“Comuty” - Um Approach para Otimização da Comuta Universitária com Baldeação.

Alef Berg[1], Danillo Medeiros[1], Ednaldo Martins[1], Fábio A. E. Melo[1], Hiago Vicktor[3]

[1]Ciência da Computação – Centro de Informática – Universidade Federal da Paraíba (UFPB) – João Pessoa – PB – Brasil

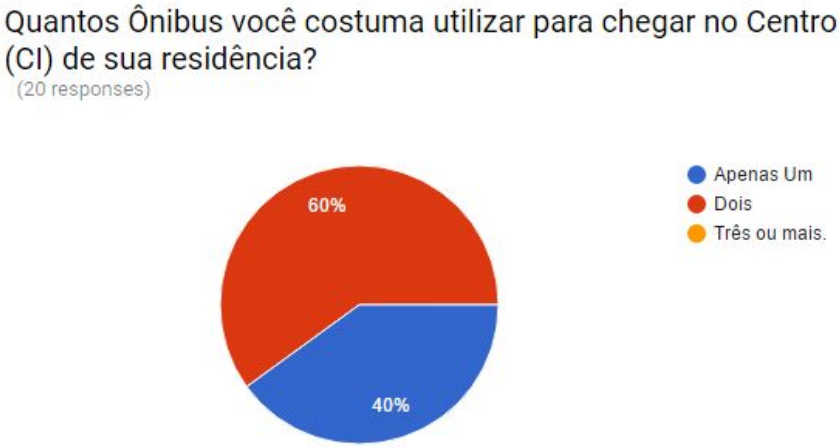
Abstract. *To-do.*

Resumo. *Este.*

1. Introdução

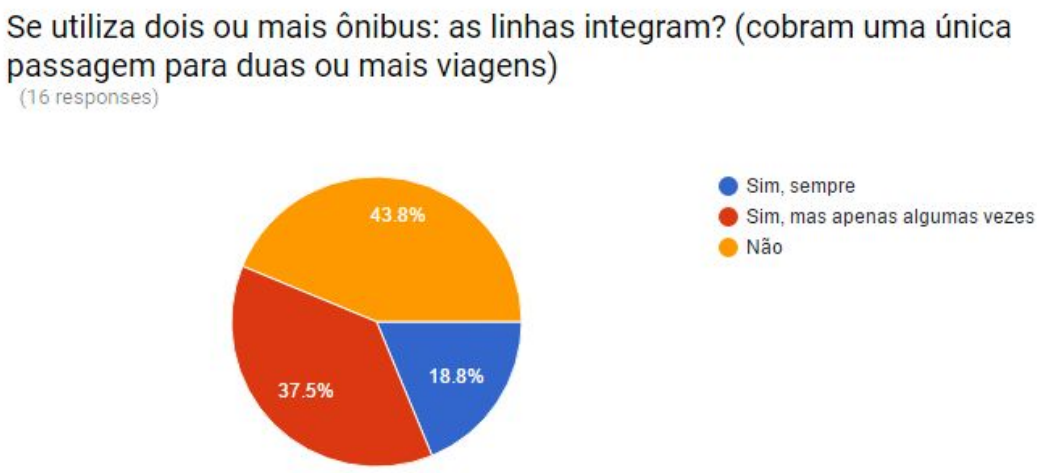
Em boa parte dos bairros na cidade de João Pessoa, o transporte público pode ser considerado como ineficiente e problemático, o que torna cada vez mais difícil o transporte diário. A motivação para este projeto vem de um problema real que afeta muitos estudantes do CI: a dificuldade de baldeação e o longo tempo de comuta para o campus.

Em uma pesquisa anônima feita com alunos do Centro de Informática, é possível observar que há uma grande porcentagem dos alunos do CI utilizam de dois ônibus



É também notável que, apesar da existência de rotas que fazem integração de passagens

permitindo uma baldeação sem custo adicional, nem sempre é possível para o público usufruir de rotas com tal benefício. como observado na pesquisa, muitas combinações de rotas não possuem essa funcionalidade.



Por meio de coleta de dados via "crowdsourcing", podemos extrair essas informações e utilizá-las para popular no nosso modelo.

