

Uma Arquitetura e Ferramentas para Problemas de Localização de Facilidades no Setor Público

Gilberto F. Sousa Filho¹, Bruno Jefferson S. Pessoa², José Jorge L. Dias Jr.¹,
Lucídio dos Anjos Formiga Cabral²

¹Departamento de Ciências Exatas – Universidade Federal da Paraíba (UFPB)
Rio Tinto – PB – Brasil

²Centro de Informática – Universidade Federal da Paraíba (UFPB)
João Pessoa – PB – Brasil

gilberto@dce.ufpb.br, brunopessoa@di.ufpb.br,
jorge@dce.ufpb.br, lucidio@di.ufpb.br

Abstract. *One of the problems in the city planning is related to decision-making process about the facility location in public sector. This issue is known as Facility Location Problem in the Operational Research field. Hence, this paper presents a proposal of a software architecture for information systems in a facility location problem domain. Besides, two tools, developed from the proposed architecture, as well as some results about efficiency in its utilization are presented.*

Resumo. *Um dos problemas no planejamento urbano diz respeito ao processo decisório sobre as localizações das instalações no setor público. Esta problemática é conhecida no ramo da Pesquisa Operacional como o Problema de Localização de Facilidades. Neste sentido, este artigo apresenta uma proposta de arquitetura de software para sistemas de informação no domínio do problema em questão. Além disso, duas ferramentas, desenvolvidas a partir da arquitetura proposta, são apresentadas, assim como alguns resultados sobre a eficiência em sua utilização.*

1. Introdução

Cidade inteligente, também chamada cidade em rede, cidade digital ou cidade informacional, significa a cidade com sistema de informação que ajuda seu planejamento, construção e outros serviços. Geralmente, cidades inteligentes envolvem sistemas de informação que gerenciam serviços públicos oferecidos por uma cidade.

Uma questão bastante delicada, tratada de forma incipiente no planejamento urbano das grandes cidades, diz respeito ao processo decisório sobre as localizações das instalações do poder público capazes de oferecer serviços à população. Construir uma escola, um hospital ou um posto policial, sem ponderar questões como a demanda por tais serviços, a distância percorrida pela população para ser atendida, bem como o impacto no trânsito e nos custos dos serviços a partir desses deslocamentos, pode incorrer em verdadeiras catástrofes urbanas capazes de perdurar por longos anos.

Imagine uma situação na qual se deseja instalar postos de policiamento, dentro de um conjunto de locais candidatos em uma determinada cidade, a fim de reduzir o tempo em que as viaturas chegam aos locais onde estão ocorrendo os crimes. Supondo que se tem, a priori, informações sobre as demandas de todas as regiões da cidade por esse tipo de serviço e as distâncias entre elas e os locais candidatos, a tomada de decisão

sobre o melhor local a ser escolhido representa um problema de otimização combinatória de grande complexidade.

A problemática em questão é conhecida no ramo da Pesquisa Operacional como o Problema de Localização de Facilidades. Várias foram as soluções propostas, abrangendo desde métodos exatos a métodos aproximados que utilizam metaheurísticas.

Neste sentido, este artigo apresenta uma arquitetura de software para sistemas de informação no domínio do problema citado e duas ferramentas desenvolvidas utilizando tal arquitetura. Nestas são abordados três problemas específicos de localização de facilidades. Além disso, é apresentada uma validação da eficiência dos algoritmos utilizados nestas ferramentas.

O artigo está organizado da seguinte forma: a seção 2 discute os problemas conhecidos de Localização de Facilidades. A seção 3 apresenta a arquitetura proposta para alicerçar o desenvolvimento de sistemas no domínio do problema de localização de facilidades, assim como duas ferramentas que foram implementadas seguindo essa arquitetura. A seção 4 apresenta alguns resultados computacionais sobre as ferramentas. Finalmente, a seção 5 realiza as considerações finais.

2. Problemas de Localização de Facilidades

De forma geral, os problemas de localização tratam da questão de selecionar locais para a instalação de facilidades que oferecem serviços para atendimento de uma determinada demanda. Na realização desta tarefa, pode-se priorizar o atendimento da maior demanda possível ou a minimização das distâncias entre as facilidades e os usuários dos serviços disponibilizados por elas. Desta maneira, os problemas de localização podem ser divididos em duas classes de problemas, as quais diferenciam-se segundo o principal objetivo a ser atingido. São elas: problemas de cobertura e de localização de medianas.

