 - Resultado com *path-relinking*

(continua)

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 62393

1) *Cluster* da mediana FEIRA DE SANTANA-BA:

ARACAJÚ-SE	BARREIRAS-BA	CAMPINA GRANDE-PB	ITAPARICA-PE
GUARANHUNS-PE	ILHÉUS-BA	ITABUNA-BA	PAULO-AFONSO-BA
JOÃO PESSOA-PB	MACEIÓ-AL	NATAL-RN	SALVADOR-BA
PETROLINA-PE	PORTO SEGURO-BA	RECIFE-PE	
S. Mª DA VITÓRIA-BA	VALENÇA-BA	V. DA CONQUISTA-BA	

Total de cidades neste *cluster*: 19. Distância (custo): 10557

2) *Cluster* da mediana RIBEIRÃO PRETO-SP:

AMERICANA-SP	ANÁPOLIS-GO	ARAÇATUBA-SP	ARARAQUARA-SP
BARRETOS-SP	BARBACENA-MG	BAURU-SP	B. HORIZONTE-MG
BRASÍLIA-DF	C. DE ITAPEMIRIM-ES	CALDAS NOVAS-GO	CAMPINAS-SP
CAMPO GRANDE-MS	CAMPOS DOS GOYT-RJ	CASCADEL-PR	CORUMBÁ-MS
CUIABÁ-MT	CURITIBA-PR	DOURADOS-MS	F. DO IGUAÇU-PR
FRANCA-SP	COIÂNIA-GO	GOV. VALADARES-MG	GUARATINGUETÁ-SP
IPATINGA-MG	ITABIRA-MG	ITUMBIARA-GO	JUIZ DE FORA-MG
LIMEIRA-SP	LONDRINA-PR	MARÍLIA-SP	MARINGÁ-PR
MONTES CLAROS-MG	OURINHOS-SP	PARANAGUÁ-PR	PATOS DE MINAS-MG
PETRÓPOLIS-RJ	POÇOS DE C.-MG	PONTA GROSSA-PR	PONTA PORÃ-MS
PRES. PRUDENTE-SP	RIO DE JANEIRO-RJ	RIO VERDE-GO	RONDONÓPOLIS-MT
SANTOS-SP	SÃO CARLOS-SP	S. J. DO RIO PRETO-SP	S. J. DOS CAMPOS-SP
SÃO LOURENÇO-MG	SÃO PAULO-SP	SINOP-MT	SOROCABA-SP
TAUBATÉ-SP	UBERABA-MG	UBERLÂNDIA-MG	VITÓRIA-ES

Total de cidades neste *cluster*: 57. Distância (custo) total neste *cluster*: 33771

3) *Cluster* da mediana PORTO VELHO-RO:

BOA VISTA-RR	MANAUS-AM	RIO BRANCO-AC	VILHENA-RO
--------------	-----------	---------------	------------

Total de cidades neste *cluster*: 4. Distância (custo) total neste *cluster*: 3835

4) *Cluster* da mediana PORTO ALEGRE-RS:

BAGÉ-RS	BLUMENAU-SC	CAXIAS DO SUL-RS	CHUÍ-RS
CRICIÚMA-SC	FLORIANÓPOLIS-SC	ITAJAÍ-SC	JOINVILLE-SC

Tabela 6 - Resultado com *path-relinking*

(conclusão)

DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 62393

LAGES-SC	NOVO HAMBURGO-RS	PASSO FUNDO-RS	PELOTAS-RS
SANTA MARIA-RS	S. DO LIVRAMENTO-RS	TUBARÃO-SC	URUGUAIANA-RS
Total de cidades neste <i>cluster</i> : 16. Distância (custo) total neste <i>cluster</i> : 6380			
5) <i>Cluster</i> da mediana TERESINA-PI:			
BELÉM-PA	FORTALEZA-CE	IMPERATRIZ-MA	JUAZEIRO DO N.-CE
MOSSORÓ-RN	PALMAS-TO	PARNAÍBA-PI	PICOS-PI
SÃO LUÍS-MA	SOBRAL-CE	TUCURÍ-PA	
Total de cidades neste <i>cluster</i> : 11. Distância (custo) total neste <i>cluster</i> : 7850			