Exemplo de dados de Integração:

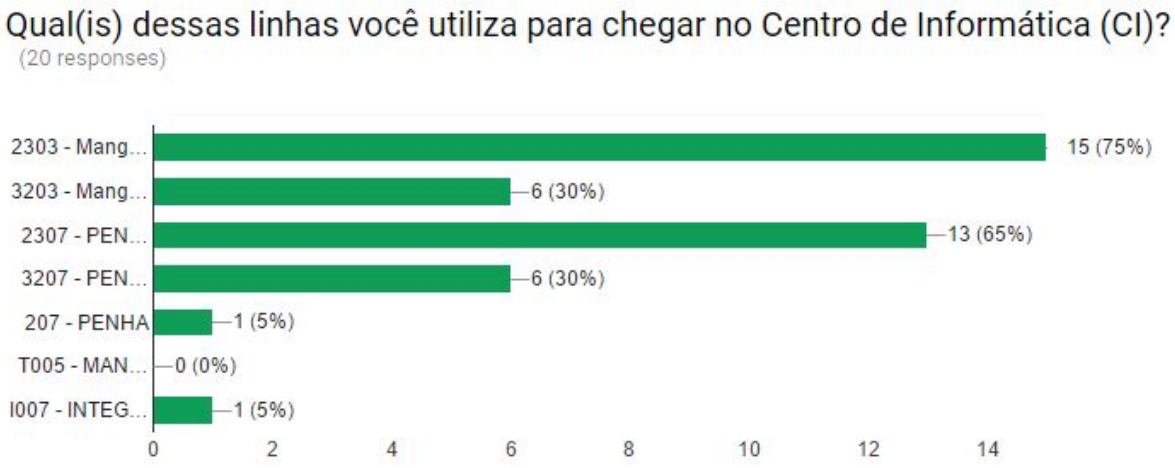
Linha 1	Linha 2	Integram?	Confirmações
3203	2501	SIM	5 Usuários
2307	5201	NÃO	3 Usuários
5110	2307	NÃO	2 Usuários

2. Fundamentação Teórica

O Agente "Comuty" se aplica de fontes de dados em tempo real de localização dos ônibus e do usuário para mostrar a rota mais eficiente e barata, quando se utiliza de mais de uma linha de ônibus por vez. o foco do agente é de tornar essa baldeação a mais agradável possível.

Para implementar este agente, utilizamos dois *layers* de linhas distintos: as linhas que imediatamente chegam no destino (nível A), e as linhas que passam próximo à essas linhas(nível B).

Através de pesquisa, obtivemos as linhas mais populares para o CI, das quais utilizaremos neste projeto (2303,3203,2307,3207) como rotas de nível A.



O Agente calcula a heurística da baldeação quando o ponto em que uma linha de nível B chegar em uma rota onde se encontra uma linha de nível A, como demonstrado na figura abaixo. (no exemplo, na parada descrita, o usuário desce do 1510 e embarca no 3207.

Subsequentemente o Agente deve retornar ao usuário opções de rotas possíveis.

Exemplo: um estudante parte do bairro do Geisel com destino ao CI, o agente retorna estas opções de ônibus:

#	Linha 1	Linha 2	Tempo para Desembarque e Baldeação	Tempo de Baldeação para o Ônibus 2	Chegada no Destino Final	Integra?
1	5201	2303	25min	5 min	45 min	SIM
2	2501	3203	20 min	20 min	75 min	SIM
3	2501	3207	20 min	5 min	60 min	NÃO

3. Modelagem e Implementação do Agente Inteligente

No sistema *Comuty,* utilizamos dois agentes que funcionam em conjunto na forma de cliente/servidor.

Em questão da viabilidade de obtenção de dados, utilizaremos uma API externa para obter os dados necessários para o projeto, e uma base de dados estática como failsafe, caso tais dados dinâmicos estejam indisponíveis.