Como representante dos problemas de cobertura, temos o Problema de Localização de Máxima Cobertura (PLMC) [Church e Reville 1974], o qual tem o objetivo de localizar um conjunto de facilidades de forma a prover a máxima cobertura para os serviços demandados. A noção de cobertura, no presente contexto, faz menção à definição de uma distância de serviço, que é a distância crítica além da qual a área de demanda não é considerada coberta [Galvão 1999]. Uma área de demanda, portanto, é considerada coberta se está localizada dentro dos limites da distância crítica.

Os problemas do tipo PLMC têm uma enorme aplicação no setor público, sendo utilizados na localização de serviços emergenciais e não-emergenciais. Enquadram-se no segundo grupo a localização de escolas, pontos de acesso a redes sem fio, agências de correios, entre outros. O grupo de serviços emergenciais inclui, por exemplo, a localização de hospitais, postos policiais, serviços de atendimento de emergência por ambulâncias e a localização das instalações do corpo de bombeiros.

Considerando que o conjunto de facilidades e pontos que geram demanda podem ser tratados como uma rede de nós, e que pode não haver instalações ou facilidades suficientes para atender a toda demanda, o modelo matemático, o qual visa maximizar a demanda atendida, pode ser descrito como segue:

Maximizar

$$\sum_{i \in U} h_i z_i \quad (1)$$

Sujeito a

$$\sum_{j \in N_i} y_j - z_i \geq 0, \forall i \in U \quad (2)$$

$$\sum_{j \in F} y_j = p \quad (3)$$

$$y_j \in \{0,1\}, \forall j \in F \quad (4)$$

$$z_i \in \{0,1\}, \forall i \in U \quad (5)$$

Onde,

$F = \{i; i = 1, 2, \dots, m\}$, representa o conjunto de locais candidatos para instalação das facilidades;

$U = \{j; j = 1, 2, \dots, n\}$, denota o conjunto de pontos que geram demanda;

D_c = distância de cobertura;

d_{ij} = distância entre a demanda do nó i e o candidato à facilidade j ;

$N_i = \{j \mid d_{ij} \leq D_c\}$, conjunto de todos os locais candidatos a facilidades que podem cobrir o ponto de demanda i ;

$y_j = 1$, se o vértice j é uma facilidade e $y_j = 0$, caso contrário;

h_i = demanda do nó i ;

p = número de facilidades para localizar;

$z_i = 1$ se a demanda do nó i é coberta e $z_i = 0$, caso contrário;

Como dito anteriormente, a função objetivo (1) tem como propósito maximizar a demanda total coberta. A restrição (2) garante que a demanda do nó i só é atendida se existir, no mínimo, um candidato à facilidade que possa atendê-la. A restrição (3) limita o número de facilidades a serem alocadas a um valor p , fornecido como entrada para o problema. As restrições (4) e (5) refletem a natureza binária das variáveis de decisão utilizadas no modelo.

O representante da outra classe de problemas de localização mencionada é o Problema das *P-Medianas* [Revelle e Swain 1970]. Em relação ao PLMC, há uma mudança de foco da cobertura máxima de atendimento para a minimização das distâncias entre as facilidades e os clientes que geram demandas. Sendo assim, seu principal objetivo é associar clientes a facilidades, de forma a reduzir somatório das distâncias entre eles. Apesar de haver a possibilidade de um cliente estar dentro da

região de cobertura de mais de uma facilidade, a associação mencionada ocorrerá apenas entre ele e uma única facilidade, a saber, aquela que estiver mais próxima.

Repare que o problema da *P-Mediana* trata a minimização das distâncias de forma global, isto é, sua preocupação é com o somatório de todas as distâncias entre clientes e suas respectivas facilidades. Uma variação interessante do problema em comento, denominado de *P-Mediana*^{max}, visa minimizar a máxima distância entre os elementos referidos [Correia 2011]. Os dois modelos são abordados com mais detalhes em seções posteriores neste trabalho.

A formulação matemática para o *P-Mediana* clássico é descrita logo abaixo:

Minimizar

$$\sum_{i \in U} \sum_{j \in F} d_{ij} x_{ij} \quad (7)$$

Sujeito a

$$\sum_{j \in F} x_{ij} = 1, \forall i \in U \quad (8)$$

$$x_{ij} \leq y_j, \forall i \in U, \forall j \in F \quad (9)$$

$$\sum_{j \in F} y_j = p \quad (10)$$

$$x_{ij} \in \{0,1\}, \forall i \in U \text{ e } y_j \in \{0,1\}, \forall j \in F \quad (11)$$

As variáveis de decisão, assim como os conjuntos que fazem parte do modelo acima têm significado idêntico àqueles que fazem parte do PLMC. Os únicos elementos que não constam no modelo PLMC são d_{ij} , que representa a distância entre um cliente i e a facilidade j , e x_{ij} , que indica com valor 1 que o cliente i foi atendido pela facilidade j , $x_{ij} = 0$ caso contrário.

