



**UNIVERSIDADE FEDERAL DO TOCANTINS
CAMPUS UNIVERSITÁRIO DE PALMAS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL DE
SISTEMAS**

VILSON SOARES DE SIQUEIRA

**SISROUTING: UM SISTEMA DE APOIO A DECISÃO COM A
UTILIZAÇÃO DA METAHEURÍSTICA GRASP APLICADA
PROBLEMA DE ROTEAMENTO DO ÔNIBUS ESCOLAR**

**BRASIL
2016**



**UNIVERSIDADE FEDERAL DO TOCANTINS
CAMPUS UNIVERSITÁRIO DE PALMAS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL DE
SISTEMAS**

VILSON SOARES DE SIQUEIRA

**SISROUTING: UM SISTEMA DE APOIO A DECISÃO COM A
UTILIZAÇÃO DA METAHEURÍSTICA GRASP APLICADA
PROBLEMA DE ROTEAMENTO DO ÔNIBUS ESCOLAR**

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional de Sistemas, da Universidade Federal do Tocantins, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional de Sistemas.

Orientador: Prof. Dr. Marcelo Lisboa Rocha

**BRASIL
2016**

**Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal do Tocantins**

S676s Soares de Siqueira, Wilson.
 SISROUTING: Um Sistema De Apoio A Decisão Com a Utilização da
 Metaheurística GRASP Aplicada Problema De Roteamento Do Ônibus
 Escolar. / Wilson Soares de Siqueira. – Palmas, TO, 2016.
 59 f.

 Dissertação (Mestrado Profissional) - Universidade Federal do Tocantins,
 Campus Universitário de Palmas - Curso de Pós-Graduação (Mestrado) em
 Modelagem Computacional de Sistemas, 2016.

 Orientador: Prof. Dr. Marcelo Lisboa Rocha

 1. Otimização Combinatória. 2. Ônibus Escolar. 3. SBRP. 4.
 Metaheurística. 5. GRASP. I. Rocha, Marcelo Lisboa. II. Universidade
 Federal do Tocantins. III. Título.

CDD 4

TODOS OS DIREITOS RESERVADOS – A reprodução total ou parcial, de qualquer
forma ou por qualquer meio deste documento é autorizado desde que citada a fonte.
A violação dos direitos do autor (Lei nº 9.610/98) é crime estabelecido pelo artigo 184
do Código Penal.



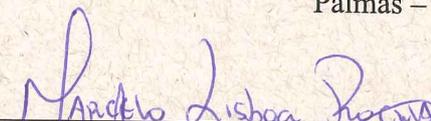
ATA Nº 06/2016

ATA DA SESSÃO PÚBLICA DE APRESENTAÇÃO E DEFESA DE
DISSERTAÇÃO

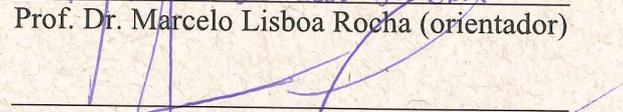
Aos 29 dias do mês de Fevereiro de 2016, realizou-se no auditório do Bloco III , na Fundação Universidade Federal do Tocantins, Campus Universitário de Palmas, às 14 horas, a defesa de dissertação de mestrado do aluno **Vilson Soares de Siqueira**, regularmente matriculado no Curso de Pós-Graduação em Modelagem Computacional e Sistemas, com o título **“SISRROUTING: Um Sistema de Apoio à Decisão com a Utilização da Metaheurística GRASP Aplicado ao Problema do Roteamento do Ônibus Escolar”**, perante a Comissão Julgadora aprovada pela Comissão Coordenadora do curso como segue: Prof. Dr. Marcelo Lisboa Rocha (orientador), Prof. Dr. Geoge França dos Santos (membro interno), Prof. Dr. Waldecy Rodrigues (membro externo) e Prof. Dr. Gerson Pesente Focking (membro externo). Após a exposição do seu trabalho, o aluno foi arguido pelos componentes da Comissão. O Senhor Presidente, Prof. Dr. Marcelo Lisboa Rocha, solicitou que o aluno se retirasse para que a Comissão procedesse ao julgamento. O aluno foi APROVADO, sendo-lhe atribuído o grau de Mestre em Modelagem Computacional de Sistemas desde que cumpridas as exigências descritas pela Comissão Julgadora, além daquelas listadas pelo Regulamento do Curso em um prazo de 30 dias. Além dessas exigências, o aluno deverá entregar na Coordenação do Mestrado Profissional em Modelagem Computacional de Sistemas uma cópia do Termo de Autorização para Publicização Digital de Teses e Dissertações na Biblioteca Digital de Teses e Dissertações da Universidade Federal do Tocantins (BDTD/UFT), devidamente preenchido e assinado, dois exemplares impressos e encadernados em capa dura e dois exemplares em formato digital em CD-ROM com capa, em arquivo PDF da versão definitiva da dissertação. Esses exemplares deverão estar devidamente corrigidos, segundo as sugestões da banca examinadora e assinados pelo orientador e avaliadores. O Senhor Presidente deu a sessão por encerrada e, nada mais havendo a constar, lavra a presente ata que segue assinada por todos os membros da Comissão.

Palmas – TO, 29 de fevereiro de 2016.

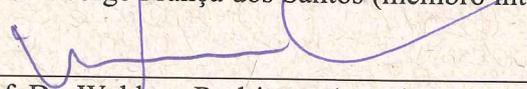
Banca:



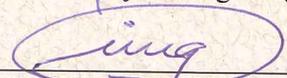
Prof. Dr. Marcelo Lisboa Rocha (orientador)



Prof. Dr. George França dos Santos (membro interno)



Prof. Dr. Waldecy Rodrigues (membro externo)



Prof. Dr. Gerson Pesente Focking (membro externo)

Este trabalho é dedicado a minha esposa Glaucia Sobrinho Santos e meu Filho Guilherme Santos Siqueira, que mim apoiaram e estiveram ao meu lado durante esse período. E aos meus Pais Valdir Soares de Siqueira e Tereza Maria de Azevedo Siqueira por me proporcionar uma base familiar e me incentivar a sempre acreditar e buscar realizar os meus sonhos.

AGRADECIMENTOS

Primeiramente agradeço a Deus, por sempre mim abençoar e mim dar forças para alcançar os meus objetivos.

Agradeço ao orientador Prof. Dr. Marcelo Lisboa Rocha, pelos ensinamentos e paciência durante as orientações, pelo apoio e incentivo durante esse período, por acreditar na minha capacidade, para o desenvolvimento deste trabalho.

Agradeço a minha esposa Glaucia e meu filho Guilherme pelo apoio, compreensão e o amor recebido durante todo esse tempo. Agradeço aos meus pais, Valdir e Tereza pela educação, ensinamentos e amor proporcionado.

Agradeço o apoio do Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, por oportunizar a minha capacitação. E ainda, a todos desta instituição que estiveram envolvidos para isto tornasse possível.

Agradeço a todos professores do Programa de Pós-graduação em Modelagem Computacional e Sistemas da Universidade Federal do Tocantins – UFT, por transmitir seus conhecimentos durante a realização das disciplinas, em especial aos Professores Marcelo Lisboa Rocha, David Nadler Prata, Patrick Letouze, os quais tive a oportunidade de estar mais próximo durante este período.

Agradeço aos Colegas da turma do Mestrado, pela amizade e o compartilhamento de conhecimentos. Principalmente aos amigos Ancelmo Frank, Samuel Costa e Jeane Pamela e Elvis Nascimento os quais dividimos os trabalhos durante a realização das disciplinas.

Ao professor Claudio de Sousa Galvão, pela contribuição durante os estudos das siglas das formulações matemáticas presentes nos artigos estudados.

Ao amigo, Professor Dr. Henrique Fonseca Elias de Oliveira, IFGOIANO campus Ceres, que mim incentivou a estudar.

Aos alunos e amigos Pedro Araújo Cortez e Rafael Pereira Filgueiras, integrantes do Grupo de Desenvolvimento de Sistemas do PIBID do IFTO campus Araguatins, pelas contribuições no desenvolvimento de um módulo do sistema e do Aplicativo de consulta de rotas.

RESUMO

O problema de roteamento do ônibus escolar (PROE), é um importante problema de ordem prática, estudado em otimização combinatória. É formulado através de um conjunto de paradas, frotas de ônibus, escolas e garagem, onde a partir destes conjuntos, busca-se criar rotas otimizadas visando a redução do custo operacional do serviço. Este trabalho apresenta duas grandes contribuições para a melhoria da solução do PROE, sendo elas, o desenvolvimento de um algoritmo baseado na metaheurística GRASP + 2-Opt, para a geração de rotas otimizadas, e um sistema de apoio a decisão para o PROE, com a utilização de funções do Google Maps v3, para proporcionar uma visualização ágil da atual situação do problema para o administrador do sistema, isto, através do uso de marcadores de localizações para paradas de ônibus, escolas e garagem. O sistema foi testado de duas formas. A primeira, com a utilização de instâncias de referência da literatura e a segunda com uma simulação de um ambiente do mundo real. Os resultados são comparados com os principais trabalho da literatura do problema, assim conseguindo gerar soluções com uma redução significativa na quantidade de ônibus utilizados, bem como no tempo de processamento para a geração das rotas.

Palavras-chave: Otimização Combinatória, ônibus escolar, SBRP, Metaheurística GRASP.

ABSTRACT

The school bus routing problem (SBRP) is an important practical problem, studied in combinatorial optimization. It is formulated through a set of stops, bus fleets, schools and garage, where from these sets, we seek to create optimized routes in order to reduce the operating cost of the service. This work presents two great contributions to the improvement of SBRP solution, are the following, the development of an algorithm based on GRASP + 2-Opt, for generating optimal routes and a system decision support for the SBRP, with the use of Google Maps v3 functions, to provide a agile view of the current situation of the problem to the system administrator, through the use of marker locations for bus stops, schools and garage. The system was tested in two ways. First, with the use of benchmark instances the literature and the second with a simulation of a real-world environment. The results are compared with the main work problem literature, thus achieving generate solutions with a significant reduction in the number of buses used and the computational time for generating the route.

Keywords: Combinatorial Optimization, school bus, SBRP, Metaheuristic GRASP.

LISTA DE ILUSTRAÇÕES

Figura 1. Representação do Problema a ser resolvido.....	31
Figura 2. Rota Gerada com o GRASP.....	32
Figura 3. Rota melhorada com Algoritmo 2-opt	33
Figura 4. Avaliação de performance das CPUs utilizadas.	34
Figura 5. Gráfico Comparativo dos resultados computacionais.....	37
Figura 6. Cadastros do sistema com Google Maps v3, recursos AutoComplete e Marker	41
Figura 7. Função Marker do Google Maps v3, localizações de paradas, escolas e garagem...41	
Figura 8. Função Polilines Google Maps v3, visualizador de rotas	42
Figura 9. Aplicativo Android, para visualizar as rotas	43

LISTA DE TABELAS

Tabela 1. Critérios utilizados para a seleção das paradas com GRASP.....	31
Tabela 2. Cálculos utilizados para a seleção das paradas com GRASP	32
Tabela 3. Entrada de dados para instâncias de referências de Park et, al (2012).	36
Tabela 4. Comparação dos resultados computacionais.	37
Tabela 5. Comparação dos resultados computacionais em percentual (%).	38
Tabela 6. Resultados computacionais obtidos com a execução dos algoritmos LBH e GRASP	39

LISTA DE EQUAÇÕES

Equação 1. Fórmula para o cálculo do tempo de percurso	26
Equação 2. Fórmula para o cálculo do tempo de coleta dos estudantes de uma parada.....	27
Equação 3. Fórmula para o cálculo do tempo de entrega dos estudantes na escola.....	27
Equação 4. Fórmula para o cálculo da distância entre duas localizações.....	28
Equação 5. Fórmula para criar a Lista Restrita de Candidatos do GRASP.....	29

LISTA DE ALGORITMOS

Algoritmo 1. Algoritmo de Braca et al., 1997 com adaptação	21
Algoritmo 2. Método construtivo do GRASP aplicado ao PROE.....	24
Algoritmo 3. Matriz de distâncias de LCRi.....	28
Algoritmo 4. Matriz de distâncias de LRCj.....	29
Algoritmo 5. Procedimentos de Busca Adaptativos Gulosos Randomizados.....	29
Algoritmo 6. Algoritmo 2-opt	30

LISTA DE ABREVIATURA E SIGLAS

JSON: JavaScript Object Notation, 41

Km: Quilômetro, 26

LBH: Location Based Heuristic, 19

LCR: Lista Restrita de Candidatos, 25

SBRP: School Bus Routing Problem, 13, 16

VRP: Vehicle Routing Problem, 13, 16

SUMÁRIO

1. INTRODUÇÃO	13
1.1. JUSTIFICATIVA	14
1.2. OBJETIVO GERAL	15
1.3. OBJETIVOS ESPECÍFICOS	15
1.4. ORGANIZAÇÃO DA DISSERTAÇÃO	15
2. REVISÃO DE LITERATURA	16
2.1. DESCRIÇÃO DO PROBLEMA DE ROTEAMENTO DO ÔNIBUS ESCOLAR	16
2.2. O PROBLEMA NP-DIFÍCIL	17
2.3. PROGRAMAÇÃO LINEAR	17
2.3.1. RESOLUÇÃO DO PROE ATRAVÉS DO MÉTODO EXATO.....	17
2.3.2. MÉTODO HÍBRIDO.....	18
2.4. O ATENDIMENTO EM ZONAS RURAIS E URBANA	19
2.4.1. O PROE PARA ZONAS URBANAS.....	19
2.4.2. O PROE PARA ZONAS RURAIS.....	20
2.5. OS TIPOS DE FROTAS	20
2.6. O ALGORITMO DE BRACA	21
3. METODOLOGIA UTILIZADA PARA RESOLVER O PROE	23
3.1. A METAHEURÍSTICA GRASP	23
3.1.1. O ALGORITMO CONSTRUTIVO.....	24
3.1.2. FASE DE CONSTRUÇÃO DA LISTA RESTRITA DE CANDIDATOS.....	27
3.1.3. MELHORIA PARA A SOLUÇÃO INICIAL COM ALGORITMO 2-OPT.....	30
3.2. APRESENTAÇÃO DO GRASP APLICADO AO PROE	30
4. ANÁLISE E DISCUSSÃO DOS RESULTADOS	34
4.2. SIMULAÇÃO PARA UM PROBLEMA DO MUNDO REAL	39
4.3. O SISTEMA DE APOIO A DECISÃO PARA O PROE	40
4.3.1. ENTRADAS DE DADOS.....	40
4.3.2. MAPEAMENTO E VISUALIZAÇÕES DO AMBIENTE.....	41
4.3.3. VISUALIZADOR DE ROTAS.....	42
4.3.4. APLICATIVO DE VISUALIZAÇÃO DE ROTAS.....	42
5. CONCLUSÃO E TRABALHO FUTUROS	44
5.1. CONCLUSÃO	44
5.2. TRABALHOS FUTUROS	45
REFERÊNCIAS BIBLIOGRÁFICAS	46

ANEXOS	49
APÊNDICE A – ARTIGO PUBLICADO	50

1. INTRODUÇÃO

O problema de Roteamento do Ônibus Escolar (PROE), do termo em inglês (School Bus Routing Problem - SBRP) é um importante problema de otimização combinatória que foi modelado por Newton e Thomas (1969). O PROE é uma Variação do Problema de Roteamento de Veículos (PRV) do termo inglês Vehicle Routing Problem (VRP), que é comumente tratado na pesquisa operacional.