3.1 Descrição PEAS

Agente	Sensores (Entrada)	Atuadores	Medidas de Desempenho	Ambiente
Comuty-Backend	Posições de GPS de todas as linhas de Ônibus aplicáveis ao problema. Localização dos Ônibus em Relação ás Paradas de Ônibus e seus respectivos Identificadores	Envia o valor de Tempo de Espera e IDs dos ônibus para o Comuty-User, (quando requisitado.)	Definição de Precisão do valores enviados por meio de testes automatizados.	Base de Dados e Mapas de Rotas de Ônibus da Cidade. Servidor Local ou, Backend na Nuvem (AWS, Azure ou Heroku)
Comuty-User	Localização do Usuário. (Proximidade de uma das Linhas de Ônibus unitárias) Linhas de Ônibus Disponíveis na Localização do Usuário (Resposta do Comuty-Backend) Tempo de Espera das Linhas Relevantes (Resposta do Comuty-Backend)^	Exibe as principais possíveis soluções no Aplicativo	Classificação dos Resultados em um valor definido pelo usuário. (Em 1-5 Estrelas)	User-level: Smartphone, Desktops, Sistemas IOT (Monitores Públicos de Rotas)

3.2 Propriedades do Ambiente

- **Completamente Observável:** Como o agente possui os dados de localização de

todas as linhas em circulação em João Pessoa e um mapa completo da cidade. pode ser criado um modelo relativamente acurado do ambiente urbano.

- **Determinístico:** Os estados subsequentes do agente podem ser previstos por suas escolhas, de acordo com os dados disponíveis.
- **Dinâmico:** Os valores dos objetos no ambiente (tempo de espera, posição do ônibus) estão em constante mudança.
- **Discreto (Módulo de Usuário/Comuty-User):** existe um número finito de decisões possíveis que o agente pode deliberar. não pode ser considerado **contínuo** pois se é tirado um *snapshot* dos dados disponíveis no momento de interação com o usuário, para que se consuma menos recursos e seja montada uma solução mais coerente.
- **Contínuo (Módulo de Servidor/Comuty-Backend):** pois monitora de forma contínua a posição dos ônibus e calcula seu tempo, para que possa enviar a versão mais recente dos dados para o **User**
- **Multiagente Cooperativo:** Utiliza-se de dois agentes que funcionam em cooperação, um como provedor de dados (*Comuty-Backend),* e outro como receptor e interpretador, para que seja formada a solução final (*Comuty-User).*

3.3 Tipo de Agente

A Plataforma *Comuty* pode ser categorizada como **Agente Otimizador baseado em Utilidade**, pois compara entre diferentes rotas e estados de mundo, e utiliza-se desses dados para definir o valor de sua função de utilidade, gerada por forma de algoritmos de busca heurística e de critérios definidos pelo usuário.

3.4 Esboço Conceitual em Pseudocódigo

```
class parada // objeto parada

{
int ID_Parada //identificador único de parada
int lat,long // latitude e longitude
Array int Ônibus que Passam no Local

//

}

class rota // rota de um ônibus

int codigo_onibus // código do Ônibus
Paradas[TOTAL_PARADAS] // Array de Objetos da Classe Parada.
interface para receber Tempo de Espera (minutos)


class integração //verifica se o ônibus integra

integra(codigo primeiro_onibus, codigo segundo_onibus){

       retorna true se integra, false se não}

/* Funções */

Previsão(parada p, char** parametros),
```

```
/*
for(i=0; i < numero arbitrário de soluções; i++)
recebe objeto parada (inicial);
recebe parâmetros. (integra sim/não)(rota mais rápida ou barata).
executa algorítimo heurístico
retorna um array ordenado do objeto solução. */

Solução(onibus o, parada p)

/*
função que recebe objeto Ônibus e Parada
caso mais de um ônibus, retorna ponto de baldeação.
calcula tempo
calcula baldeação
calcula pesos.
retorna solução. */
```