A função objetivo (7) tem como propósito selecionar p vértices (facilidades ou medianas) de forma a minimizar o somatório das distâncias entre os vértices de demanda (clientes) e a facilidade mais próxima. As restrições (8) e (9) asseguram que cada vértice i (cliente) é vinculado a um único vértice j (facilidade). Já a restrição (10) determina o número de facilidades a serem localizadas, enquanto que o conjunto de restrições (11) estabelece que as variáveis de decisão assumam valores binários.

As ferramentas deste trabalho propõem soluções para cada um dos problemas apresentados, fazendo o uso de metaheurísticas em sua implementação. Sua utilização baseia-se no fato de que métodos exatos, os quais buscam pela solução ótima, se tornam ineficientes quando instâncias grandes são tratadas em problemas de complexidade NP-HARD. Como é sabido, os problemas supramencionados apresentam tal comportamento [Garey e Johnson 1979].

3. Arquitetura e Ferramentas para Problemas de Localização de Facilidades

Esta seção tem o objetivo de apresentar a proposta de arquitetura de software para sistemas de informação no domínio de problemas de localização de facilidades, com seus componentes e tecnologias. Além disso, são apresentadas duas ferramentas

desenvolvidas a partir de tal arquitetura, as quais objetivam solucionar problemas inerentes ao serviço público.

3.1. Arquitetura proposta

A Figura 1 ilustra a arquitetura proposta. Ela é composta por três camadas: *Facility Location Service*, *Client Layer* e *WebGIS Layer*, e segue a ideia de SOA (*Service-Oriented Architecture*), onde funcionalidades da aplicação são desenvolvidas como serviços para aplicações clientes ou outros serviços [Endrei et al. 2004]. Neste sentido, a camada *Facility Location Service* é responsável por oferecer serviços interoperáveis que, através de uma interface bem definida, resolvem problemas de localização de facilidades. Estes serviços são oferecidos à camada *Client Layer*, que é responsável pela GUI (*Graphical User Interface*). *WebGIS Layer* é a camada na qual recursos de algum Sistema de Informação Geográfico Web são disponibilizados a fim de que a GUI consiga informações geográficas para manipular mapas e localizações. As próximas subseções explicarão melhor cada camada em detalhes, seus componentes e tecnologias.

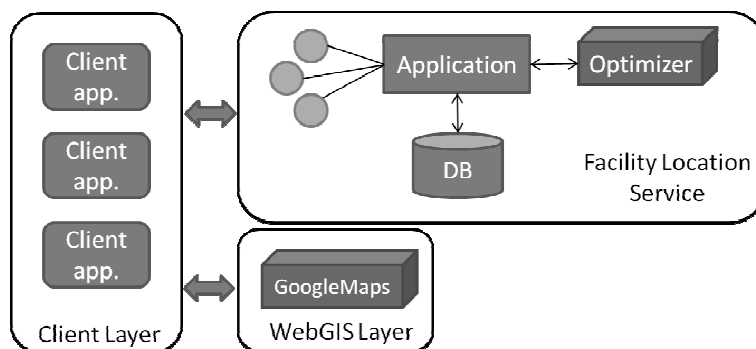


Figura 1. Arquitetura da proposta.

3.1.1. Facility Location Service

Esta camada possui basicamente três componentes: *Application*, onde é implementada a lógica da solução para problemas de localização; o *Optimizer*, onde reside a implementação do algoritmo otimizador; e um Banco de Dados onde as informações já processadas são armazenadas, para fins de desempenho. Além disso, o componente *Application* oferece um conjunto de serviços interoperáveis e reutilizáveis, com uma interface bem definida para a camada *Client Layer*, possibilitando que ferramentas clientes sejam desenvolvidas independentemente de linguagem de programação.

Os serviços oferecidos por esta camada são os problemas de localização de facilidades discutidos na seção 2: Problema de Localização de Máxima Cobertura (PLMC), Problema das *P-Medianas* e sua variação, o *P-Medianas*^{MAX}.

A aplicação foi implementada utilizando a plataforma JEE (Java Enterprise Edition). Foi utilizado o framework *Axis* para utilizar Web Services a fim de fornecer as funcionalidades através de serviços interoperáveis.

3.1.2 Implementações do Optimizer

A chave para a flexibilidade da camada *Facility Location Service* está no componente *Optimizer*, que é composto por uma coleção de componentes que variam conforme a

classe do problema de localização abordado. Em nossa proposta temos: Localização de Máxima Cobertura; P -mediana não-capacitado e P -mediana^{MAX}.