O problema pode ser enunciado da seguinte maneira: Dado um conjunto de estudantes, escolas e garagem são definidas as paradas de ônibus, que devem estar localizadas próximo ou na porta da residência dos estudantes, entretanto, a definição da parada de ônibus está sujeita a uma restrição de distância, onde, os estudantes devem andar até chegar a mesma. Cada estudante deve estar associado a uma parada. Com estas informações definidas, podem ser utilizados métodos de Programação Linear Inteira, métodos heurísticos ou métodos híbridos para criar a solução de rotas. Os ônibus devem percorrer todas paradas de ônibus, com o objetivo de coletar todos os estudantes associados, logo em seguida, eles são transportados e entregues em suas escolas de destino. No final do dia letivo, o estudante é transportado novamente para o mesmo local onde foi coletado.

Segundo Bowerman et al., (1995), Teng e Yang (2008), Martínez e Viegas (2011); Park et al., (2012), Riera-Ledesma e Salazar-González (2012) o PROE é um problema que tem um enunciado fácil de entender porém difícil de resolver. Esses autores classificam o PROE como um problema NP-Difícil.

Por meio de uma revisão de literatura sobre este problema nos principais periódicos foi verificado que os números de publicações disponíveis sobre o tema é relativamente pequeno, quando é comparado com a quantidade de trabalhos que apresentam métodos para resolver o clássico problema de roteamento de veículos (PRV).

Na literatura, foram encontrados alguns estudos de caso do PROE com aplicação em cidades brasileiras. Souza e Siqueira (2010) apresentaram uma proposta de um método heurístico para resolver o problema no estado do Paraná. Os autores realizaram um estudo de caso, considerando atendimento para estudantes residentes em zonas rurais e urbanas. Faraj et al., (2013) realizaram um estudo de caso do problema do transporte escolar em zonas rurais no estado de Minas Gerais. Steiner e Zamboni (2000) desenvolveram uma solução para o transporte escolar de áreas urbanas da cidade de Curitiba, Paraná. Mandujano et al., (2012) apresentaram metodologias para as localizações das escolas e para otimizar o transporte escolar

em áreas rurais. A metodologia foi baseada em dois modelos de programação linear inteira mista também conhecida como hibridismo. O primeiro trata das localizações das escolas e os problemas de dimensionamento e o segundo, trata do roteamento de ônibus escolar. O estudo de caso foi realizado na cidade de Barão do Grajaú, Maranhão.

O objetivo deste trabalho é criar um algoritmo capaz de gerar uma solução com rotas otimizadas, depois, implementa-lo em um sistema de apoio a decisão para o PROE. Pretende-se que o sistema seja capaz de gerar rotas de ônibus para o atendimento de estudantes residentes em zonas rurais e urbanas. Outro ponto importante é que seja considerado que a frota de ônibus pode ser heterogênea.

Entretanto, fazendo o uso do sistema de apoio a decisão para o PROE, o administrador do sistema conseguirá ter uma visualização rápida da situação real de localização dos endereços dos estudantes, paradas de ônibus, escolas, garagens e a geração de rotas otimizadas para o transporte escolar e ainda verificar a viabilidade de cada rota criada, através do visualizador de rotas do sistema, que também, está disponível em um aplicativo para smartphone com sistema operacional Google Android. As rotas de ônibus são geradas no início do ano ou no início do semestre letivo.

Os resultados computacionais comprovam a eficiência do algoritmo da metaheurística GRASP, aplicada ao PROE comparados com dados de instâncias de referências da literatura.

1.1. JUSTIFICATIVA

No Brasil, o transporte escolar é financiado e ofertado pelo Ministério da Educação em parceria com os Estados e Municípios. É regulamentado pela lei 9394/96, com acréscimo na lei 10.709/2003. Um número considerável de estudantes das redes estaduais e municipais necessitam do transporte escolar para ter acesso a escola, principalmente os residentes na zona rural.

Com o desenvolvimento do sistema de apoio a decisão para o PROE, será possível gerenciar informações de usuários, paradas de ônibus, frotas, garagem e também gerar rotas otimizadas para o transporte escolar. Contudo o sistema pode contribuir de forma significativa para tomada de decisão. Visto que a otimização dos custos com o serviço, proporciona a redução de gastos do dinheiro público.

1.2. OBJETIVO GERAL

O objetivo geral deste trabalho é criar um algoritmo capaz de gerar rotas de ônibus otimizadas utilizando a metaheurística GRASP, afim de desenvolver um sistema de apoio a decisão para o problema de roteamento do ônibus escolar.

1.3. OBJETIVOS ESPECÍFICOS

- Realizar uma revisão de literatura do PROE.
- Desenvolver um algoritmo para o PROE, aplicando a metaheurística GRASP.
- Desenvolver um sistema de apoio a decisão
- Realizar testes e simulações computacionais com a metaheurística GRASP comparando-a com os principais algoritmos de referência para o problema, afim de comprovar a sua eficiência.

1.4. ORGANIZAÇÃO DA DISSERTAÇÃO

Este trabalho está organizado da seguinte forma:

- Capítulo 1: apresenta uma introdução sobre o problema de roteamento do ônibus escolar, justificativa e os objetivos do trabalho.
- Capítulo 2: apresenta uma revisão de literatura para o trabalho proposto.
- Capítulo 3: Descreve a metodologia de desenvolvimento do algoritmo baseado na metaheurística GRASP aplicada ao PROE e uma simulação de funcionamento.
- Capítulo 4: são apresentados os resultados dos testes computacionais comparados com os principais trabalhos de referência da literatura e o sistema de apoio a decisão.
- Capítulo 5: são apresentados, a conclusão e trabalhos futuros.

2. REVISÃO DE LITERATURA

2.1. DESCRIÇÃO DO PROBLEMA DE ROTEAMENTO DO ÔNIBUS ESCOLAR

O problema de Roteamento do Ônibus Escolar (PROE), do termo em inglês (School Bus Routing Problem - SBRP) é um problema de otimização combinatória que foi modelado por Newton e Thomas (1969). O PROE é uma Variação do Problema de Roteamento de Veículos (PRV) do termo inglês Vehicle Routing Problem (VRP), que é comumente tratado na pesquisa operacional.

O problema pode ser enunciado da seguinte maneira: Dado um conjunto de estudantes, escolas e garagem são definidas as paradas de ônibus, que devem estar localizadas próximo ou na porta da residência dos estudantes, entretanto, a definição da parada de ônibus está sujeita a uma restrição de distância, onde, os estudantes devem andar até chegar a mesma. Cada estudante deve estar associado a uma parada. Com estas informações definidas, podem ser utilizados métodos de Programação Linear Inteira, métodos heurísticos ou métodos híbridos para criar a solução de rotas. Os ônibus devem percorrer todas paradas de ônibus, com o objetivo de coletar todos os estudantes associados, logo em seguida, eles são transportados e entregues em suas escolas de destino. No final do dia letivo, o estudante é transportado novamente para o mesmo local onde foi coletado.

De acordo com Desrosiers et al. (1981) o PROE está dividido em cinco etapas: preparação dos dados, seleção das paradas, geração das rotas, ajuste das janelas de tempo e escalonamento das rotas.

Preparação de dados. Neste sub-problema é especificado, a rede de estradas e são preparados quatro tipos de entradas de dados para PROE: estudantes, escolas, veículos e matriz de distâncias . Os dados para os estudantes incluem a localização (endereço) de suas casas, a escola de destino e o tipo de aluno. O tipo de aluno é geral ou deficientes. (PARK; KIM, 2010).

Seleção de paradas. Esta etapa procura selecionar um conjunto de paradas de ônibus e fazer alocação dos estudantes às mesmas. No entanto, há restrições em relação à distância máxima que os estudantes devem caminhar de sua casa até ponto de ônibus mais próximo. (PARK; KIM, 2010).

Geração das Rotas. Para FARAJ et, al. (2013) a etapa mais significativa para o PROE é a geração das rotas, onde os dados oriundos das etapas anteriores são processados a fim de serem construídas as melhores rotas possíveis.

Ajuste das janelas de tempo. De acordo com Park e Kim (2010), na maioria dos estudos, o horário inicial e final das escolas são considerados como restrições. No entanto, há uma série de trabalhos que consideram o horário como uma variável de decisão e as tentativas de encontrar ótimos horários para iniciar e terminar, pode maximizar o número de rotas atendidas sequencialmente por um único ônibus, desta forma reduzir o número de ônibus utilizados.

Escalonamento de Rotas. Newton e Thomas (1969), desenvolveram um modelo multi-escola para determinar todas as rotas de ônibus para uma escola. Eles assumiram que existem períodos de tempo distintos e as escolas começam em diferentes períodos. Desta forma, criaram uma cadeia de rotas que podem ser sequencialmente atendida pelo mesmo ônibus.

2.2. O PROBLEMA NP-DIFÍCIL

Segundo Bowerman et al., (1995), Teng e Yang (2008), Martínez e Viegas (2011); Park et al., (2012), Riera-Ledesma e Salazar-González (2012), o PROE é um problema que tem um enunciado fácil de entender porém difícil de resolver. E o classificam como um problema NP-Difícil, ou seja, são problemas para os quais não existem algoritmos que os resolvam de forma exata em tempo polinomial.

Bowerman et al., (1995), enfatizaram que o problema de combinações de seleção de paradas e geração de rota de ônibus para uma única escola é um problema NP-difícil. Na seleção paradas de ônibus, cada aluno deve ser associado a uma parada de ônibus e cada parada tem uma quantidade de estudantes que devem ser coletados. Usando essas restrições, o problema de seleção de paradas de ônibus pode ser transformado em um problema generalizado de atribuição, que também é conhecido como um problema NP-Difícil (FISHER, 1986). O problema de geração de rota de ônibus com restrições de capacidade veículo e de tempo máximo de percurso corresponde as restrições de capacidade e distância do PRV aberto, que é conhecido como um problema NP-hard (BEKTAS; ELMASTAS, 2006).

2.3. PROGRAMAÇÃO LINEAR

2.3.1. RESOLUÇÃO DO PROE ATRAVÉS DO MÉTODO EXATO

O método exato é utilizado para a resolver problemas de otimização combinatória, contudo, na literatura sobre o PROE é aplicado apenas em problemas relativamente pequenos, assim, destaca-se o seu elevado nível de complexidade para resolução do problema e seus

subproblemas. A aplicação do método exato ao PROE são apresentados em Schittekat et al., (2006), Fügenschuh (2009) e Chen et al., (2015).

Schittekat et al., (2006) criaram uma formulação matemática para o PROE, utilizando programação linear inteira, com os resultados dos testes computacionais, comprovaram que com o uso do método exato, pode-se resolver apenas problemas de pequeno porte, desta forma, mostraram a resolução do problema para um quantitativo de 50 estudantes e 10 paradas.

Fügenschuh (2009) apresentou uma formulação de programação linear inteira para otimizar a integração do horário de início escolar e o transporte público. O desenvolvimento deste modelo foi impulsionado pela demanda do mundo real em áreas rurais dos municípios, onde os estudantes são os maiores grupos de usuários e a maioria das escolas começam atualmente no mesmo horário. Fügenschuh (2009) utilizou técnicas de branch-and-cut.

Para Chen et al., (2015) a formulação utilizando programação linear inteira baseado em problema de frota de ônibus do tipo heterogênea é mais útil na resolução de problemas do mundo real. Tal formulação pode reduzir a complexidade do modelo em termos do número de variáveis de decisão, número de restrições e o tamanho do arquivo de modelo.

2.3.2. MÉTODO HÍBRIDO

Nos últimos anos, muitas heurísticas e metaheurísticas têm sido propostas para problemas de otimização combinatória como Tabu Search (Busca Tabu), Genetic Algorithms (Algoritmo Genético), Ant Colony Optimization (Otimização de Colônia de Formigas) e GRASP - Greedy Randomized Adaptive Search Procedure (Procedimentos de Busca Adaptativos Gulosos Randomizados).

Sghaier et al., (2013) desenvolveram uma solução para o PROE, através do desenvolvimento de um novo operador para o algoritmo Genético, para a geração de rotas. Para comprovar a sua funcionalidade, realizou uma comparação da sua proposta com mais quatro Tipos de operadores genéticos disponível na literatura.

Pacheco et al., (2012) desenvolveram uma solução para o PROE com a aplicação da metaheurística Busca Tabu, por meio da programação adaptativa de memória multiobjetivo, e a comparou com uma implementação do algoritmo genético de seleção não-dominado, que é uma abordagem muito conhecida na otimização multiobjetivo.

Huo et al., (2014) utilizaram uma estratégia de roteamento localização-alocação que foi utilizado para coletar os dados dos usuários, a partir dela é realizada a análise de distribuição

de endereços, seleção de paradas e geração da rota. Então, o problema é resolvido pelo algoritmo de otimização colônia de formigas. Os resultados enfatizam que o modelo proposto é aplicável a uma situação do mundo real.

Schittekat et al., (2013) utilizaram a metaheurística GRASP + VND, aplicado ao PROE, com o objetivo de resolver o problema de modo iterativo, melhorando assim a solução em cada iteração consecutiva. Desta forma, obtiveram bons resultados para uma quantidade de 80 paradas de ônibus e 800 estudantes.

2.4. O ATENDIMENTO EM ZONAS RURAIS E URBANA

Durante a revisão de literatura, foram identificados dois tipos de classificação de prestação de serviço em zonas rurais e zonas urbanas.

2.4.1. O PROE PARA ZONAS URBANAS

Soluções e estudos de caso para o PROE em zonas urbanas são frequentemente encontradas na literatura. Bowerman et al., (1995) apresentaram uma solução heurística para a cidade Wellington no Canada. Braca et al., (1997) implementaram o algoritmo LBH (Location Based Heuristic) que utiliza um algoritmo de Carga Mista, isto quer dizer que, os ônibus podem coletar os estudantes de escolas distintas, associados a uma parada de ônibus. A cada rota criada, um ônibus pode atender escolas diferentes. Durante o percurso o ônibus visita cada parada, coleta os estudantes e os transporta para suas respectivas escolas. No final do dia letivo deixa-os nos mesmos locais onde foram coletados. Este trabalho foi aplicado a um estudo de caso na Cidade de Nova York nos Estados Unidos.