Tabela 7- Resultado com *path-relinking* após 200000 iterações, e 20000 iterações na solução de cada um dos problemas menores**DISTÂNCIA TOTAL DA MELHOR SOLUÇÃO ENCONTRADA: 62393**Total de cidades neste *cluster*: 19. Distância (custo): 105571) *Cluster* da mediana RIBEIRÃO PRETO-SP:

AMERICANA-SP	ANÁPOLIS-GO	ARAÇATUBA-SP	ARARAQUARA-SP
BARRETOS-SP	BARBACENA-MG	BAURU-SP	B. HORIZONTE-MG
BRÁSILIA-DF	C. DE ITAPEMIRIM-ES	CALDAS NOVAS-GO	CAMPINAS-SP
CAMPO GRANDE-MS	CAMPOS DOS GOYT-RJ	CASCAVEL-PR	CORUMBÁ-MS
CUIABÁ-MT	CURITIBA-PR	DOURADOS-MS	F. DO IGUAÇÚ-PR
FRANCA-SP	COIÂNIA-GO	GOV. VALADARES-MG	GUARATINGUETÁ-SP
IPATINGA-MG	ITABIRA-MG	ITUMBIARA-GO	JUIZ DE FORA-MG
LIMEIRA-SP	LONDRINA-PR	MARÍLIA-SP	MARINGÁ-PR
MONTES CLAROS-MG	OURINHOS-SP	PARANAGUÁ-PR	PATOS DE MINAS-MG
PETRÓPOLIS-RJ	POÇOS DE C.-MG	PONTA GROSSA-PR	PONTA PORÃ-MS
PRES. PRUDENTE-SP	RIO DE JANEIRO-RJ	RIO VERDE-GO	RONDONÓPOLIS-MT
SANTOS-SP	SÃO CARLOS-SP	S. J. DO RIO PRETO-SP	S. J. DOS CAMPOS-SP
SÃO LOURENÇO-MG	SÃO PAULO-SP	SINOP-MT	SOROCABA-SP
TAUBATÉ-SP	UBERABA-MG	UBERLÂNDIA-MG	VITÓRIA-ES
VOLTA REDONDA-RJ			
Total de cidades neste <i>cluster</i> : 57. Distância (custo) total neste <i>cluster</i> : 33771			

2) *Cluster* da mediana PORTO VELHO-RO:

BOA VISTA-RR	MANAUS-AM	RIO BRANCO-AC	VILHENA-RO
Total de cidades neste <i>cluster</i> : 4. Distância (custo) total neste <i>cluster</i> : 3835			

3) *Cluster* da mediana PORTO ALEGRE-RS:

BAGÉ-RS	BLUMENAU-SC	CAXIAS DO SUL-RS	CHUI-RS
CRICIÚMA-SC	FLORIANÓPOLIS-SC	ITAJAÍ-SC	JOINVILLE-SC
LAGES-SC	NOVO HAMBURGO-RS	PASSO FUNDO-RS	PELOTAS-RS
SANTA MARIA-RS	S. DO LIVRAMENTO-RS	TUBARÃO-SC	URUGUAIANA-RS
Total de cidades neste <i>cluster</i> : 16. Distância (custo) total neste <i>cluster</i> : 6380			

4) *Cluster* da mediana TERESINA-PI:

BELÉM-PA	FORTALEZA-CE	IMPERATRIZ-MA	JUAZEIRO DO N.-CE
MOSSORÓ-RN	PALMAS-TO	PARNAÍBA-PI	PICOS-PI
SÃO LUÍS-MA	SOBRAL-CE	TUCURÍ-PA	
Total de cidades neste <i>cluster</i> : 11. Distância (custo) total neste <i>cluster</i> : 7850			

5) *Cluster* da mediana FEIRA DE SANATANA-BA:

ARACAJÚ-SE	BARREIRAS-BA	CAMPINA GRANDE-PB	ITAPARICA-PE
GUARANHUNS-PE	ILHÉUS-BA	ITABUNA-BA	PAULO-AFONSO-BA
JOÃO PESSOA-PB	MACEIÓ-AL	NATAL-RN	SALVADOR-BA
PETROLINA-PE	PORTO SEGURO-BA	RECIFE-PE	
S. Mª DA VITÓRIA-BA	VALENÇA-BA	V. DA CONQUISTA-BA	
Total de cidades neste <i>cluster</i> : 19. Distância (custo) total neste <i>cluster</i> : 10557			

Um fato que chama atenção é o de Ribeirão Preto-SP, cidade brasileira com melhor índice de vocação mediana, ser a mediana do maior *cluster*. A Tabela 8 mostra uma possível correlação entre o tamanho do *cluster* e o índice de vocação mediana da cidade em questão: na medida em que piora o índice_{vm}, diminui o tamanho do *cluster* atendido por aquela mediana. Esta possível correlação entre o tamanho do *cluster* e o índice_{vm} não foi estudada neste trabalho.

Tabela 8 - Quando piora o índice de vocação mediana, diminui o tamanho do *cluster*

Mediana	Número de cidades no <i>cluster</i>	Índice vocação mediana
Ribeirão Preto-SP	57	1,328
Feira de Santana-BA	19	1,871
Porto Alegre-RS	16	2,056
Teresina-PI	11	2,454
Porto Velho-RO	4	3,347

5 CONCLUSÕES

Este trabalho se baseou na hipótese de que havia espaço para aprimoramento de métodos que resolvem problemas de otimização usando meta-heurísticas. Em particular, a escolha para demonstrar a veracidade desta hipótese recaiu sobre a meta-heurística GRASP, seus métodos reativos e procedimentos de *path-relinking*, especializados na solução do modelo generalizado de localização de p -medianas. Para alcançar o objetivo buscou-se explorar o processo reativo da GRASP por duas vias, ou seja, adotando o que se passou a denominar de “GRASP duplamente reativa”, já que dois parâmetros de reação passaram a atuar, cooperativamente, nesse cenário. Para entendimento próprio e para comunicar o procedimento à comunidade interessada no assunto, passou-se a explicar tal procedimento com uma metáfora que aqui foi denominada de “metáfora da peneira”. Todo o esforço feito confirmou a boa reputação da meta-heurística GRASP no contexto da resolução de problemas combinatórios destinados a aplicações tecnológicas do cotidiano.

A questão básica aqui tratada se constitui de como decidir, a partir de locais elegíveis conhecidos ou a partir de um grande número de locais descritos como área, uma localização para instalar uma facilidade e como atribuir ou alocar clientes a ela.

O passo seguinte foi o de explorar as capacidades do *path-relinking*. As experiências realizadas mostraram tratar-se de um procedimento capaz de

aproximar os resultados da GRASP aos de soluções dos algoritmos exatos. A principal proposta desse método é a de intensificar a busca entre soluções elite encontradas nas fases anteriores do algoritmo. De todas as formas acredita-se que não está esgotado o espaço de aprimoramentos aqui explorado, o que constitui estímulo para prosseguir testando ideias semelhantes. Além disso, tal esforço sempre se ocupa de um aspecto particular desse problema que é amplo e variado nas diversas modalidades do problema e suas aplicações.

As simulações realizadas passaram a ser conduzidas com dados reais, se enquadrando no que se pode chamar de contribuição à logística de localização. A extensão dos experimentos foi relativamente grande na medida em que se procurou variar a quantidade de *clusters*, o número de iterações, assim como os valores do parâmetro de intensificação do *path-relinking* que passou a ser usado em caráter definitivo.

Um subproduto interessante dessa fase de simulação de *clusters* foi a invenção de um índice que passou a ser chamado de “índice de vocação mediana” que é uma representação numérica capaz de medir a proximidade da cidade em questão com as demais cidades brasileiras, médias e grandes, constantes da matriz do DNIT. Esta é uma informação logística de valor tanto para um estudo locacional qualitativo, quanto para projetos de localização em que se explora o lado quantitativo. A pesquisa descobriu por via dos experimentos conduzidos que a cidade detentora do melhor índice de vocação mediana é RIBEIRÃO PRETO–SP, e a cidade com o pior índice é BOA VISTA–RR. Essa informação não exclui as cidades intermediárias em relação a este índice, já que participando de uma análise locacional elas podem apresentar outras características interessantes, tanto no aspecto qualitativo, quanto do lado quantitativo.