Mesmo resolvendo a mesma classe de problemas, podem existir vários componentes distintos que se diferenciam pela classe de algoritmos utilizados para sua resolução. Dessa forma, eles podem pertencer à classe de algoritmos exatos (*Branch & Bound*, *Branch and Cut* e Relaxação Lagrangeana) ou aproximados (metaheurísticas *GRASP*, *Tabu Search*, *VNS*).

Por conseguinte, o usuário além de escolher o tipo de problema que deseja resolver, tem a opção de escolher o método que melhor se adapte as dimensões da instância do problema e a necessidade da precisão da solução.

A metaheurística *GRASP - Greedy Randomized Adaptive Search Procedure* – foi proposta neste trabalho para a resolução do problemas de localização: P -mediana, P -mediana^{MAX} e Máxima Cobertura. A escolha desta metaheurística deve-se ao seu emprego com êxito em outros trabalhos que tratam de problemas de localização [Resende 2002], [Resende 2003].

O *GRASP* é um procedimento iterativo que combina várias propriedades favoráveis de outras heurísticas. Cada iteração do *GRASP* consiste de dois estágios: uma fase de construção da solução e uma fase de busca local. A melhor solução obtida dentre todas as iterações é considerada a solução final.

Na fase de construção de uma solução, iniciamos com um conjunto vazio que iterativamente recebe um elemento até formar uma solução viável. Nesta etapa, dois aspectos são analisados a cada iteração: o aleatório e o guloso. Esta dualidade deve-se ao fato de que o próximo elemento a ser escolhido para compor a solução partirá de uma lista, denominada Lista Restrita de Candidatos (LRC), que contém os α melhores elementos candidatos segundo um critério guloso especificado. O valor de α é um parâmetro do algoritmo.

As soluções obtidas na fase de construção do *GRASP* não são garantidas como ótimos locais, considerando uma dada vizinhança (diz-se que uma solução s é localmente ótima, se não existe uma solução melhor na vizinhança de s). Portanto, o emprego da segunda fase do *GRASP* é feito com o intuito de se melhorar a solução obtida na fase de construção. Um algoritmo de busca local é utilizado nesta segunda fase para, sucessivamente, substituir a solução atual por uma solução melhor, encontrada na vizinhança da solução atual.

A metaheurística *GRASP* é um algoritmo genérico para resolução de problemas combinatórios, logo, para que o mesmo seja aplicado a vários tipos de problemas distintos, há a necessidade da definição da representação do problema e da forma com a qual serão escolhidos os facilitadores que servirão cada cliente.

3.1.3 WebGIS Layer

Nos problemas de localização observamos que a matriz de distâncias é um dado de entrada necessário para sua resolução. Para que a solução seja mais próxima do real, é necessário que essas distâncias tenham uma boa aproximação, para isso propomos uma integração da arquitetura proposta com um *WebGIS*.

O *WebGIS* utilizado foi o do *Google Maps* que através de sua API, fornece uma biblioteca de serviços por meio de JavaScript. Dentre os serviços fornecidos, a aplicação faz uso especial de três: *setCenter*, para exibição dos mapas digitais; *getLocations*, para localização de pontos geográficos, permitindo a localização através de endereços reais ou por coordenadas de latitude/longitude; e *loadFromWayPoints*, que cria rotas para destinos especificados e também calcula a distância entre pontos [MAPS 2012].

3.1.4. Client Layer

Diferentes tipos de clientes podem ser desenvolvidos para usar os serviços oferecidos pelas camadas *Facility Location Service* e *WebGIS*. O objetivo destas aplicações clientes é oferecer mecanismos para interação com o usuário. Uma vez que as funcionalidades dos problemas de localização foram implementadas e disponibilizadas via Web Services, estas aplicações clientes podem ser implementadas utilizando qualquer linguagem de programação. Este é um aspecto importante, uma vez que aumenta a diversidade de possibilidades para desenvolver estas aplicações.

3.2 Implementações das Ferramentas

Como mencionado anteriormente, a aplicação proposta para este trabalho é um sistema de apoio ao planejamento público que permite entidades como governos e empresas prestadoras de serviços planejarem com mais eficiência o oferecimento de seus serviços ao público. Neste sentido, duas ferramentas foram desenvolvidas seguindo a arquitetura discutida na seção anterior: Ferramenta para Posto de Coleta de Transmissão (PCT) do TRE e Ferramenta para Máxima Cobertura de Clientes. Estas duas ferramentas são discutidas nas subseções seguintes.