Li e Fu (2002) desenvolveram um algoritmo multi-objetivo para o PROE e realizaram um estudo de caso em uma escola de jardim de infância na cidade de Hong Kong na China. Euchí e Mraihí (2012) implementaram o algoritmo de colônia de formigas, e realizaram um estudo de caso na Tunísia. Fulin e Yueguang (2012) realizaram melhorias no algoritmo de Exame de Partículas Quântico e o aplicou ao PROE, para otimizar as rotas de ônibus da Escola de Nacionalidade de Gansu na China. Arias-Rojas et al., (2012) desenvolveram um sistema de roteamento de ônibus escolar para a cidade de Bogotá na Colômbia.

Alguns autores desenvolveram soluções capazes de resolver o PROE, com atendimento em zona rural e zona urbana. Souza e Siqueira (2010) criaram um sistema para a otimização do transporte escolar no estado do Paraná, Brasil. Prasetyo et al., (2011) criou um sistema de apoio a tomada de decisão para a secretaria de educação da cidade de Surabaya na Indonésia, com

objetivo de criar rotas otimizadas e assim conseguir atender mais pessoas carentes que necessitam do transporte público. Bögl et al., (2006) implementaram uma proposta para o PROE com trocas de ônibus ou transferências, neste caso, os estudantes fazem trocas de ônibus durante o percurso até a escola. E enfatizou que o serviço ofertado diminui a qualidade, porém, destaca que a economia no custo total com transporte escolar é significativa. Ellegood et al., (2015) utilizaram o método de Carga Mista para resolver o problema na cidade de Missouri, Estados Unidos.

2.4.2. O PROE PARA ZONAS RURAIS

Estudos específicos para zonas rurais sobre o PROE, são realizados por pesquisadores de diversas partes do mundo. Fügenschuh et al., (2005) apresentaram uma formulação matemática para o PROE para atender as áreas rurais da Alemanha. Fügenschuh (2009) realizou um estudo de caso e os resultados obtidos mostraram que, pode-se economizar cerca de 1 milhão de Euros, em um período de um ano com a otimização das rotas do ônibus escolar. Carvalho et al., (2010) publicaram um estudo de caso sobre as condições do transporte escolar em zonas rurais do Brasil. Apontaram as condições das estradas e as dificuldades de acesso à escola dos estudantes que residem na zona rural. Bronshtein et al., (2014) apresentaram uma proposta para resolver o transporte escolar de um grupo de estudantes, residentes em fazendas na Rússia. Castillo-López e López-Ospina (2015) afirmaram que as condições geográficas e socioeconômicas das zonas rurais no Chile, causam ineficiência na qualidade de desempenho de aproveitamento acadêmico dos estudantes. Mandujano et al., (2012) apresentaram uma metodologia para otimizar a localização dos estudantes que necessitam do transporte escolar em áreas rurais. A metodologia é baseada em dois modelos de programação linear inteira mista, sendo que o primeiro trata do problema de localização das escolas e paradas e o segundo trata do roteamento do ônibus escolar. O estudo de caso foi realizada na cidade de Barão do Grajaú, Maranhão.

2.5. OS TIPOS DE FROTAS

Na literatura são encontrados dois tipos de frotas de ônibus sendo elas: homogêneas e heterogêneas.

As frotas são consideradas homogêneas, quando cada ônibus possui a mesma quantidade de assento. Os seguintes autores: Bögl et al., (2006), Euchie e Mraih (2012); Kim

et al., (2012); Park et al., (2012); Song e Kim, (2013), resolveram o PROE utilizando apenas frotas homogêneas através de métodos exatos, híbridos e heurísticos.

Para tipos de frotas heterogêneas, os ônibus possuem capacidades diferentes. Foi identificado que, Arias-Rojas et al., (2012); Pacheco et al., (2012); Díaz-Parra et al., (2013); Faraj, et. al., (2013), Chen et al., (2015), aplicaram métodos heurísticos para resolver o problema com este tipo de frota.

2.6. O ALGORITMO DE BRACA

Para comparar a viabilidade da metaheurística proposta por este trabalho, foi implementado o algoritmo de Braca et al., (1997), na qual, é utilizado o algoritmo heurístico baseado em localização (LBH) do termo em inglês (Location Based Heuristic). No estudo original foi utilizado uma frota de ônibus escolar homogênea e com capacidade para 66 estudantes. Pequenas adaptações foram necessárias para a implementação. O algoritmo é mostrado no Algoritmo 1.

Algoritmo 01. Algoritmo de Braca et al., (1997)

```

1:  $P = \{1, 2, \dots, n\}$  Conjunto de Paradas não Visitadas
2:  $R = \{ \emptyset \}$  Variável vazia, que receberá o conjunto de rotas
3: ENQUANTO ( $P \neq \emptyset$ ) FAÇA{
4:      $j =$  Uma parada sorteada aleatoriamente do conjunto  $P$ ;
5:      $P = P$  //remove a parada  $j$  do conjunto  $P$ 
6:      $R' = A_{escola}(j)$  destinada a parada  $j$ 
7:     REPITA{
8:          $ci = tamanhoRota(i, R')$ ,  $i \in P$ 
9:         LET  $ck = \min_{i \in P} \{ci\}$ 
10:        IF  $ck < +\infty$  THEN{
11:             $R' = R' + \{k\}$  //insere a para  $k$  em  $R'$ 
12:             $P = P$  // remove a parada  $k$  do conjunto  $P$ 
13:        }
14:    } ATÉ  $ck = +\infty$ 
15:    SE (número de estudantes de  $R' < 11$ ) ENTÃO {
16:         $movaParada(R, R')$ 
17:         $R = R + R'$ 
18:    }
19: }
20: Mostra da Solução  $R$ 

```

Algoritmo 1. Algoritmo de Braca et al., 1997 com adaptação

A função $tamanhoRota(i, R')$, tem a finalidade de calcular a distância da parada j até cada uma das paradas i , com destino a $escola(i)$, feito isto, retorna a parada i que possuir o menor custo de inserção, logo, ela será inserida na rota R' . Em cada iteração do loop da (linha

7) é adicionado um parada i com o menor custo de inserção na rota R' . Portanto, não é inserido uma parada i qualquer, e sim, a que possui o menor custo de inserção e esteja associada com a escola i . Se todas as soluções geradas são inviáveis então retorna $+\infty$.

A função $movaParada(\mathbf{R}, \mathbf{R}')$, verifica se o número de estudantes atendidos na rota \mathbf{R}' é menor que 11, caso não exista mais paradas a serem visitadas, ele entrará em conflito com uma restrição que não pode ser evitada, então a função verifica se em alguma rota \mathbf{R}' , há assentos suficientes para incluir os estudantes das paradas, caso, seja encontrado uma rota viável capaz de inclui-los, isso é realizado, senão a execução do algoritmo é abortada e todas as paradas visitadas são retornadas para o conjunto de paradas \mathbf{P} não visitadas. Em seguida inicia-se o loop inicial e o processo iterativo para a criação das rotas é realizado novamente com o sorteio aleatório de uma nova parada j , o loop continua até que a solução seja encontrada. Park et al., (2012) implementou este algoritmo e constatou em seus testes computacionais que dependendo do número de instâncias, não haverá uma solução viável para o problema. No entanto, ele ignorou a função $movaParada(\mathbf{R}, \mathbf{R}')$ em sua implementação. Park et al., (2012) ainda considerou que essa função é trivial.

3. METODOLOGIA UTILIZADA PARA RESOLVER O PROE

Para resolver o Problema de Roteamento do Ônibus Escolar, foram realizados estudos dos principais trabalhos da literatura. No entanto, serão aprofundados os estudos em apenas dois trabalhos de referência da literatura, sendo eles: Braca et al., (1997) e Park et al., (2012). O critério utilizado para aprofundar nestes dois artigos, foram os seguintes: número de citações em outros trabalhos, número de paradas e alunos atendidos, instâncias de referências para testes, eficiência na solução e melhores resultados da literatura.

As regressões lineares propostas por Braca et al., (1997), foram utilizadas para a criação das seguintes funções: tempo necessário para a coleta dos estudantes e o tempo necessário para a entrega dos estudantes em sua escola de destino. Mais detalhes podem ser vistos na seção 3.1.1.

Park et al., (2012), apresentaram melhorias para o método Mix Load (Carga Mista), onde, seus resultados computacionais são comparados com os métodos Single Load (Carga Única) e o algoritmo de Braca et al., (1997). Os dados de entrada (paradas, escolas e garagem e ônibus) estão disponíveis em http://logistics.postech.ac.kr/Mixed_SBRP_Benchmark.html, e também os resultados computacionais das instâncias com a solução das rotas criadas. As instâncias são utilizadas como fonte de referência, para a comparação dos resultados computacionais da presente proposta, com outras técnicas relevantes para o PROE.

3.1. A METAHEURÍSTICA GRASP

O GRASP (Greedy Randomized Adaptive Search Procedure) que significa em português Procedimentos de Busca Adaptativos Gulosos Randomizados, foi desenvolvida por Feo e Resende (1995). É um procedimento iterativo, onde cada iteração consiste em uma fase de construção de uma solução inicial de forma adaptativa gulosa aleatorizada, combinada com uma aplicação de uma heurística de melhoramento, tipicamente um procedimento de busca local. A melhor solução dentre todas as iterações será então o resultado final (FEO e RESENDE, 1995). O pseudocódigo do algoritmo construtivo do GRASP aplicado ao PROE é apresentado no Algoritmo 2.

3.1.1. O ALGORITMO CONSTRUTIVO

O algoritmo construtivo do PROE com a aplicação da metaheurística GRASP é mostrado no Algoritmo 2.

Cada conjunto de dados corresponde a uma tabela do banco de dados, sendo que a mesma possui atributos e associações.

Algoritmo 2. Método Construtivo para a Metaheurística GRASP

```
1:  $P = \{1, 2, \dots, p\}$  //Conjunto de informações das Paradas não Visitadas
2:  $O = \{1, 2, \dots, o\}$  //Conjunto de Ônibus
3:  $G = \{1, 2, \dots, g\}$  //Conjunto de Depósito/Garagem
4:  $E = \{1, 2, \dots, e\}$  //Conjunto de Escolas
5:  $R = \{\emptyset\}$  //Variável vazia, que receberá o conjunto de rotas
6: ENQUANTO ( $P \neq \emptyset$ ) FAÇA{
7:     SE ( $R == \emptyset$ ) ou ( $\text{horarioDeAtendimentoOnibus} > \text{horárioDeInicioDiaEscolar}$ ){
8:          $b =$  Sorteio aleatório de um Ônibus  $o$  do conjunto  $O$  pertence a  $G$ ;
9:     } SENÃO {
10:        Gerar uma nova rota para o ônibus;
11:    }
12:     $ci = \text{LRCi}(\alpha, ge, P)$ ; // Lista Restrita de Candidatos (LRC), com todas as paradas
13:     $pi =$  Uma parada sorteada aleatoriamente do conjunto LRCi de  $ci$  pertence a  $P$ ;
14:     $R' = pi, escola(i)$ ; //escola  $e$  destinada a parada  $i$ ;
15:     $P = P$ ; //remove a parada  $i$  do conjunto  $P$ ;
16:    ENQUANTO ( $\text{capacidade do ônibus } b \leq \text{lotação}$ ) FAÇA{
17:         $cj = \text{LRCj}(\alpha, escola(i), pi, P)$ ; //LCR, algoritmo GRASP
18:         $pj =$  Uma parada sorteada aleatoriamente do conjunto de  $cj$  pertence a  $P$ ;
19:        SE ( $\text{HorarioDeChegada} \leq \text{horarioDoInicioDaAula}$ ){
20:             $R' = j, escola(i)$ ;
21:             $P = P$  //remove a parada  $j$  do conjunto  $P$ ;
22:        }
23:        SENÃO {
24:            Para a execução e o ônibus vai para  $escola(i)$ ;
25:        }
26:    }
27:    descruzamentoDeRota( $R'$ ); //algoritmo 2-opt
28:     $R = R + R'$ 
29: }
30: Mostra da Solução  $R$ 
```

Algoritmo 2. Método construtivo do GRASP aplicado ao PROE

As seguintes fases são necessárias para iniciar o método construtivo desta proposta:

- Fase 1, é selecionado um parâmetro (Urbano ou Rural), respectivamente representado pelos números binários (0 e 1).
- Fase 2, de acordo com o parâmetro passado, os conjuntos são filtrados no Banco de dados e as variáveis são carregadas em estruturas de dados na Memória RAM.

- Fase 3, é iniciado a execução do algoritmo.

Na fase 2, os seguintes conjuntos de entradas de dados são obtidos, onde, P é Conjunto de Paradas, O é o Conjunto de Ônibus, G é o conjunto de Garagem, E é o Conjunto de Escolas e R é o conjunto de rotas iniciado vazio.

Na fase 3, A sequência de execução do algoritmo é dividida em 8 etapas, são as seguintes:

Etapa 1, é iniciado o loop de iterações do método construtivo, logo, é verificado uma condicional, se o conjunto da solução R' é igual a vazio ou se o horário de atendimento do ônibus é maior que o horário de início da aula, caso a condicional seja verdadeira é sorteado de forma aleatória um ônibus o do Conjunto O . Se o horário de atendimento do ônibus for menor que o horário de início da aula, o mesmo ônibus será utilizado para atender uma nova rota. O ônibus possui restrição de capacidade, onde não é permitido que o seu limite seja ultrapassado. Quando a restrição de horário de atendimento do ônibus for satisfeito, o ônibus será retirado do conjunto O .

Etapa 2, é executado a função $LRCi(\alpha, ge, P)$, que é responsável por calcular e criar uma matriz de custos com as distâncias do local de saída do ônibus, que pode ser uma Garagem G ou uma Escola E , com destino a todas as paradas não visitadas $p \in P$, (ver no Algoritmo 3), Após criar a matriz de custos é executado a função $criarLRC(\text{matrizCustos})$, que por sua vez utiliza critérios de seleção para criar uma Lista Restrita de Candidatos (LRC) do GRASP, logo, a função retorna um subconjunto com as melhores paradas $p \in P$, (veja no Algoritmo 5) todos os detalhes são mostrados na sessão 3.1.2

Etapa 3, ci recebe a LRC retornada pela função $LRCi(\alpha, ge, P)$, onde é escolhido de forma aleatória uma parada $pi \subset ci / ci \in P$, logo, a parada pi e a sua $escola(i)$ associada, são inseridos na solução da rota R' , assim, inicia a construção de uma rota, depois de ser incluída em R' , a parada pi é removida do conjunto P .