REFERÊNCIAS

AIEX, R. M.; BINATO, S.; RESENDE, M. G. G. Parallel GRASP with *path-relinking* for job shop scheduling. **Parallel Computing - Special issue: Parallel computing in numerical optimization**, v. 29, n.4, p. 393 – 430, 2003.
[http://dx.doi.org/10.1016/S0167-8191\(03\)00014-0](http://dx.doi.org/10.1016/S0167-8191(03)00014-0)

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, Brazil.1988.

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.
Revista Produção Online, Florianópolis, SC, v.15, n. 3, p.807-829, jul./set. 2015.

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

[http://dx.doi.org/10.1016/S0377-2217\(98\)00237-9](http://dx.doi.org/10.1016/S0377-2217(98)00237-9)

BOUDIA, M.; LOULY, M.A.O., e PRINS, C. A reactive GRASP and *path-relinking* for a combined production–distribution problem. **Computers & Operations Research**, n. 34, p. 3402 – 3419, 2007. <http://dx.doi.org/10.1016/j.cor.2006.02.005>

CAIRES, L. F. V.; ASSIS, L. P. e, ANDRADE, A. V. Aplicação do método *path-relinking* na resolução do problema roteamento de veículos com coleta e entrega simultâneas. In: XLII SBPO. 31, 38, 2010, Bento Gonçalves. **Anais...** Bento Gonçalves: SOBRAPO, 2010, p. 1745 – 1756.

FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. **Journal of Global Optimization**, 6, p. 109 – 133, 1995. <http://dx.doi.org/10.1007/BF01096763>

FRINHANI, R. M. D. **GRASP com *path-relinking* para agrupamento de dados biológicos.** Originalmente apresentada como dissertação de mestrado, Instituto de Ciências Exatas - Universidade Federal de Minas Gerais, 2011.

<http://hdl.handle.net/1843/SLSS-8GQJ46>

GARCÍA, J. A. D. **Algorithmic approaches for the single source capacitated plant location problem.** Originalmente apresentada como tese de doutorado, Universitat Politècnica de Catalunya, 2001.

GLOVER, F., LAGUNA, M., & MARTÍ, R. **Scatter search and *path-relinking*: Advances and applications.** 30 - 55, 2003.

http://dx.doi.org/10.1007/0-306-48056-5_1

GLOVER, F.; KOCHENBERGER, G. A.. **Handbook of Metaheuristics.** International series in operations research & management science. Boston, Dordrecht, London: Kluwer Academic Publishers. 55. Eds 2003.

<http://dx.doi.org/10.1007/b101874>

LAGUNA, M. & MARTÍ, R. GRASP and *path-relinking* for 2-layer straight line crossing minimization. **INFORMS J. on Computing**, v. 11, n. 1, p. 44–52. 57, 1999.

<http://pubsonline.informs.org/doi/pdf/10.1287/ijoc.11.1.44>

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, p. 267–283, 2000.

<http://dx.doi.org/10.1023/A:1026546708757>

NEMHAUSER, G. L. **Introduction to Dynamic Programming.** Series in decision and control. Wiley. 69, 1966.

RANGEL, M. C., ABREU, N. M. M., & NETTO, P. O. B. GRASP para o pqa: um limite de aceitações para soluções iniciais. **Pesquisa Operacional**, v. 20, n. 1, p. 46–58. 32, 47. 2000.

<http://dx.doi.org/10.1590/S0101-74382000000100006>

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.

<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.9495>

TALBI, E.G. **Metaheuristics: From Design to Implementation**. John Wiley & Sons, 2009.
<http://dx.doi.org/10.1002/9780470496916>



Artigo recebido em 27/06/2013 e aceito para publicação em 25/06/2015
DOI: [http://dx.doi.org/ 10.14488/1676-1901.v15i3.1628](http://dx.doi.org/10.14488/1676-1901.v15i3.1628)