3.2.1. Ferramenta para Posto de Coleta de Transmissão (PCT) do TRE

Atualmente o Processo Eleitoral Brasileiro, utiliza a seguinte logística na realização das eleições de cada cidade: inicialmente as urnas são distribuídas para seus respectivos locais de votação; depois de encerradas as eleições as mídias são levadas a pontos de coletas e transmissão (PCT), que são escolhidos empiricamente, e depois de apurados os votos, o resultado é divulgado.

Para que o resultado da eleição seja divulgado mais rápido, cada entidade eleitoral local precisa minimizar o tempo de deslocamento das mídias aos PCTs. Desse modo é necessário escolher entre todas as zonas eleitorais da cidade algumas para serem instalados pontos de internet que pudessem receber os votos das urnas das demais zonas e transmiti-los ao TRE-PB. A ferramenta surgiu como uma solução permitindo aos gestores melhores escolhas das zonas que poderiam se tornar PCTs minimizando a média das distâncias que as demais zonas eleitorais teriam que percorrer para levar os votos das urnas.

A ferramenta é uma aplicação Web, desenvolvida em *PHP*, onde o acesso se torna possível através de um browser.

Os principais requisitos desta ferramenta são:

- **Cadastro de Escolas Eleitorais:** Indica a localização de cada escola eleitoral, pelo endereço, pela latitude e longitude ou através da posição no mapa.

- **Otimização de concentradores:** Minimiza as distâncias entre as escolas e seus Pontos de Concentração e Transmissão (PCT).
- **Geração de Relatórios:** Lista as escolas eleitorais associadas a cada PCT e os custos desta associação.

Nesta ferramenta, a interface possibilita o usuário definir que parte da cidade ele deseja inserir pontos candidatos para serem facilitadores (PCT), através de um mapa, utilizando o GoogleMaps.

O usuário poderá escolher a quantidade de PCTs que ele deseja planejar e verificar a otimização. Depois disso, o usuário poderá escolher duas opções de otimização: (1) Minimizar a máxima distância entre as escolas e PCTs, ou (2) Minimizar somatório das distâncias entre as escolas e PCTs. Estas duas opções são, respectivamente, a utilização da classe de problema P -mediana^{MAX} e P -Mediana.

É importante lembrar que esta ferramenta utiliza os serviços da camada *Facility Location Service* da arquitetura, para poder realizar as otimizações, e utiliza um WebGIS (GoogleMaps) para obter os mapas e as informações geográficas.

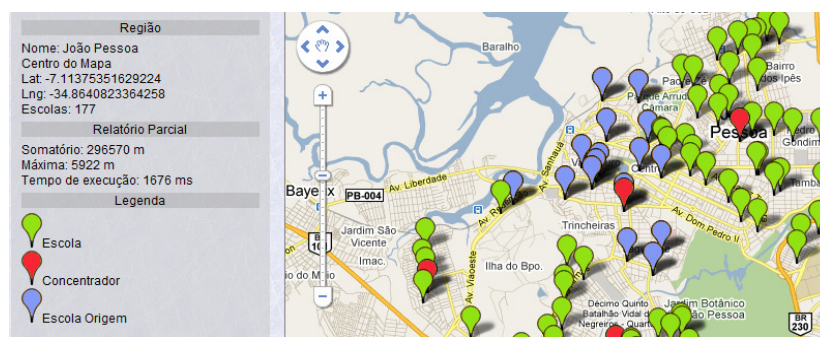


Figura 2. Tela da Ferramenta mostrando a otimização realizada.

3.2.2. Ferramenta de Máxima Cobertura de Clientes

Esta ferramenta propõe resolver o problema de localização de máxima cobertura descrito na seção 2. Na ferramenta, um conjunto de pontos candidatos é marcado em um mapa, assim como um conjunto de clientes com suas demandas. A ferramenta então mostra quais destes pontos candidatos são escolhidos a fim de atender a maior demanda possível dos clientes. A Figura 3 ilustra um exemplo de resultado que a ferramenta gerou. Cada cliente possui um valor de demanda específico. O usuário define a quantidade de facilitadores que ele deseja ativar de um conjunto de candidatos.

Esta ferramenta também foi desenvolvida em *PHP*, e pode ser acessada via browser. Os requisitos principais desta ferramenta são:

- **Cadastro de Clientes:** Permite cadastrar a localização de cada cliente, pelo endereço, pela latitude e longitude através da posição no mapa.
- **Otimização da Cobertura de Clientes:** Maximiza a cobertura dos clientes pelos seus facilitadores.
- **Geração de Relatórios:** Lista os clientes associados a cada facilitador e os custos desta associação.