Etapa 4, enquanto a capacidade do ônibus, for maior que a quantidade de estudantes coletados, uma nova parada é visitada. Um procedimento iterativo é realizado no loop da (linha 16) no Algoritmo 2, desta forma, cj recebe a LRC da função $LRCj(\alpha, escola(i), pi, P)$, de maneira semelhante ao passo 2. Uma vez que são passados dois novos paramentos: a parada pi e a $escola(i)$, assim, a função $LRCj(\alpha, escola(i), pi, P)$ é responsável por calcular e criar uma matriz de custos entre a parada pi até todas as paradas $p \in P$ que tem como destino a $escola(i)$, (ver no Algoritmo 4), após criar a matriz de custos, a função a $criarLCR(\text{matrizCustos})$ usa os

critérios da LRC do GRASP, para criar um conjunto das melhores paradas $p \in P$, (veja no Algoritmo 5). Todos os detalhes são mostrados na sessão 3.1.2.

Etapa 5, c_j recebe a LRC da função $LRC_j(\alpha, p_i, escola(i), P)$ e é escolhido de forma aleatória uma parada $p_j \subset c_j / c_j \in P$ e a sua $escola(i)$ que está associada. Antes de inserir a parada p_j ao conjunto R' , então, é verificado se o tempo do percurso da parada p_j até a $escola(i)$ não infringem a restrição de que o ônibus deve chegar na escola antes da horário de início da aula. A seguinte fórmula foi utilizada para validar o tempo do percurso:

$$Tempo (s) = (Distância / Velocidade) * 60 * 60$$

Equação 1. Fórmula para o cálculo do tempo de percurso

onde a *Distância* é o valor retornado da matriz de custos, e para a *Velocidade* é considerado que o ônibus seja dirigido a uma velocidade média de 32 Km por hora. Adaptação baseada no trabalho de Braca et al., (1997), que é equivalente a 20 milhas por hora.

Então a variável *Tempo* é somada com o horário de chegada na parada p_j . Se a condicional da etapa 6 for verdadeira, a parada p_j é inserida na rota R' . Depois de ser incluída em R' , a parada p_j é removida do conjunto P . Ressalta-se que esta fase está sujeito a verificação das restrições da etapa 6.

Etapa 6, se o horário de chegada do ônibus na escola for menor que horário de início da aula. A cada iteração da etapa 4 é adicionado uma parada na rota R' , até que a restrição de capacidade do ônibus seja atendida.

A cada iteração para a criação de uma rota R' , os seguintes atributos são inseridos: o tempo do percurso entre a parada i até j (tempo em segundos), a quantidade de estudantes coletados em cada parada, o tempo de espera para coletar todos os estudantes de uma parada, (tempo em segundos), a hora de chegada na parada e a distância percorrida em quilômetros (Km).

Para calcular o tempo de espera do ônibus, durante a coleta dos estudantes foi utilizado uma regressão linear proposta por Braca et al., (1997), para instâncias de referências.

Onde S_i é o tempo de espera em segundos e n_i é a quantidade de estudantes que serão coletados em cada parada.

$$Si = 1.9 + 2.6ni$$

Equação 2. Fórmula para o cálculo do tempo de coleta dos estudantes de uma parada.

Também foi utilizado outra regressão linear proposta por Braca et al., (1997) , para calcular o tempo de espera do desembarque dos estudantes na escola, a seguinte formulação é apresentada:

$$Sn + i = 2.9 + 1.9ni$$

Equação 3. Fórmula para o cálculo do tempo de entrega dos estudantes na escola.

Onde $Sn + i$ é o tempo de espera em segundos e ni é a quantidade total de estudantes que embarcaram no ônibus.

No final de cada rota, além das informações supracitadas são adicionados as seguintes informações: a escola de destino, a somatória dos estudantes, o horário de chegada na escola e o tempo de espera para o desembarque dos estudantes na escola.

O método iterativo continua até que a restrição de capacidade do ônibus ou a restrição do horário de início do dia letivo sejam atendidos. Se as restrições de horário de chegada na escola ou a capacidade do veículo, não forem satisfeitos, o ônibus seguirá com destino a escola e deixará os estudantes.

Etapa 7, é utilizado a função ***descruzamentoDeRota(R')***, onde foi implementado o algoritmo 2-opt, que tem a finalidade de descruzar as rotas, assim diminuir a distância da solução ***R'***. Os detalhes são mostrados na seção 3.1.3.

Etapa 8, mostra a solução de todas as rotas construídas, $R = \{R_1, R_2, \dots, R_n\}$.

3.1.2. FASE DE CONSTRUÇÃO DA LISTA RESTRITA DE CANDIDATOS

Na fase de construção é gerado uma solução viável para o problema através de um procedimento parcialmente guloso e parcialmente aleatório. Onde o parâmetro α tem a função de determinar o tamanho da Lista Restrita de Candidatos (LRC). O α é um valor que varia de 0 a 1, onde o $\alpha=0$, determina que o método será totalmente guloso e o $\alpha=1$, determina que o método será totalmente aleatório. Este parâmetro é responsável por determinar o tamanho da

LRC, tal que, quanto mais próximo de 1, for o valor de α , maior será o número de paradas, inseridos ao subconjunto e pior será a solução inicial para o problema.

Dados os parâmetros iniciais das funções $LRC_i(\alpha, g, P)$ e $LRC_j(\alpha, p_i, escola(i), P)$, uma matriz de custos é construída.

Onde variáveis recebem os seguintes parâmetros:

- α pode assumir um valor de 0 até 1;
- g é o local de saída do ônibus, que pode ser uma garagem ou uma escola;
- P é o conjunto de paradas não visitadas;
- p_i é a primeira parada que o ônibus deve visitar ao sair do local de saída definido em g ;
- $escola(i)$ é a escola associada a primeira parada visitada, feito isto somente serão incluídos na rota R' paradas que tem a mesma escola como destino.

O algoritmo 3, é a função $LRC_i(\alpha, g, P)$, sua finalidade é criar uma matriz de custos de inserção, para iniciar a construção de uma rota R' a partir de P . Portanto, são calculados os custos do percurso da Garagem g ou Escola e até cada parada p não visitada, somado com a distância da parada p até a escola e associada a ela. O custo é formulado por:

$$Custo = distancia(C_i, C_j) + distancia(C_j, C_k)$$

Equação 4. Fórmula para o cálculo da distância entre duas localizações

Algoritmo 3. Lista Restrita de Candidatos Inicial (LRCi) - Considera todas as Paradas

```

1: FUNÇÃO LRCi( $\alpha, g, P$ ){
2:   ENQUANTO ( $P \neq \emptyset$ ) FAÇA {
3:      $p_i = P[n]$ ;
4:      $e = escola(i) \subset P[n] \in E$ ;
5:      $custo = distancia(g, p_i) + distancia(p_i, e)$ ;
6:      $matrizCustos = matrizCustos + custo$ ;
7:   }
8:   criarLRC( $matrizCustos$ );
9:   RETORNA LRC;
10: }
```

Algoritmo 3. Matriz de distâncias de LCRi

Após criar a matriz de custos, a função $criaLRC(matrizCustos)$ é chamada para identificar as melhores potenciais paradas p , que serão inseridas na rota R' .

Algoritmo 4. Lista Restrita de Candidatos por escola (LRCj) – Considera apenas as Paradas da *escola(i)*

```

1: FUNÇÃO LRCj ( $\alpha$ ,  $p_i$ , escola(i),  $P$ ) {
2:   ENQUANTO ( $P \neq \emptyset$ ) FAÇA {
3:      $p_j = P[n]$ ;
4:     SE ( $p_j \in \text{escola } e$ ) ENTÃO {
5:        $\text{custo} = \text{distancia}(p_i, p_j) + \text{distancia}(p_j, e)$ ;
6:     }
7:      $\text{matrizCustos} = \text{matrizCustos} + \text{distancia}$ ;
8:   }
9:   criarLRC(matrizCustos);
10:  RETORNA LRC;
11: }
```

Algoritmo 4. Matriz de distâncias de LRCj.

Para este trabalho foi considerado que o valor de $\alpha=0,1$. (Um exemplo é mostrado na tabela 2). A cada iteração do Algoritmo 5 é construído uma LRC , através da matriz de custos, da seguinte acordo com a formulação:

$$GRASP = \text{matrizCustos } p \in P \leq \min + \alpha * (\max - \min)$$

Equação 5. Fórmula para criar a Lista Restrita de Candidatos do GRASP

Onde são selecionadas as melhores potenciais paradas e agrupadas em um subconjunto de paradas viáveis, logo, uma entre as melhores paradas candidatas é sorteada aleatoriamente, tomada do subconjunto LRC. O Algoritmo 5, mostra a geração da LRC.

Algoritmo 5. GRASP: Procedimentos de Busca Adaptativos Gulosos Randomizados

```

1: FUNÇÃO criarLRC(matrizCustos) {
2:   ENQUANTO (matrizCustos  $\neq \emptyset$ ) FAÇA {
3:      $\text{verificaCustos} = \text{matrizCustos}[n]$ ;
4:      $\min = \text{menor}(\text{matrizCustos } p \in P)$ ;
5:      $\max = \text{maior}(\text{matrizCustos } p \in P)$ ;
6:      $GRASP = \text{matrizCustos } p \in P \leq \min + \alpha * (\max - \min)$ ;
7:     SE ( $GRASP \leq \text{verificaCustos}$ ) ENTÃO {
8:        $LRC' = GRASP$ ;
9:     }
10:     $LRC = LRC + LRC'$ ;
11:  }
12:  RETORNA LRC;
13: }
```

Algoritmo 5. Procedimentos de Busca Adaptativos Gulosos Randomizados

O algoritmo 4, que é a função $LRCj(\alpha, pi, escola(i), P)$, tem características semelhantes ao algoritmo 3, porém, são calculadas as distâncias da parada pi , até todas as paradas não visitadas que estão associadas com a $escola(i)$, que foi passada como parâmetro para a função. No entanto, as demais paradas não serão incluídas na matriz de custos. (Para mais detalhes veja no Algoritmo 4).

3.1.3. MELHORIA PARA A SOLUÇÃO INICIAL COM ALGORITMO 2-OPT

A heurística de melhoramento 2-Opt foi proposta por Croes (1958), a ideia desta heurística é bastante simples, dado um conjunto R' de uma rota R_n construída, elimine duas arestas da solução e insira novamente duas arestas de forma cruzada, ou seja, se as arestas removidas foram os pares que ligam as paradas (k_1, k_2) e (j_1, j_2) , as arestas inseridas ligam as paradas na forma (k_1, j_2) e (j_1, k_2) . Se esta nova configuração for melhor que a anterior, ou seja, se a distância diminuir, mantenha a nova rota. Caso contrário, escolha novamente duas arestas para análise.

Algoritmo 6. Algoritmo 2-opt

```

1: FUNÇÃO descruzamentoDaRota( $R'$ ){
2:   PARA ( $i=1; i<quantidadeParadas-2; i++$ ) FAÇA {
3:     PARA ( $j=i+1; j<quantidadeParadas-1; j++$ ) FAÇA {
4:       SE( $d(vet[i-1],vet[i])+d(vet[j],vet[j+1])>d(vet[i-1],vet[j])+d(vet[i],vet[j+1])$ ){
5:          $aux = vet[i];$ 
6:          $vet[i]=vet[j];$ 
7:          $vet[j]=aux;$ 
8:       }
9:     }
10:  }
11:  RETORNA  $R'$ ;
12: }
```

Algoritmo 6. Algoritmo 2-opt

3.2. APRESENTAÇÃO DO GRASP APLICADO AO PROE

A figura 1, ilustra um exemplo do GRASP aplicado ao PROE, na aplicação prática foram consideradas as seguintes entradas de dados: uma Garagem, cinco Paradas e uma Escola. Também são dadas as todas distâncias entre, garagem-paradas, garagem-escolas e paradas-escola. Para iniciar a construção de uma rota, o ônibus escolar sai da garagem e deve visitar todas as paradas, coletar os alunos e deixá-los na sua escola.

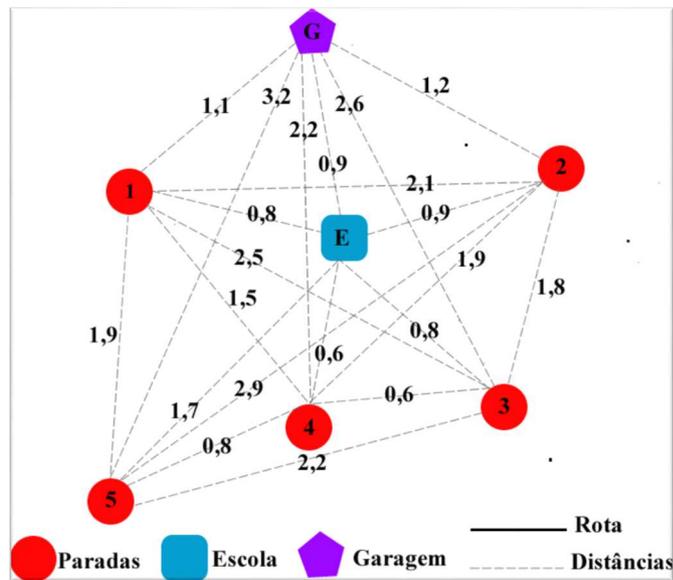


Figura 1. Representação do Problema a ser resolvido

Para a aplicação prática do problema, foi definido que o parâmetro adaptativo é $\alpha=0,1$ *¹. Quando a iteração 1 é aplicada na fórmula disponível na tabela 2, o resultado retornado é GRASP = 2,2, no entanto, todas as paradas com distâncias menor ou igual ao valor 2,2 serão adicionadas ao conjunto LRC, que são {P1 e P2}. Logo, uma parada é sorteada de forma aleatória. Sendo sorteado a parada P2. Veja na figura 1 que P1 e P2 são as paradas mais próximas da Garagem, que é ponto de partida do ônibus. Este método iterativo será executado sequencialmente até que a rota seja construída.

Matriz de Custos = $distancia(C_i, C_j) + distancia(C_j, C_k)$						
	P1	P2	P3	P4	P5	Sorteado
Iteração 1: $G \rightarrow P_n \rightarrow E$	1,9	2,1	3,4	2,8	4,9	P2
Iteração 2: $P2 \rightarrow P_n \rightarrow E$	2,9	-	2,6	2,5	4,6	P3
Iteração 3: $P3 \rightarrow P_n \rightarrow E$	3,3	2,7	-	1,2	3,9	P4
Iteração 4: $P4 \rightarrow P_n \rightarrow E$	2,1	2,8	1,4	-	2,5	P1
Iteração 5: $P1 \rightarrow P_n \rightarrow E$	-	3,0	3,3	2,1	3,6	P5

Legenda Paradas indisponíveis Paradas Selecionadas Paradas não Visitadas

Tabela 1. Critérios utilizados para a seleção das paradas com GRASP.