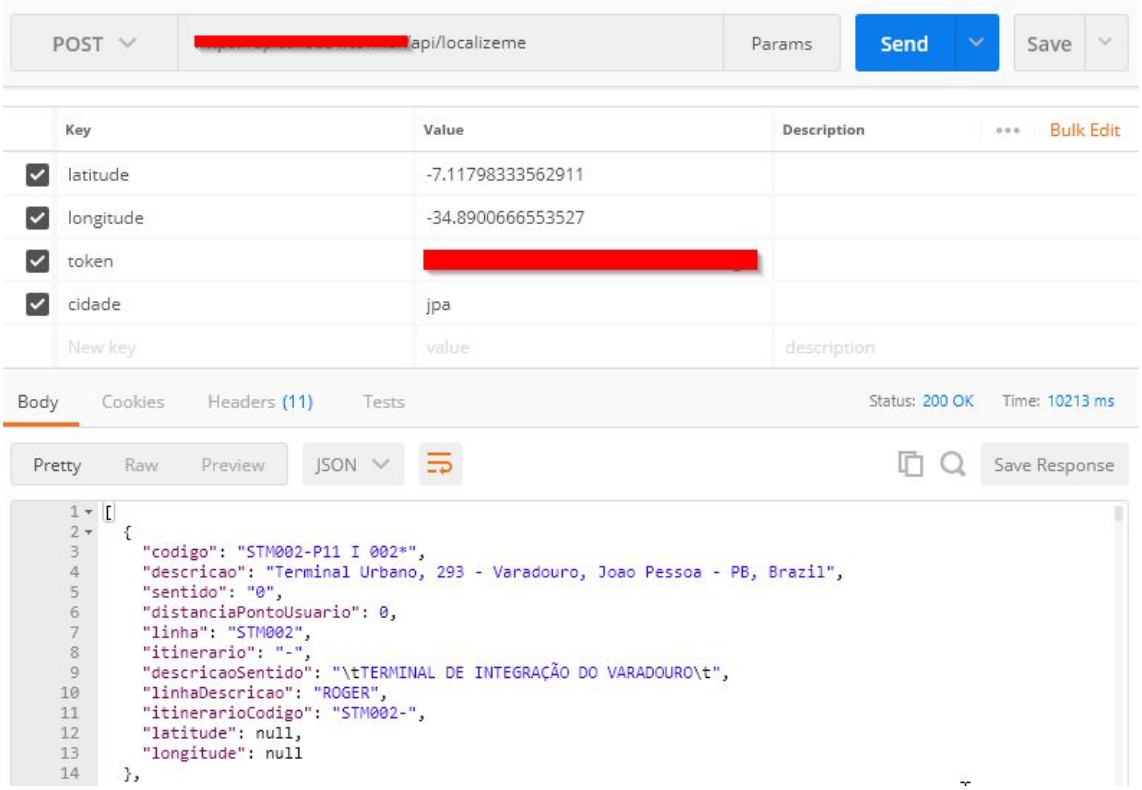
4. Métodos

Utilizamos a Busca A* com Pesos para cada membro. (elaborar)

Para que possamos fazer a pesquisa pela melhor opção possível.

- Obtemos os dados da Localização do Usuário através do aplicativo *cliente (Comuty-User).*
- Após encontrado, são localizadas os IDs das paradas mais próximas desta localização e dos ônibus que permutam por essas paradas.
- Linhas que em suas rotas compartilham paradas com as linhas *finais* determinadas pelo banco de dados do servidor (*Comuty-Backend)* são adicionadas ao escopo da busca ex: linhas_finais = {2303,2307,3203,3207,207,T005}
- Linhas em que suas rotas passam próximo de paradas onde passam as linhas *finais* (em até 100m) que se encontram também são verificadas, com uma adição extra de pontuação no cálculo do peso.
-