O usuário ainda pode escolher a opção para o cálculo das distâncias, que podem ser (1) *Driving*, que faz medições considerando rotas pelas rodovias levando em conta os sentidos de mão e contramão do trânsito, (2) *Walking*, que calcula a distância considerando o caminho que uma pessoa pode fazer sem considerar o sentido do trânsito, e (3) *Air*, onde os valores das medidas são obtidos pela distância linear entre as localidades, considerando o raio de curvatura do globo terrestre.

Seguindo a arquitetura proposta, esta ferramenta utiliza o serviço de Máxima Cobertura da camada *Facility Location Service*, para executar a otimização.

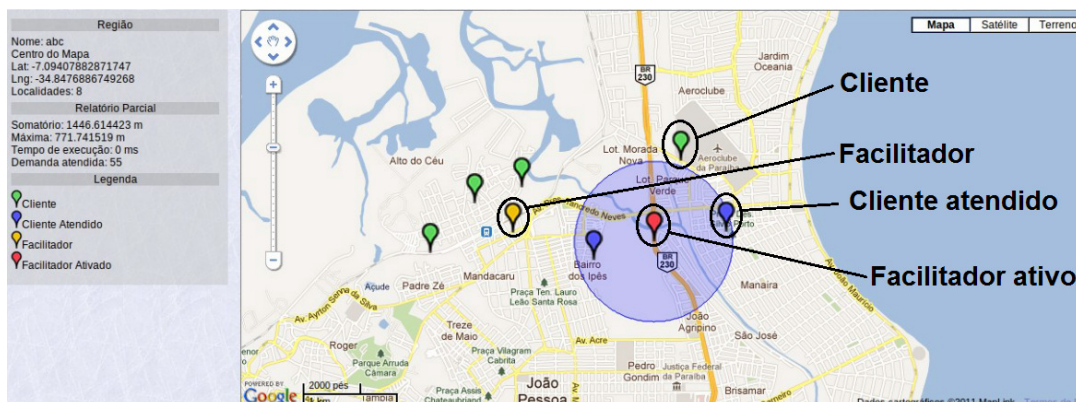


Figura 3. Tela da Ferramenta mostrando a otimização realizada.

4. Resultados Computacionais

Além de ter uma validação da arquitetura, através da implementação de duas ferramentas, é importante validar também os resultados que estas ferramentas apresentam em relação aos problemas propostos. Neste sentido, esta seção apresenta os resultados computacionais obtidos com as implementações de três componentes otimizadores para a resolução de três problemas distintos de localização: *P-mediana*, *P-mediana*^{MAX} e Cobertura Máxima. A metaheurística GRASP foi adaptada para a resolução de cada um dos problemas.

Os componentes que resolvem o *P-Mediana* e *P-Mediana*^{MAX} foram aplicados a um estudo de caso com o Tribunal Regional Eleitoral, já o componente que resolve a Cobertura Máxima teve seus resultados comparados com a literatura.

Estas metaheurísticas foram implementadas utilizando a linguagem C++. Todos os experimentos computacionais foram realizados em um computador com sistema operacional Linux Ubuntu 9.04 e com processador Intel Core 2 Quad 2.33 Ghz com 2 GB de memória RAM.

4.1 Aplicação no Tribunal Regional Eleitoral

A instância utilizada para a obtenção dos resultados apresentados nesta seção foi gerada a partir das informações fornecidas pelo Tribunal Regional Eleitoral da Paraíba, dados estes relacionados com o pleito de 2008. Foram fornecidas as localizações de 177 locais de votação utilizados na cidade de João Pessoa, a matriz de distâncias foi calculada com o auxílio do *WebGIS Layer* abordado na seção 3, gerando assim uma instância intitulada JP_177. Essa mesma instância foi dividida por zonas para gerarmos outras instâncias, onde o número de elementos e de concentradores varia (Z01, Z64, Z70, Z76, Z77).

Para efeito de comparação entre os resultados obtidos para as duas classes de problemas, o *P-Mediana* original e sua variação *P-Mediana^{MAX}*, calculamos os valores da solução real gerada pela equipe de logística do TRE-PB no ano de 2008, que não utilizou um sistema para a tomada de decisão, sendo suas escolhas efetuadas apenas pela experiência que a equipe obteve no planejamento de outros pleitos.

A primeira coluna da Tabela 1 define a instância testada. As colunas restantes são divididas em três grupos: TRE-PB, *P-Mediana* e *P-Mediana^{MAX}*. No caso do TRE-PB a coluna *Soma* indica o valor obtido pela soma das distâncias, em metros, entre as Escolas Eleitorais e seu respectivo PCT definido em seus relatórios. Já a coluna *Max_dist* representa a maior distância entre uma escola eleitoral e seu PCT. Para os grupos *P-Mediana* e *P-Mediana^{MAX}*, as colunas encontradas são: *Soma* e *Max_dist* que indicam o valor obtido pela metaheurística GRASP e, Δ indica o *gap*, que é a diferença percentual entre as soluções:

$$\Delta = [(z - z^*) / z^*] \times 100. \quad (13)$$

Tabela 1. Comparação dos resultados do TRE-PB com a metaheurística GRASP (*P-Mediana* e *P-Mediana^{MAX}*).