A Tabela 2 mostra passo a passo o processo iterativo do GRASP e como é construído a solução do problema.

*¹ O valor de α pode variar de 0 até 1. Caso seja definido que o $\alpha=0$, a LRC será totalmente gulosa, pois o resultado dos cálculos da fórmula será sempre o menor valor, desta forma a LRC sempre retorna uma única parada. Caso seja definido $\alpha=1$, o resultado dos cálculos da fórmula sempre será o maior valor, desta forma todas as paradas serão incluídas na LRC, logo, é sorteado uma parada de ônibus aleatória, sem nenhum critério de inserção.

$GRASP = p(P) / matrizCustos \ p \in P \leq \min + \alpha * (\max - \min);$	Paradas da LRC	Sorteado
Iteração 1: $1,9 + 0,1*(4,9-1,9) = 2,2$	{ P1 e P2 } $\leq 2,2$	P2
Iteração 2: $2,5 + 0,1*(4,6-2,5) = 2,71$	{ P3 e P4 } $\leq 2,71$	P3
Iteração 3: $1,2 + 0,1*(3,9-1,2) = 1,47$	{ P4 } $\leq 1,47$	P4
Iteração 4: $2,1 + 0,1*(2,5-2,1) = 2,14$	{ P1 } $\leq 2,14$	P1
Iteração 5: $3,6 + 0,1*(3,6-3,6) = 3,6$	{ P5 } $\leq 3,6$	P5
Solução Inicial	R = {G, P2, P3, P4, P1, P5, E, G}	
Após melhoria com o algoritmo 2-opt	R = {G, P2, P3, P4, P5, P1, E, G}	

Tabela 2. Cálculos utilizados para a seleção das paradas com GRASP

Na iteração 1 é verificado a distância da Garagem até cada uma das paradas, somando a distâncias da garagem até a parada com a distância da parada até a escola, desta forma é construído a matriz de custos apresentada na tabela, logo, é extraído da mesma o valor máximo e o valor mínimo. Nota-se que o valor máximo é 4,9 e o valor mínimo é 1,9, aplicando na formula da tabela 2, é retornado o resultado 2,2. Logo são verificados quais são as paradas que tem a distância menor ou igual a 2,2, e armazenados na LRC. Veja na tabela 1 que as paradas P1 e P2 são as paradas mais próximas do ponto de partida, esse processo é definido como potenciais paradas para começar a rota, logo uma delas será sorteada de forma aleatória. A parada P2 foi sorteada iniciando a construção da rota, R = {G, P2, ...}.

Na iteração 2 é realizado o mesmo procedimento da iteração 1, porém, o local de saída é P2, logo o valor do GRASP = 2,71, as paradas P3 e P4 são inseridos na LRC, então P3 é sorteado. A solução R= {G, P2, P3, ...}

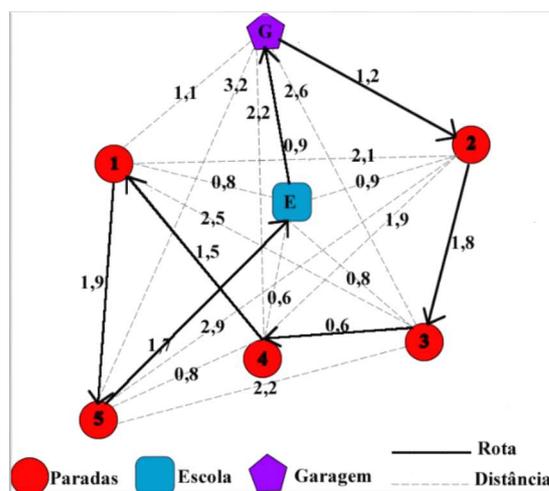


Figura 2. Rota Gerada com o GRASP

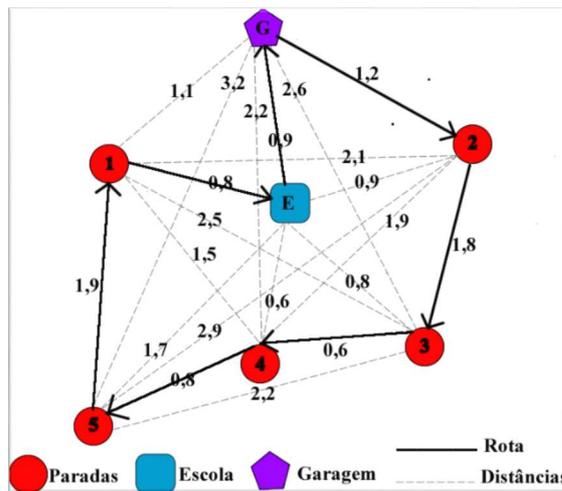


Figura 3. Rota melhorada com Algoritmo 2-opt

Na iteração 3, o valor do GRASP = 1,47, é inserido P4 no LRC, P4 é sorteada, Na Iteração 4 o valor do GRASP = 2,15 e P1 é inserido no LRC, logo, P1 é sorteado. Por fim na iteração 5 o valor do GRASP=3,6, note que na última iteração o valor do GRASP será igual ao custo da última parada que deve ser visitada.

Após o procedimento iterativo foi gerada a seguinte solução para a rota $R = \{G, P2, P3, P4, P1, P5, E, G\}$. Como pode ser visto na figura 2, a rota construída não está otimizada, somando as distancias tem-se o custo total de 9,6.

Quando é aplicado o algoritmo heurístico 2-opt é verificado se há uma melhoria na solução e então o algoritmo retorna a rota otimizada. Veja na figura 3, que o custo total da rota diminui para 8,0. A rota otimizada é representa por $R = \{G, P2, P3, P4, P5, P1, E, G\}$. O algoritmo 2-Opt pode ser visto no Algoritmo 6.

4. ANÁLISE E DISCUSSÃO DOS RESULTADOS

Os resultados do trabalho serão apresentados da seguinte forma:

- Comparação dos resultados computacionais, afim de validar a eficiência do algoritmo proposto, comparando-o com as instâncias de referências de Park et al., (2012) e o algoritmo de Braca et al., (1997) .
- Simulação de um ambiente do mundo real, comparados com a implementação do algoritmo LBH de Braca et al., (1997).
- O sistema de gerenciamento de apoio a decisão para o PROE.

Para obter os resultados computacionais com a execução do algoritmo proposto foi utilizado um computador com a seguinte configuração: CPU² Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz, 8Gb de memória RAM 1333MHz DDR3 e sistema operacional Microsoft Windows Professional versão 8.1. Já o código fonte da heurística GRASP proposta, foi implementada na linguagem de programação PHP versão 5.5.12.

Para realizar uma avaliação justa do tempo computacional necessário para gerar a solução, foram analisados os desempenhos das CPUs utilizadas, de acordo com as melhores soluções. Para isto foi utilizado o site CPU Benchmark que é especializado em avaliação de CPUs, na ocasião foram comparados a CPU da proposta de Park et, al. 2012, que utilizou uma CPU Intel Pentium 4 3.00GHz, com a CPU utilizada para a obtenção resultados deste trabalho, que é um Intel Core i5-3230M 2.60GHz. O resultado da avaliação é mostrado na figura 4 e está disponível em [https://www.cpubenchmark.net/compare.php?cmp\[\]=1849&cmp\[\]=1074](https://www.cpubenchmark.net/compare.php?cmp[]=1849&cmp[]=1074).

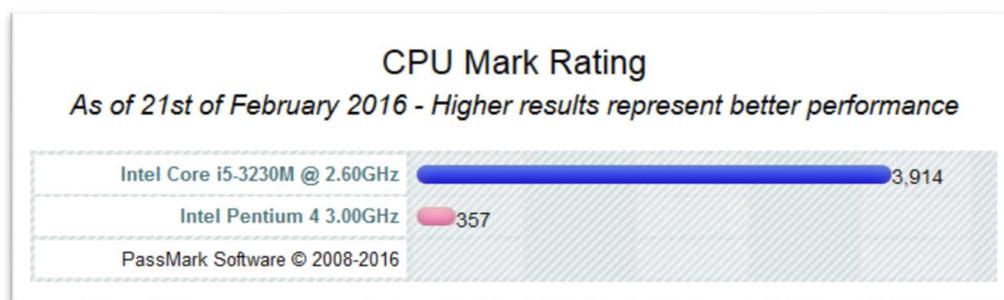


Figura 4. Avaliação de performance das CPUs utilizadas.

² CPU do termo em inglês (Central Processing Unit) que significa em português Unidade Central de Processamento.

De acordo com a classificação do site supracitado a CPU Intel P4 3.00GHz tem classificação número 1917 e o Intel Core i5-3230M 2.60GHz tem a classificação número 557. Essa classificação foi elaborado com a base na avaliação entre todos os processadores existentes no mercado, onde foi atribuído uma classificação para cada um deles.

Note que, a performance da avaliação do CPU utilizado para os resultados deste trabalho é aproximadamente 11 vezes melhor que o CPU utilizado por Park et, al. 2012.

Por outro lado o código fonte desta proposta, foi escrita com a linguagem de programação PHP (Hiptertext Preprocessor), que é interpretada e executado (rápido), já a linguagem de programação C/C++ é compilada e executada (muito rápido). Diante desta comparação, pode ser justificado uma comparação de equidade entre os trabalhos. Já que cada proposta tem uma certa vantagem em relação a outra.

4.1. RESULTADOS COMPUTACIONAIS COM INSTÂNCIAS DE REFERÊNCIA

Para comprovar a eficácia da metaheurística proposta, buscou-se os trabalhos de referência para o PROE que fornecem instâncias de referência de grande porte, para avaliar o comportamento do algoritmo GRASP com situações de problemas mais complexos. Também foi comparado o tempo computacional necessário para gerar a solução para o problema.

Para realizar os testes computacionais foram utilizadas as instâncias de referências de Park et al., (2012), citadas em vários trabalhos relacionados ao PROE, que estão disponíveis no link http://logistics.postech.ac.kr/Mixed_SBRP_Benchmark.html.

Os dados são resultados do estudo de Park et al., (2012), onde foram apresentados melhorias para o algoritmo de Carga Mista (Mix Load), em relação ao trabalho de Braca et, al (1997), que por vez, utilizou o algoritmo LBH de Bramel e Samchi-Levi (1992). Na ocasião o os resultados apresentados pelo algoritmo proposto por Park et al., (2012), conseguiram gerar soluções melhores para o PROE, que até o momento são as melhores da literatura.

A tabela 4, apresenta os dados referentes a 16 instâncias de referências para o PROE, onde foram testados 8 bases de dados com situações diferentes, com o número de paradas de ônibus entre 250 a 2000 e com o número de escolas entre 5 a 50. Foram testados as bases de dados com dois valores para o tempo máximo de percurso para o ônibus percorrer uma rota. Os valores são, 2700 e 5400 segundos, respectivamente 45 e 90 (minutos). No entanto, a rota é viável apenas se não violar a restrição de tempo máximo do percurso. Quanto menor, for a restrição de tempo de percurso da rota, maior será a quantidade de ônibus necessário para atender a demanda. Isto pode ser verificado nos valores do limite inferior, que representa a

quantidade mínima de ônibus necessários para atender todos os estudantes de cada base de dados. Os detalhes sobre cada uma das instâncias de teste proposta por Park et al., (2012) estão apresentadas na tabela 3.

Instâncias de referência da literatura de Park et, al. (2012)					
Instância	Tempo percurso	Quantidade			Limite Inferior
		Escolas	Paradas	Estudantes	
RSRB01	2700	6	250	3409	22
RSRB02	2700	12	250	3671	21
RSRB03	2700	12	500	6844	45
RSRB04	2700	25	500	6867	31
RSRB05	2700	25	1000	13,765	58
RSRB06	2700	50	1000	12,201	38
RSRB07	2700	50	2000	26,934	83
RSRB08	2700	100	2000	32,048	53
RSRB01	5400	6	250	3409	22
RSRB02	5400	12	250	3671	21
RSRB03	5400	12	500	6844	45
RSRB04	5400	25	500	6867	31
RSRB05	5400	25	1000	13,765	58
RSRB06	5400	50	1000	12,201	38
RSRB07	5400	50	2000	26,934	83
RSRB08	5400	100	2000	32,048	53
MÉDIA	4050,0	35,0	937,5	13217,4	43,9

Tabela 3. Entrada de dados para instâncias de referências de Park et, al (2012).

Na tabela 4 são apresentados os resultados computacionais da proposta apresentada, com a execução do algoritmo GRASP. Para isso, foram comparados dois requisitos:

- Quantidade de ônibus necessários para atender toda a demanda de transporte escolar.
- Tempo de processamento computacional necessário para gerar a solução com todas as rotas.

Com a otimização/minimização da quantidade de ônibus necessários para atender o transporte escolar, pode-se reduzir o custo total referente a prestação do serviço, consequentemente serão economizados gastos com manutenção de ônibus, economia de combustível, redução de gastos com pneus, salário do motorista e principalmente o gastos com aquisições ou alugueis de novos ônibus.

Na figura 5, são demonstrados os gráficos de comparações da quantidade de ônibus utilizados gerados por cada um dos quatro algoritmos para a solução.

Comparação dos resultados da metaheurística GRASP em relação aos principais trabalhos da literatura do PROE									
Instância	VEÍCULOS REQUERIDOS				TEMPO COMPUTACIONAL				
	Braca LBH	Park Mixed Load	Park Single Load	GRASP	Braca LBH	Park Mixed Load	Park Single Load	GRASP	
2700 (tempo em segundos)	RSRB01	36	32	35	30	26,6	6,0	2,5	0,44
	RSRB02	34	29	32	22	32,6	4,5	2,6	0,37
	RSRB03	77	61	66	48	123,8	17,7	4,6	1,06
	RSRB04	78	65	68	45	121,1	11,8	5,4	0,84
	RSRB05	118	106	124	90	574,7	30,4	9,4	2,45
	RSRB06	106	93	103	73	695,5	21,4	9,6	1,89
	RSRB07	201	171	190	167	3296,1	95,2	18,1	6,90
	RSRB08	212	168	176	165	3455,6	49,8	24,3	12,51
5400 (tempo em segundos)	RSRB01	31	28	31	24	46,2	5,0	2,3	0,96
	RSRB02	29	25	30	15	43,3	4,5	2,6	0,93
	RSRB03	61	51	61	42	166,8	15,4	4,6	2,17
	RSRB04	60	52	57	33	183,7	11,9	5,0	1,52
	RSRB05	86	86	106	73	866,9	30,6	8,3	4,87
	RSRB06	81	75	82	50	760,8	17,3	8,5	3,53
	RSRB07	163	147	158	119	4422,7	79,8	15,9	13,04
	RSRB08	186	149	158	105	5728,4	45,4	21,3	12,31
MÉDIA	97,4	83,6	92,3	67,5	1284,0	27,9	9,1	6,3	

Tabela 4. Comparação dos resultados computacionais.