Exemplo de Request/Resposta para o Servidor *Backend* usando HTTP-POST

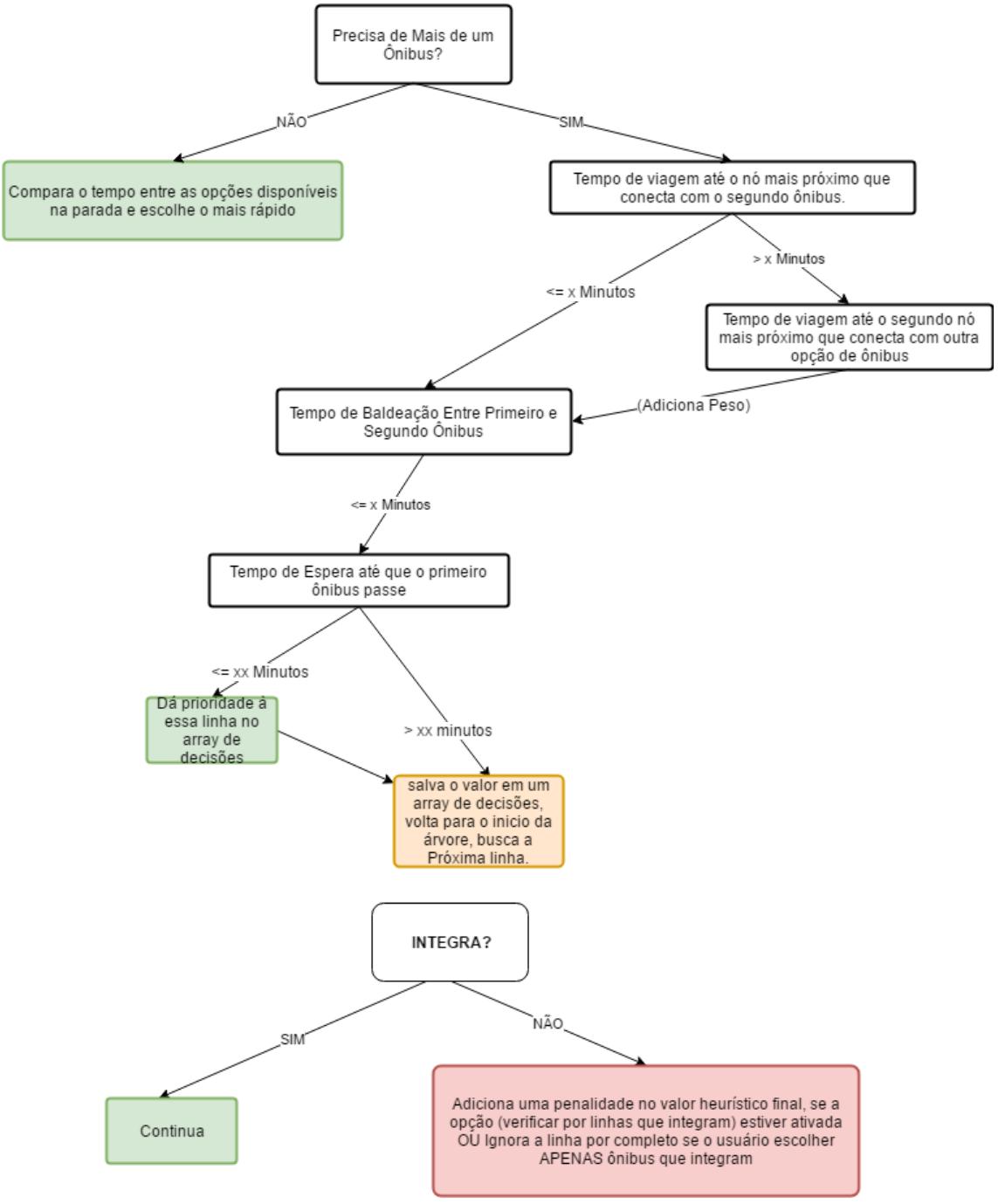


São descartados ônibus que vão em destinos opostos

Através paraPara cada

4.1 Resolução de Problemas por Meio de Busca

4.1.2 Árvore de Decisão para Cálculo de Pesos



4.2 Representação do Conhecimento e Raciocínio

5. Resultados

Para

Referências

http://www.devmedia.com.br/introducao-a-web-semantica/26181

http://www.w3c.br/Padroes/WebSemantica

http://www.scielo.br/pdf/ci/v33n1/v33n1a16

https://pdfs.semanticscholar.org/566c/1c6bd366b4c9e07fc37eb372771690d5ba31.pdf

tem o pdf pra baixar:

https://periodicos.ufsc.br/index.php/eb/article/view/155

https://portalseer.ufba.br/index.php/revistaici/article/view/2669