Instância		TRE-PB (z)		P-Mediana (z*)				P-Mediana ^{MAX} (z*)			
Nome	p	Soma	Max_dist	Soma	Δ (%)	Max_Dist	Δ (%)	Soma	Δ (%)	Max_Dist	Δ (%)
JP_177	26	277486	8821	140172	-49,485	4107	-53,441	142410	-48,678	2524	-71,386
Z01	3	59252	3384	45576	-23,08%	3123	-7,71%	51598	-12,92%	2384	-29,55%
Z64	10	31176	4283	9533	-69,42%	753	-82,42%	9591	-69,24%	753	-82,42%
Z70	4	53743	8394	34254	-36,26%	2594	-69,10%	34577	-35,66%	2594	-69,10%
Z76	5	75990	8821	44498	-41,44%	6525	-26,03%	48332	-36,40%	3316	-62,41%
Z77	4	57325	4153	42037	-26,67%	3078	-25,88%	44165	-22,96%	2840	-31,62%

O *P-Mediana* obteve melhorias, com relação a soma das distâncias, superiores a 23% em todas as instância, comparadas ao resultado do TRE-PB, atingindo 69,42% de ganho na instância Z64.

Podemos observar que os resultados obtidos pelo *P-Mediana^{MAX}* com relação a máxima distância, com relação a instância JP_177/p=26, alcançaram 71,3% de melhoria com relação ao valor da solução usada pelo TRE-PB, já o modelo *P-Mediana* original melhorou 53,4%, que é um bom resultado, porém bem abaixo do resultado do *P-Mediana^{MAX}* proposto neste trabalho.

Podemos observar também que o modelo *P-Mediana^{MAX}* obteve esta melhora sem aumentar muito o somatório das distâncias, obtendo uma melhoria de 48,6% do resultado do TRE-PB contra 49,4% do *P-Mediana* original.

4.2 Resultados do Otimizador de Máxima Cobertura

Lorena (2002) aplicou uma heurística Lagrangean/surrogate (HLS) para o problema de Máxima Cobertura, onde algumas das restrições do modelo MCLP apresentadas na seção 2 são relaxadas. Na tentativa de diminuir a complexidade do problema original em seus experimentos computacionais, foram usados dados do mundo real. Os dados do mundo real com 402 e 818 localidades (instâncias SJC) foram obtidos em uma base de dados georeferenciada da cidade de São José dos Campos, Brasil. As localidades representam porções da cidade e o número de casas, apartamentos e prédios públicos e

comerciais em cada bloco representa a demanda a ser coberta pela solução (os dados estão disponíveis em <http://www.lac.inpe.br/~lorena/instancias.html>).

Tabela 2. Resultados computacionais para SJC402.

Instância			HLS		GRASP			
N	p	S	Z	Tempo(s)	Z*	Tempo (s)	Δ	
402	1	800	6555	0,55	6555	0,095	0	
402	2	800	11339	10,16	11339	0,201	0	
402	3	800	14690	11,09	14690	0,383	0	
402	4	800	15658	13,73	15658	0,456	0	
402	5	800	15970	29,11	15970	1,092	0	
402	6	800	15984	38,01	15984	0,671	0	
402	1	1200	10607	0,71	10607	0,139	0	
402	2	1200	14832	7,14	14832	0,244	0	
402	3	1200	15984	13,46	15984	0,358	0	
402	1	1600	15438	0,77	15438	0,19	0	
402	2	1600	15984	11,87	15984	0,314	0	

Tabela 3. Resultados computacionais para SJC818.

Instância			HLS		GRASP			
N	P	S	Z	Tempo(s)	z*	Tempo (s)	Δ	
818	1	800	8393	1,48	8393	0,324	0	
818	2	800	13306	29,16	13306	0,756	0	
818	3	800	17507	37,02	17507	1,036	0	
818	4	800	21428	43,83	21428	3,236	0	