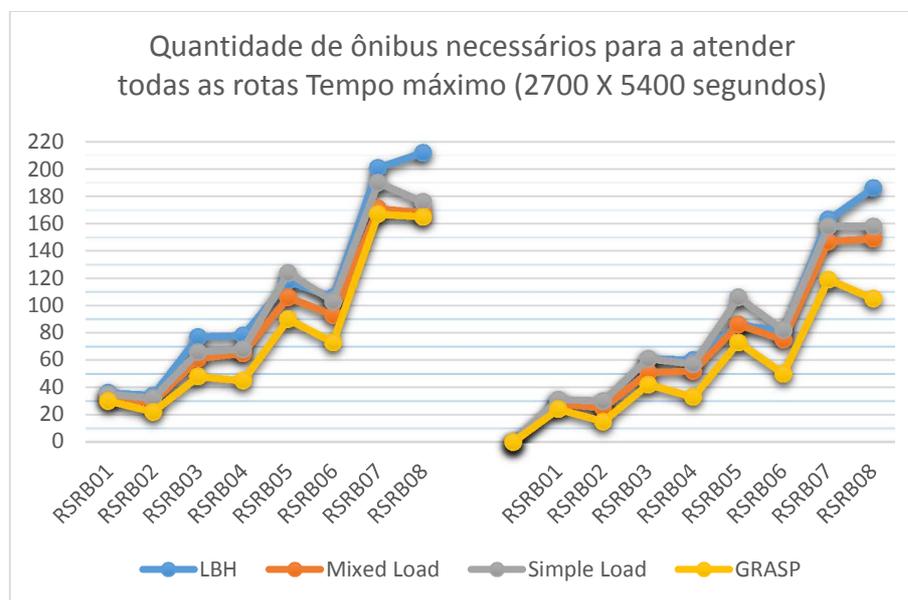


Figura 5. Gráfico Comparativo dos resultados computacionais.

Na tabela 4 são apresentados os resultados computacionais da proposta deste trabalho, destacados e representados pela coluna com o título GRASP. Nesta coluna é demonstrado que o algoritmo GRASP, obteve maior eficiência que os resultados de referência para o PROE em

todas as instâncias. Destacando a redução na quantidade de ônibus (média de 67,5 contra 83,6 do melhor competidor) em todas as instâncias e também no tempo de processamento computacional (média de 6,3 contra 8,1 do melhor competidor). Com os resultados, pode-se comprovar a eficácia da proposta apresentada em relação aos outros trabalhos de referências.

Na tabela 5, são mostrados os valores em percentagem da melhoria na solução com a metaheurística GRASP, quanto a minimização da quantidade de ônibus e a percentagem de tempo em relação ao tempo utilizado em relação ao trabalho de referência. Também são mostrados os percentuais comparativos para cada instância.

Para avaliar os resultados, foi calculado a média geral de todas as instâncias, o GRASP foi capaz de reduzir o número de ônibus utilizados em 30,7% em relação ao algoritmo LBH de Braca et, al (1997), 19,3% em relação ao algoritmo com carga mista (Mixed load) e 26,9% em relação ao algoritmo de carga simples (Single Load) de Park et, al. 2012.

Comparação dos resultados com a metaheurística GRASP em relação aos principais trabalhos da literatura do PROE (em %)							
	Instância	Redução (em %) Veículos necessários			Tempo computacional necessário para o GRASP em relação as soluções de referência (em %)		
		Braca LBH	Park Mixed Load	Park Single Load	Braca LBH	Park Mixed Load	Park Single Load
2700 (tempo em segundos)	RSRB01	16,7%	6,3%	14,3%	1,7%	7,3%	17,6%
	RSRB02	35,3%	24,1%	31,3%	1,1%	8,2%	14,2%
	RSRB03	37,7%	21,3%	27,3%	0,9%	6,0%	23,0%
	RSRB04	42,3%	30,8%	33,8%	0,7%	7,1%	15,6%
	RSRB05	23,7%	15,1%	27,4%	0,4%	8,1%	26,1%
	RSRB06	31,1%	21,5%	29,1%	0,3%	8,8%	19,7%
	RSRB07	16,9%	2,3%	12,1%	0,2%	7,2%	38,1%
	RSRB08	22,2%	1,8%	6,3%	0,4%	25,1%	51,5%
5400 (tempo em segundos)	RSRB01	22,6%	14,3%	22,6%	2,1%	19,2%	41,7%
	RSRB02	48,3%	40,0%	50,0%	2,1%	20,7%	35,8%
	RSRB03	31,1%	17,6%	31,1%	1,3%	14,1%	47,2%
	RSRB04	45,0%	36,5%	42,1%	0,8%	12,8%	30,4%
	RSRB05	15,1%	15,1%	31,1%	0,6%	15,9%	58,7%
	RSRB06	38,3%	33,3%	39,0%	0,5%	20,4%	41,5%
	RSRB07	27,0%	19,0%	24,7%	0,3%	16,3%	82,0%
	RSRB08	43,5%	29,5%	33,5%	0,2%	27,1%	57,8%
	MÉDIA	30,7%	19,3%	26,9%	0,5%	22,6%	69,2%

Tabela 5. Comparação dos resultados computacionais em percentual (%).

O tempo computacional do GRASP foi menor que a média geral de todas as instâncias, em relação ao tempo necessário para gerar a solução, sendo possível observar que, o tempo médio de processamento foi de apenas 0,5 % do tempo médio de processamento do algoritmo

LBH de Braca et, al (1997), 19,3% do tempo médio de processamento do algoritmo com carga mista (Mixed load) e 26,9% do tempo médio de processamento algoritmo de carga simples (Single Load) de Park et, al. 2012. Ressalta-se que, a metaheurística GRASP proposta nesse trabalho superou os resultados da proposta de Park et al., (2012) que apresentam os melhores resultados da literatura até este momento, tanto no tocante a qualidade da solução (minimização do número de ônibus) quanto ao tempo computacional.

4.2. SIMULAÇÃO PARA UM PROBLEMA DO MUNDO REAL

Para realizar os testes computacionais do algoritmo proposto com dados reais de uma cidade, foi criada uma instância com 1.169 estudantes, 87 paradas de ônibus, 8 escolas e 1 garagem.

Execuções	LBH			GRASP		
	Tempo Comput.	Custo total (Km)	Qtd. Ônibus	Tempo Comput.	Custo total (Km)	Qtd. Ônibus
1	0,102018	125,85	7	0,020321	81,33	5
2	0,138338	111,09	7	0,022123	75,25	5
3	0,049529	104,70	7	0,021825	73,34	4
4	0,023481	107,90	7	0,024696	96,34	6
5	0,036810	130,12	8	0,020666	83,5	6
6	0,041592	105,97	7	0,030243	85,56	6
7	0,041592	105,97	7	0,024932	83,84	5
8	0,022194	119,28	8	0,023506	86,59	5
9	0,064605	100,10	7	0,044162	80,42	5
10	0,067856	109,53	7	0,055045	84,36	6
11	0,024572	103,69	7	0,024351	74,83	5
12	0,031936	110,99	7	0,030865	78,37	6
13	0,027873	105,55	7	0,031942	83,29	5
14	0,023549	109,09	7	0,022652	71,38	5
15	0,057546	102,77	7	0,028323	83,01	5
16	0,061676	118,55	7	0,027575	88,07	6
17	0,038436	124,82	8	0,022762	77,33	6
18	0,025858	110,24	7	0,057546	82,03	6
19	0,024077	116,32	8	0,040380	88,70	6
20	0,023080	111,56	7	0,027088	81,52	6
Média	0,046330	111,70	7,2	0,030050	81,95	5,45

Tabela 6. Resultados computacionais obtidos com a execução dos algoritmos LBH e GRASP

Para cada uma das paradas de ônibus, escolas e garagens cadastradas foram inseridas as suas respectivas coordenadas geográficas extraídas do sistema Google Maps v3, com

localizações reais da cidade de Araguatins, Tocantins. A figura 6 da seção 4.3.2 mostra o mapa com as localizações marcadas.

Para gerar a solução com as rotas de ônibus foram realizadas simulações com a execução do algoritmo proposto, que foi apresentado na figura 3 e também foi implementado e executado o algoritmo LBH, apresentado no Algoritmo 1, para fins comparativos.

Na tabela 6, são apresentadas as médias dos resultados computacionais obtidos com a execução de cada um dos algoritmos. Para isto, foram realizadas repetições de 20 execuções para o algoritmo GRASP, logo, o resultado é carregado em uma variável responsável por armazenar o conjunto com todas as soluções geradas, por cada execução do algoritmo, assim, são obtidos 20 soluções. O mesmo procedimento foi feito com o algoritmo LBH. Ressalta-se que os dois algoritmos fazem leitura da mesma entrada de dados. A média foi calculada da seguinte maneira: somatória da coluna é dividida pelo número de execuções. As colunas são representadas por tempo computacional (tempo necessário para gerar uma solução), custo total (é a somatória das distâncias de percurso de todas as rotas em quilômetros - Km) e a quantidade de ônibus necessários para atender todas as paradas.

Os resultados mostram que com uso do algoritmo GRASP, obteve-se uma redução significativa de 36,30% na distância percorrida em Km, uma economia de 32,11% na quantidade de ônibus necessária e redução de 54,17% no tempo computacional.

4.3. O SISTEMA DE APOIO A DECISÃO PARA O PROE

4.3.1. ENTRADAS DE DADOS

As entradas dos dados são fundamentais para os sistemas de informações, na proposta apresentada, as entradas de dados são fundamentais para gerar a solução, os parâmetros de entrada são gravados no banco de dados e as informações são filtradas para auxiliar o administrador do sistema no apoio a tomada de decisão.

Para garantir uma maior precisão na solução geração de rotas pelo sistema, alguns cuidados foram tomados, especialmente em relação ao endereço do usuário, que no sistema são associados com as coordenadas geográficas da residência, com o mesmo sendo feito para o cadastro de escolas e garagem, visto que essas informações são primordiais para a execução do algoritmo.

Para todos os cadastros que necessitam de localização geográfica, o sistema faz uma validação do endereço antes de abrir o formulário de cadastro. Isso acontece nos cadastros de usuários, paradas, escolas e garagem. Ao lado do formulário de cadastro é mostrado um

marcador, indicando o local da residência do usuário, com todas as informações do endereço. Este recurso foi desenvolvido utilizando as funções do Google Maps v3, AutoComplete e Marker, sendo que a primeira tem a finalidade de capturar o endereço e as coordenadas geográficas, e a segunda faz a marcação da localização do mapa, que é mostrado na figura 6, em formulário de cadastro. Desta forma, o usuário pode verificar se a sua localização está correta.

Cadastro de Usuários

Nome	<input type="text"/>	Senha	<input type="password"/>
CPF	<input type="text"/>	Contato	<input type="text"/>
Responsável	<input type="text"/>	Coord. Lat.	<input type="text" value="-5.6526423"/>
Escola	<input type="text" value="100004"/>	Coord. Long.	<input type="text" value="-48.1106925"/>
Parada	<input type="text" value="4"/>	Número	<input type="text" value="1015"/>
Endereço	<input type="text" value="Rua Antônio Fernandes de Oliveira"/>	CEP	<input type="text" value="77950-000"/>
Cidade	<input type="text" value="Araguatins"/>	Estado	<input type="text" value="TO"/>
País	<input type="text" value="Brasil"/>		

Figura 6. Cadastros do sistema com Google Maps v3, recursos AutoComplete e Marker

4.3.2. MAPEAMENTO E VISUALIZAÇÕES DO AMBIENTE.



Figura 7. Função Marker do Google Maps v3, localizações de paradas, escolas e garagem

Com o intuito de mostrar a distribuição de paradas de ônibus, escolas e garagens, foi criado um mapa da cidade com a visualização da marcação de todas as coordenadas geográficas das paradas de ônibus, escolas e garagem. Para isto foi utilizado a função *Marker do Google Maps v3*. A legenda da figura 7 mostra os marcadores, vermelho, verde e azul respectivamente representam paradas, escolas e garagem.

4.3.3. VISUALIZADOR DE ROTAS

O visualizador de rotas foi desenvolvido com intuito de tornar público o acesso das informações sobre as rotas do ônibus escolar, onde, os usuários do transporte escolar, podem ver qual é o percurso realizado por cada ônibus e principalmente a rota que o estudante está associado.

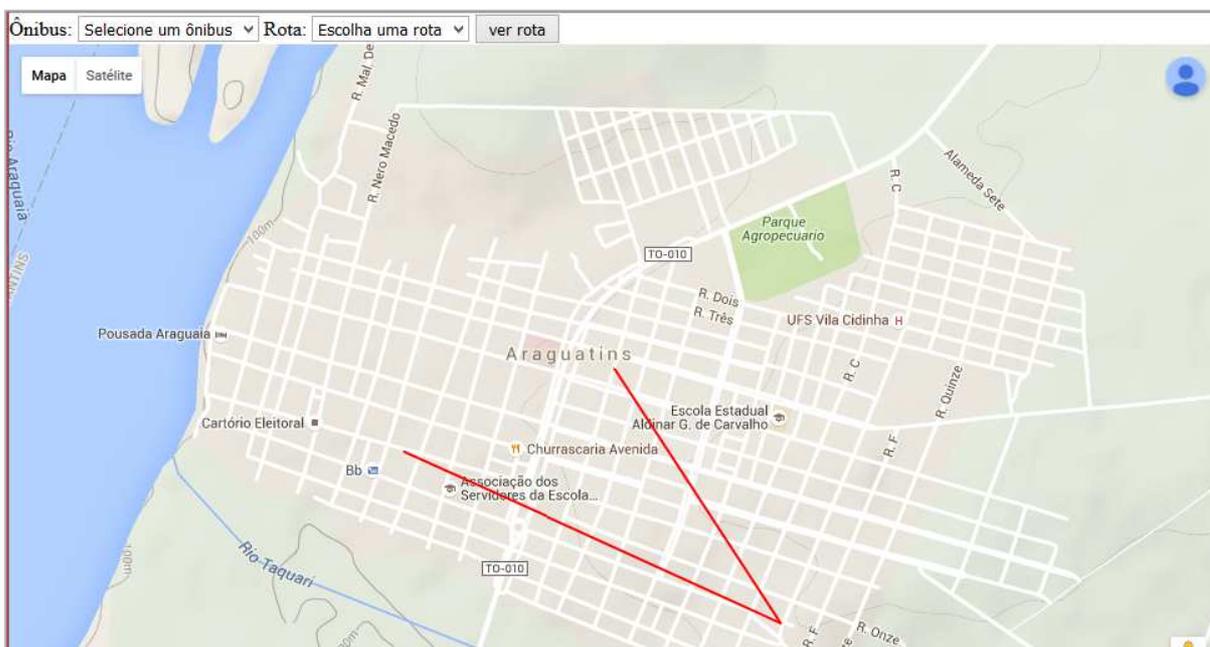


Figura 8. Função Polilines Google Maps v3, visualizador de rotas

A figura 8, mostra a visualização de uma rota. Para demonstração neste trabalho, foram utilizados os seguintes recursos do Google Maps v3: coordenadas geográficas e a função Polilines. Sendo que a primeira tem a finalidade de marcar os locais que devem ser visitados, e a segunda risca em linha reta o percurso do ônibus. Uma melhor visualização das rotas é apresentado no aplicativo mostrado na seção 4.3.4.

4.3.4. APLICATIVO DE VISUALIZAÇÃO DE ROTAS

Para melhorar o acesso às informações das rotas de ônibus escolar, foi desenvolvido um aplicativo para smartphones com Sistema Operacional Google Android.

O aplicativo faz a leitura em um arquivo de consulta no formato JavaScript Object Notation (JSON), que é gerado pelo sistema proposto e os dados são carregados em uma página web, por exemplo: www.exemplo.com/json.php. Desta forma os dados são carregados pelo aplicativo, e as rotas podem ser consultadas e carregadas no aplicativo, cada vez que for requerida. A figura 9, mostra a interface do aplicativo para visualizar as rotas.

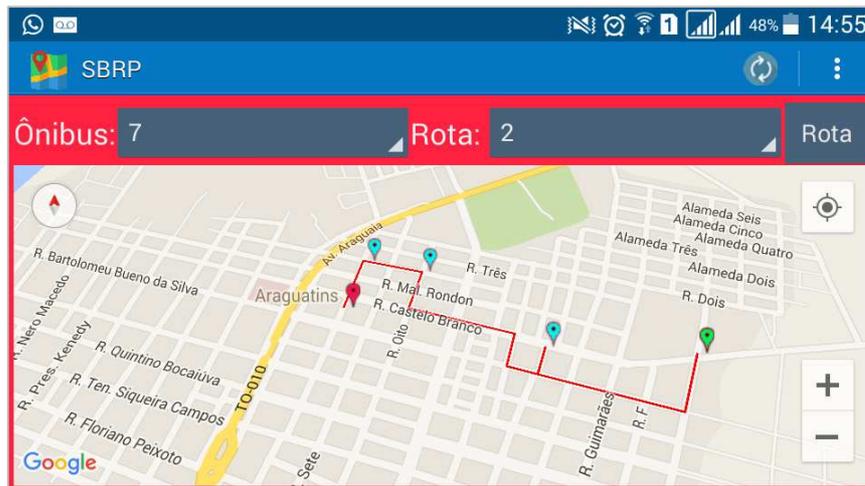


Figura 9. Aplicativo Android, para visualizar as rotas

5. CONCLUSÃO E TRABALHO FUTUROS

5.1. CONCLUSÃO

No Brasil, o transporte escolar é financiado e ofertado pelo Ministério da Educação em parceria com os Estados e Municípios. É regulamentado pela lei 9394/96, a Lei das Diretrizes Bases da Educação, com acréscimo na lei 10.709/2003. Um número considerável de estudantes das redes estaduais e municipais necessitam do transporte escolar para ter acesso a escola, principalmente os residentes na zona rural.

Este trabalho traz avanços significativos para o problema de roteamento do ônibus escolar. Através destes estudos foi desenvolvido um algoritmo com base na metaheurística GRASP, capaz de gerar rotas para o transporte escolar, com uma quantidade de dados razoável. Neste contexto, o sistema mostrou-se eficiente para um problema, que contenha até 2.000 paradas e 100 escolas. Se aplicado no Brasil, o sistema pode ser utilizado em várias cidades brasileiras.

Os testes computacionais mostram excelentes resultados para uma simulação, onde é gerado uma solução para uma instância com total de 32.048 estudantes, 2.000 paradas e 100 escolas. Para gerar a solução, o algoritmo demorou o tempo de 12,51 segundos para o executar e mostrar o resultado.

O algoritmo foi implementado em um sistema de apoio a decisão para o PROE. Com o sistema é possível gerenciar e manter informações de estudantes, paradas, ônibus, escolas. Além disso, o sistema dispõe de mapas da cidade com marcação de paradas, escolas e garagem. O sistema faz uso do Google Maps v3. Desta forma, o sistema proporciona uma visualização rápida para o administrador do sistema, sobre a distribuição de pontos de ônibus na cidade. Possui também um visualizador de mapas de rotas, geradas pelo próprio sistema. Informações que são úteis para o administrador do sistema, para o motorista que realizará a rota e para a comunidade que pode acompanhar os horários e o percurso das rotas de cada ônibus até chegar a escola.

Para melhorar o atendimento dos usuários do transporte escolar, também foi criado um aplicativo para smartphone com sistema operacional Google Android, com o objetivo visualizar as rotas realizadas pelo ônibus, bem como os locais de parada, tudo isto de forma rápida.

Para verificar as limitações da metaheurística GRASP, foram propostos dois testes para a aplicação, são eles: simulação de uma situação com uma cidade do mundo real e outra com instâncias de referências propostas por Park et al., (2012).

Os resultados computacionais comprovam que o GRASP é capaz de resolver o PROE. Em uma simulação do mundo real, foi calculando a média de 20 execuções realizadas pelos algoritmos GRASP e LBH, sendo comprovado que há uma economia significativa de 36,29% no custo total da distância percorrida, uma economia a 32,11% na quantidade de ônibus necessária e redução de 54,17% no tempo de execução.

Em comparação a estudos de referências da literatura, também foram obtidos excelentes resultados, comparando a média geral de todas as instâncias, o GRASP, foi capaz de reduzir o número de ônibus utilizados em, 30,7% em relação ao algoritmo LBH de Braca et, al (1997), 19,3% em relação ao algoritmo com carga mista (Mixed load) e 26,9% em relação ao algoritmo de carga simples (Single Load) de Park et, al. 2012 que é a referência da literatura no PROE até o momento.

Com os dados apresentados, espera-se que o trabalho possa contribuir com novos pesquisadores em continuar o estudo desse problema. Salientando que, quanto melhor for a solução encontrada, maior será a economia do recurso público e a qualidade do serviço prestado.

5.2. TRABALHOS FUTUROS

Para trabalhos futuros, pretende-se desenvolver um algoritmo de clusterização para ser implementado no sistema de apoio a decisão apresentado, com o intuito de determinar as potenciais paradas de ônibus, de acordo endereço dos estudantes utilizando coordenadas geográficas. Também serão estudados implementações de outros métodos heurísticos para o PROE, com objetivo de minimizar as limitações do sistema. Uma outra possibilidade interessante, é a implementação da metaheurística GRASP para aproveitar os recursos de processadores MULTI-CORE, de modo a obter melhor desempenho tanto no quesito de tempo de execução, como possivelmente da qualidade da solução encontrada.

REFERÊNCIAS BIBLIOGRÁFICAS

- ARIAS-ROJAS, J.; JIMÉNEZ, J.; MONTOYA-TORRES, J. Solving of school bus routing problem by ant colony optimization. **Revista EIA**, 2012.
- BEKTAS, T.; ELMASTAŞ, S. Solving school bus routing problems through integer programming. **Journal of the Operational Research Society**, v. 58, n. 12, p. 1599–1604, 2006.
- BÖGL, M.; DOERNER, K. F.; PARRAGH, S. N. The school bus routing and scheduling problem with transfers. **Networks**, v. 47, n. 1, p. 26–36, 2006.
- BOWERMAN, R.; HALL, B.; CALAMAI, P. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. **Transportation Research Part A: Policy and Practice**, 1995.
- BRACA, J.; BRAMEL, J.; POSNER, B.; SIMCHI-LEVI, D. A computerized approach to the New York City school bus routing problem. **IIE Transactions**, v. 28, 8, p. 693–702, 1997.
- BRONSHTEIN, E. M.; VAGAPOVA, D. M.; NAZMUTDINOVA, A. V. On constructing a family of student delivery routes in minimal time. **Automation and Remote Control**, v. 75, n. 7, p. 1195–1202, 2014.
- CARVALHO, W. L.; MOREIRA DA CRUZ, R. O.; CÂMARA, M. T.; GUILHERME DE ARAGÃO, J. J. Rural school transportation in emerging countries: The Brazilian case. **Research in Transportation Economics**, 2010.
- CASTILLO-LÓPEZ, I.; LÓPEZ-OSPINA, H. A. School location and capacity modification considering the existence of externalities in students school choice. **Computers & Industrial Engineering**, v. 80, p. 284–294, 2015.
- CHEN, X.; KONG, Y.; DANG, L.; HOU, Y.; YE, X. Exact and Metaheuristic Approaches for a Bi-Objective School Bus Scheduling Problem. **PloS one**, v. 10, n. 7, p. e0132600, 2015.
- CORBERÁN, A.; FERNÁNDEZ, E.; LAGUNA, M.; MARTÍ, R. . Heuristic solutions to the problem of routing school buses with multiple objectives. **JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY**, v. 53, 2000.
- CROES, G. A method for solving traveling-salesman problems. **Operations research**, 1958.
- DANTZIG, G.; RAMSER, J. The truck dispatching problem. **Management science**, 1959.
- DESROSIERS, J., FERLAND, J., ROUSSEAU, J.-M., LAPALME, G., AND CHAPLEAU, L. An Overview of School Busing System. , p. 235–243, 1981.
- DÍAZ-PARRA, O.; RUIZ-VANOYE, J. A.; BUENABAD-ARIAS, A.; COCÓN, F. Vertical transfer algorithm for the school bus routing problem. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. **IEEE** , 2012.
- ELLEGOOD, W. A.; CAMPBELL, J. F.; NORTH, J. Continuous approximation models for mixed load school bus routing. **Transportation Research Part B: Methodological**, v. 77, p. 182–198, 2015.
- EUCHI, J.; MRAIHI, R. The urban bus routing problem in the Tunisian case by the hybrid artificial ant colony algorithm. **Swarm and Evolutionary Computation**, 2012.
- FARAJ, M. F. ;SARUBBI, J. F. . M.; SILVA,C. M.; PORTO, M. F. Estudo de Caso: o Problema do Transporte Escolar Rural em Minas Gerais. **Simpósio Brasileiro de Pesquisa**

Operacional, p. 1332–1343, 2013.

FEO, T.; RESENDE, M. Greedy randomized adaptive search procedures. **Journal of global optimization**, 1995.

FISHER, M. A multiplier adjustment method for the generalized assignment problem. **Management Science**, 1986.

FARAJ, M.F.; SARUBBI, J.F.F. ; SILVA, C.M. ; PORTO, M.F. ; NUNES, N. T.R. A Real Geographical Application for the School Bus Routing Problem. **2014 IEEE 17TH International Conference on Intelligent Transportation Systems (Itsc)**, 2014.

FÜGENSCHUH, A. Solving a school bus scheduling problem with integer programming. **European Journal of Operational Research**, v. 193, n. 3, p. 867–884, 2009.

FÜGENSCHUH, A.; MARTIN, A.; STÖVEKEN, P. Integrated optimization of school starting times and public bus services. **Operations Research Proceedings**, 2005.

FULIN, S.; YUEGUANG, L. An Improved quantum-behaved particle swarm algorithm and its application in School Bus Problem. **Third International Conference on Digital Manufacturing & Automation An**, n. 1, p. 198–201, 2012.

HUO, L.; YAN, G.; FAN, B.; WANG, H.; GAO, W. School Bus Routing Problem Based on Ant Colony Optimization Algorithm. n. 1, p. 2–6, 2014.

KALOGIROU, K.; CHALKIA, E.; BEKIARIS, E.; DIEDERICHS, F. An Application for the Information of Children According their School Transportation. **Procedia - Social and Behavioral Sciences**, 2012.

KAMALI, B.; AND, S. J. M.; POHL, E. A. An Analysis of Special Needs Student Busing. **JOURNAL OF PUBLIC TRANSPORTATION**, v. 16, n. 1, 2013.

KIM, B.-I.; KIM, S.; PARK, J. A school bus scheduling problem. **European Journal of Operational Research**, 2012.

LI, L. Y. O.; FU, Z. The school bus routing problem: a case study. **Journal of the Operational Research Society**, v. 53, n. 5, p. 552–558, 2002.

MANDUJANO, P.; GIESEN, R.; FERRER, J.C. Model for Optimization of Locations of Schools and Student Transportation in Rural Areas. **Transportation Research Record: Journal of the Transportation Research Board**, v. 2283, p. 74–80, 2012.

MARTÍNEZ, L. M.; VIEGAS, J. M. Design and deployment of an innovative school bus service in Lisbon. **Procedia - Social and Behavioral Sciences**, 2011.

MONTOYA-TORRES, J. S. A.-R. J. F. J. J. R. Solving of School Bus Routing Problem by Ant Colony Optimization. **Revista EIA** , p. 193–208, 2012.

NEWTON, R. M.; THOMAS, W. H. Design of school bus routes by computer. **Socio-Economic Planning Sciences**, v. 3, n. 1, p. 75–85, 1969.

PACHECO, J.; CABALLERO, R.; LAGUNA, M.; MOLINA, J. Bi-Objective Bus Routing: An Application to School Buses in Rural Areas. **Transportation Science**, , n. September 2015.

PARK, J.; KIM, B. I. The school bus routing problem: A review. **European Journal of Operational Research**, 2010.

PARK, J.; TAE, H.; KIM, B. I. A post-improvement procedure for the mixed load school bus routing problem. **European Journal of Operational Research**, 2012.

PRASETYO, D.; MUHAMAD, J.; FAUZI, R. Supporting needy student in transportation: a

population based school bus routing in spatial environment. **International Conference on Social**, 2011.

RIERA-LEDESMA, J.; SALAZAR-GONZÁLEZ, J. J. Solving school bus routing using the multiple vehicle traveling purchaser problem: A branch-and-cut approach. **Computers and Operations Research**, 2012.

SCHITTEKAT, P.; KINABLE, J.; SÖRENSEN, K.; et al. A metaheuristic for the school bus routing problem with bus stop selection. **European Journal of Operational Research**, 2013.

SCHITTEKAT, P.; SEVAUX, M.; SIRENSEN, K. A mathematical formulation for a school bus routing problem. **IEEE transactions on systems**, 2006.

SGHAIER, S. BEN; GUEDRIA, N. BEN; MRAIHI, R. Solving School Bus Routing Problem with genetic algorithm. International Conference on Advanced Logistics and Transport, ICALT. p.7–12, 2013.