818	5	800	24531	51,03	24531	1,826	0
818	6	800	25908	73,87	25908	6,494	0
818	7	800	26933	99,8	26896	12,476	0,13
818	8	800	27783	129,84	27813	8,612	-0,10
818	9	800	28351	158,02	28372	20,198	-0,07
818	10	800	28639	197,79	28719	10,987	-0,27
818	11	800	29019	215,36	29053	11,062	-0,11
818	12	800	29103	283,91	29138	7,038	-0,12
818	13	800	29144	299,89	29143	26,812	0,003
818	14	800	29168	337,02	29168	12,784	0
818	1	1200	11612	1,71	11612	0,584	0
818	2	1200	20290	49,16	20290	1,241	0
818	3	1200	25211	35,21	25211	1,844	0
818	4	1200	27029	52,95	27029	5,081	0
818	5	1200	28513	89,97	28513	3,686	0
818	6	1200	29137	106,61	29137	8,793	0
818	7	1200	29168	137,31	29168	4,065	0
818	1	1600	16827	2,64	16827	0,616	0
818	2	1600	24646	43,72	24646	1,144	0
818	3	1600	27672	53,89	27672	1,724	0
818	4	1600	28862	62,28	28862	2,056	0
818	5	1600	29168	110,4	29168	2,447	0

As Tabelas 2 e 3 apresentam os resultados obtidos pelos problemas SJC402 e SJC818. A metaheurística *GRASP* para o problema de Máxima Cobertura obtém na maioria das soluções o mesmo valor obtido pela estratégia *HLS* cujos resultados foram extraídos de Lorena (2002), sendo seu tempo computacional mais eficiente. Podemos observar também que na instância SJC818 o *GRASP* obteve um *gap* negativo em cinco instâncias, o que significa ter obtido resultados melhores que a estratégia *HLS*.

5. Considerações Finais

Neste trabalho foi apresentada uma arquitetura de software para sistemas de informação em um domínio de problemas de localização de facilidades. Esta arquitetura foi projetada com o intuito de flexibilizar o desenvolvimento de ferramentas deste domínio. Para validarmos a flexibilidade desta arquitetura, implementamos duas ferramentas que se utilizaram de três serviços distintos de otimização para os problemas de localização: Localização de Máxima Cobertura, *P-mediana* não capacitado e *P-mediana*^{MAX}.

Nos problemas de localização a matriz de distâncias é um dado de entrada necessário para sua resolução. Para que a solução dos mesmos seja mais próxima do real, é necessário que essas distâncias tenham uma boa aproximação, para isso propomos uma integração da arquitetura proposta com um WebGIS (Google Maps).

Os resultados computacionais apresentados demonstram a boa qualidade das soluções geradas pelos componentes otimizadores. No estudo de caso do TRE-PB,

foram abordados os componentes para os problemas *P-mediana* e *P-mediana^{MAX}*, neste estudo foram obtidas melhoras acima de 50% comparadas a solução atual da instituição.

Com relação ao problema de Máxima Cobertura, a estratégia GRASP proposta se mostra mais eficiente em todas as instâncias com relação ao tempo computacional e nas instâncias de maior dimensão a estratégia é mais eficaz em muitas instâncias, encontrando melhores soluções comparadas à literatura.

Referências

- Church, R. & ReVelle, C.S. (1974). The maximal covering location problem. Papers of the Regional Science Association, 32, 101-118.
- Correia, J. H. (2011). Uma Ferramenta Web integrada a métodos híbridos aplicados a Problemas de Localização. Dissertação (Mestrado em Engenharia de Produção) — Universidade Federal da Paraíba, João Pessoa.
- Endrei, M., et al. (2004). Patterns: Service-Oriented Architecture and Web Services, IBM International Technical Support Organization. Redbooks, April.
- Garey, M.R.; Johnson, D.S. (1979) Computers and intractability: a guide to the theory of NP-completeness. San Francisco: W. H. Freeman and Co.
- Galvão, R. G.; Nobre, F. F.; Vascolcellos, M. M. (1999). Modelos matemáticos de localização aplicados à organização espacial de unidades de saúde. Rev. de saúde pública, vol. 33.
- Lorena, L. A. N. and Pereira M. A. (2002). A Lagrangean/surrogate heuristic for the maximal covering location problem using Hillsman's edition. International Journal of Industrial Engineering 9(1), 57-67.
- MAPS, Google. (2012). Família da Google Maps API. Disponível em: <https://developers.google.com/maps/?hl=pt-br>. Acessado em: 23/03/2012.
- Resende, M. G. C. e Werneck, R. F. (2002). A GRASP with path-relinking for the p-median problem. Technical Report TD-5E53XL, AT & T Labs Research.
- Resende, M. G. C. e Werneck, R. F. (2003). On the implementation of a swap-based local search procedure for the p-median problem. In R. E. Ladner, editor, Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments (ALENEX'03), p. 119-127. SIAM.
- Revelle, C. S. e Swain, R. W. (1970). Central Facilities Location. Geographical Analysis, 2: 30-42.