SONG, S.-M.; KIM, T. Customer-oriented school bus operations for childcare centers in Korea. **Computers & Industrial Engineering**, v. 66, n. 1, p. 116–124, 2013.

SOUZA, L. DE; SIQUEIRA, P. Heuristic methods applied to the optimization school bus transportation routes: a real case. **Trends in Applied Intelligent Systems**, 2010.

STEINER, M.; ZAMBONI, L. O problema de roteamento no transporte escolar. **Pesquisa Operacional**, 2000.

TENG, J.; YANG, X. Study on the Optimization of Bus Coordination Holding Control for Transit Hub. **Systems Engineering - Theory & Practice**, 2008.

ANEXOS

APÊNDICE A – ARTIGO PUBLICADO

Implementation the metaheuristic GRASP applied to the School Bus Routing Problem

Vilson Soares de Siqueira^{1,2}, Fernando Jorge Ebrahim Lima e Silva^{1,2}, Elvis Nascimento da Silva^{1,2}, Raimunda Vieira Santos da Silva¹ e Marcelo Lisboa Rocha²

¹Systems Development Department, Federal Institute of Tocantins, Brazil

²Computer Science Department, Federal University of Tocantins, Palmas, Brazil

Abstract. The school bus problem routing (SBRP) is an important practical problem studied in combinatorial optimization of operational research. It is formulated through a set of stops, buses, schools and garage, where from these sets, we seek to create optimized routes to reduce the operating cost of the service. This paper presents a solution to the SBRP, using the GRASP applied to a real problem. This meta-heuristic is divided into two stages: the construction of a viable solution and followed by a Local Search procedure. These two phases are repeated in each iteration. In the construction phase, a greedy and random function is used to construct an initial solution. It is also presented compared to the widely used heuristic and good results in the literature.

Keywords: Combinatorial Optimization, school bus, GRASP, SBRP, Metaheuristic.

1. Introduction

The school bus routing problem (SBRP) is a problem of combinatorial optimization, commonly referred to in operational research. Through a literature review, which found the number of publications available in leading journals, note that the problem is little explored in the literature, when compared to the number of jobs that treats the classic problem routing vehicles (VRP).

In the literature, it was found SBRP case studies in Brazil.[1] It presents a proposed heuristic method to solve the problem and a case study in the state of Parana, Brazil and presented the resolution for rural and urban areas. [2] It presents a case study of SBRP to rural areas in the state of Minas Gerais, Brazil. [3] It presents a solution for urban transportation to the city of Curitiba, Parana, Brazil. [4] It presents a methodology location of schools and optimizing the school transport in rural areas. This methodology is based on two models of mixed integer programming. The first deals with the school's location and sizing issues, and the second deals with the school bus routing. The case study was conducted in the city of Grajau Baron state of Maranhao in Basil.

The objective of this work is to create a method to solve the SPREP, considering the service in rural and urban areas, considering time windows, as well as a fleet of heterogeneous vehicles.

2. Description the School Bus Routing Problem

The School Bus Routing Problem (SBRP) is a combinatorial optimization problem which was modeled by Newton and Thomas [5]. The SBRP is a variation of the Vehicle Routing Problem (VRP) [6], which is commonly treated in operations research.

The SBRP aims to optimize the school bus transport, where the student is pickup in a bus stop located in front of his residence or close to it, then the student is transported and delivered in your school at the end of the school day the student is transported again for same location where it was pickup.

In the literature can be found several problems, methods and constraints to solve the SBRP. [7] It points out that the start of school time takes place at the same time, therefore, it generates a large bus demand at the beginning and end of the school day.

[8] It shows in his work on the size of the SBRP objectives of the proposal and propose a multi-objective solution, to minimize the number of buses used for pickup and delivery of students and improve the quality of the service offered. For this, treated separately each objective, considering not only the optimal solution, but also the set of measured way.

[9] It presents a model for SBRP taking into consideration the attendance to students with specific needs or vulnerable because of health problems. [10] Presents an application for monitoring of school transport in Europe to transport vulnerable users (children, people with specific needs or poor health).

In the work [11] Emphasizes on investment of the Ministry of Education of China, which invested in the launch of a pilot program of school bus optimization, which were chosen in 2011, six cities to perform the tests in a real environment, so search for the best practices for school bus service.

[12] Developed a randomized algorithm Location Based Heuristics (LBH) and showed a linear regression method for time duration of the pickup of students and for the duration of delivery time from at school, it was also considered the bus waiting time. To solve the problem it was used the Mix Load method. An improvement to the algorithm is presented by [13]. The LBH algorithm was implemented in this work, to carry out computational tests and compare with proposed results of this paper.

The Metaheuristic GRASP

The Greedy Randomized Adaptive Search Procedure (GRASP) It was developed by [14]. It is an iterative procedure where each iteration consists of a construction phase an initial solution Greedy randomized adaptive way and then a heuristic is applied for improvement, typically a local search procedure. The best solution among all the iterations is stored [14]. The figure 1 shows the pseudocode GRASP.

The Constructive Algorithm

The proposal is to develop a system decision support for SBRP, for it will be used the metaheuristic GRASP. In the literature are found resolution methods and formulation so that the problem is treatise separately, such as the types of fleet (homogeneous or heterogeneous) and types of attendance (in areas rural or urban). The aim is to create an algorithm to solve the general form problem.

Considering that, each set corresponds to a table in one database and that has attributes, primary keys and foreign keys. The following steps are presented for the construction method: In step 1, must be selected a parameter (Urban or Rural), respectively represented by binary numbers (0 and 1). In step 2, according to the last parameter, the sets are filtered from the Database, and the sets are loaded into Memory RAM. In the step 3 is started the execution of the construction method.

Given the following set of data entries, where P is the set Stops, O is the set Bus, G is the set of Garage, E is the set Schools and R is the set of routes, initiated empty. In phase 1, the loop is started the constructive method, a conditional is then checked if the solution set $R' = \emptyset$ or the bus hours of attendance $>$ that start time of school is randomly selected an o the set O , is taken at random. If the bus hours of attendance $<$ school start time, the same bus will be used to serve a new route. The bus has a capacity constraint, where is not permitted to their limit is exceeded. When the bus hours of attendance constraint is satisfied, the buses used will be taken from the set O .

In the phase 2, is executed the function $RCLi(a, ge, P)$ it is responsible for calculating and creating an array of costs from an output location of the bus, which may be the garage G or the school E , until all the stops $p \in P$, (see in figure 2), that calls the function $createRCL(arrayCosts)$ which uses criteria of GRASP, to create a set of best stops $p \in P$, (see in figure 4). All details are shown in the session 2.3.

In the phase 3, $ci \leftarrow RCLi(a, ge, P)$, it is chosen at random a stop $pi \subset ci / ci \in P$, then, the solution R' get the stop pi and the $school(i)$ which is associated with pi , starting a route Rn , after being inserted into R' , the stop pi is removed the set P .

In the phase 4, While the bus capacity \leq the quantity pickup students, a new bus stop is visited. An iterative procedure is carried out by the function $cj = \mathbf{RCLj}(\alpha, school(i), pi, \mathbf{P})$, a manner similar to phase 3. Therefore, the following parameters are sent: the stop pi and the $school(i)$, after, the function $\mathbf{RCLj}(\alpha, school(i), pi, \mathbf{P})$ it is responsible for calculating and creating an array of costs between the stop pi until all the stops $p \in \mathbf{P}$ that has as its destination the $school(i)$, (see in figure 3) that calls the function $\mathbf{createLCR}(\text{arrayCosts})$ which uses criteria of GRASP, to create a set of best stops $p \in \mathbf{P}$, (see in figure 4). All details are shown in the session 2.3.

Algorithm 1. Constructive Method for the Metaheuristic GRASP

```

1:  $\mathbf{P} = \{1, 2, \dots, p\}$  // Set the stops not visited
2:  $\mathbf{O} = \{1, 2, \dots, o\}$  // Set the buses
3:  $\mathbf{G} = \{1, 2, \dots, g\}$  // Set the garage
4:  $\mathbf{E} = \{1, 2, \dots, e\}$  // Set the schools
5:  $\mathbf{R} = \{\emptyset\}$  // Set empty, for gets the routes
6: WHILE ( $\mathbf{P} \neq \emptyset$ ) DO{
7:    $b =$  a randomly bus selected the set  $\mathbf{O} \in \mathbf{G}$ ;
8:    $ci = \mathbf{RCLi}(\alpha, ge, \mathbf{P})$ ; // Restricted Candidate List (RCL), all stops
9:    $pi =$  a randomly bus selected the set  $\mathbf{LCRi}$  de  $ci \in \mathbf{P}$ 
10:   $\mathbf{R}' = pi, school(i)$ ; // school  $e$  associated with stop  $i$ 
11:   $\mathbf{P} = \mathbf{P}$ ; // removed stop  $pi$  the set  $\mathbf{P}$ ;
12:  WHILE ( $\text{capacityBus} \leq \text{quantityStudents}$ ) DO {
13:     $cj = \mathbf{RCLj}(\alpha, school(i), pi, \mathbf{P})$ ; // RLC, algorithm GRASP
14:     $pj =$  a randomly bus selected the set  $cj \in \mathbf{P}$ ;
15:    SE ( $\text{arrivalTime} \leq \text{timeStartingSchool}$ ){
16:       $\mathbf{R}' = j, school(i)$ ;
17:       $\mathbf{P} = \mathbf{P}$  // removed stop  $j$  the set  $\mathbf{P}$ ;
18:    }
19:    ELSE {
20:      Break executing and the bus goes to  $school(i)$ ;
21:    }
22:  }
23:   $\mathbf{uncrossroute}(\mathbf{R}')$ ; // algorithm 2-opt
24:   $\mathbf{R} = \mathbf{R} + \mathbf{R}'$ 
25: }
Output the Solution  $\mathbf{R}$ 

```

Figure 1. Constructive Method for the SBRP with GRASP

In the phase 5, $cj \leftarrow \mathbf{RLCi}(\alpha, school(i), g, \mathbf{P})$, where is selected at random a stop $pj \subset cj / cj \in \mathbf{P}$, after, the solution \mathbf{R}' get the stop pj e $school(i)$ which is associated with pj , therefore, a new stop pj is inserted in the route Rn . After being included in \mathbf{R}' , the stop pj is removed the stop \mathbf{P} , provided that it meets the restrictions of phase 6.

In the phase 6, if the bus arrival time at school \leq start time of the school day. At each iteration are added the following attributes in the solution \mathbf{R}' , the duration of the the route time of a stop i until j (time in seconds), the quantity pickup students, the duration of time to pickup all the students in the stop p , (time in seconds) and the landing time students at school. The iterative method continues to until the bus capacity constraint or restriction beginning the school day, are not violated. If the arrival time at school or the capacity of the bus, satisfy the constraints, the bus goes destined for school and left the students.

In the phase 7, is used the algorithm 2-opt, to improve the solution \mathbf{R}' . The details are shown in section 2.4.

In the phase 8, Shows the solution of all the built routes, $\mathbf{R} = \{R1, R2, \dots, Rn\}$

Restrict Candidate List

accordance with the initial parameters of the functions $\mathbf{RCLi}(\alpha, g, \mathbf{P})$ e $\mathbf{RCLj}(\alpha, school(i), g, \mathbf{P})$, an array of costs is constructed. The algorithm 2, which is the function $\mathbf{RCLi}(\alpha, ge, \mathbf{P})$, creates an array of

insertion cost for the construction of a route R' as of P . Where are calculated the cost of the garage path g or school e until each stop p and the stop p until the school e associated with stop p , so, $Cost = Ci + Cj + Ck$. (one example is showed in the table 2).

Algorithm 2. Restricted Candidate List Initial (RCLi) - Considers all the stops

```

1: FUNCTION RCLi ( $\alpha$ ,  $ge$ ,  $P$ ){
2:   WHILE ( $P \neq \emptyset$ ) DO {
3:      $pi = P[n]$ ;
4:      $e = school(i) \subset P[n] \in E$ ;
5:      $cost = distance(gi, pi) + distance(pi, e)$ ;
6:      $arrayCosts = arrayCosts + cost$ ;
7:   }
8:   createRCL( $arrayCosts$ );
9:   RETURN RCL;
10: }
```

Figure 2. Array the distance the RCLi.

Algorithm 3. Restricted Candidate List Initial (RCLj) – Considers only the stop the $school(i)$

```

1: FUNÇÃO RCLj ( $\alpha$ ,  $school(i)$ ,  $pi$ ,  $P$ ) {
2:   WHILE ( $P \neq \emptyset$ ) DO {
3:      $pj = P[n]$ ;
4:     IF ( $pj \in school e$ ) THEN {
5:        $cost = distance(pi, pj) + distance(pj, e)$ ;
6:     }
7:      $arrayCosts = arrayCosts + distance$ ;
8:   }
9:   createRCL( $arrayCosts$ );
10:  RETURN RCL;
11: }
```

Figure 3. Array the distance the RCLj.

The algorithm 3, is the function $LCRj$ (α , $school(i)$, pi , P), It has similar characteristics to algorithm 2, therefore, is insertet a the parameter $school(i)$, which is intended to filter only the stops p , having as destination the $school(i)$. The array cost is calculated from the stop pi , which was defined in (line 9) of the algorithm 1.

Algorithm 4. GRASP: Greedy Randomized Adaptative Search Procedure

```

1: FUNCTION createRLC( $arrayCosts$ ){
2:   WHILE ( $arrayCosts \neq \emptyset$ ) DO {
3:      $checkCosts = arrayCosts[n]$ ;
4:      $min = \min (arrayCosts p \in P)$ ;
5:      $max = \max (arrayCosts p \in P)$ ;
6:      $GRASP = arrayCosts p \in P \leq min + \alpha * (max - min)$ ;
7:     IF ( $GRASP \leq checkCosts$ ) THEN {
8:        $RCL = GRASP$ ;
9:     }
10:     $RCL = RCL + RCL$ ;
11:  }
12:  RETURN RCL;
13: }
```

Figure 4. Greedy Randomized Adaptative Search Procedure

After the creation of the array cost the function $createRCL(arraycosts)$ is called to identify the best potential stops p , to start construction the R' . In the construction phase generates a viable solution to the problem through the method partially greedy and partially random. Where the parameter α has the function

to determine the size of the Restricted Candidate List (RCL). The α is a value ranging from 0 until 1, where $\alpha=0$, determining that the greedy method is totally and $\alpha=1$, determines that the method will be totally random. This parameter is responsible for determining the size of the RCL, the nearest to 1, the value of α , the greater the number of stops, added together and the worse the solution to the problem.

For this study was considered the $\alpha=0,1$. (One example is showed in table 2). At each iteration of Algorithm 1 is built a RCL, which is formed from the array cost, where the best potential stops are selected, with the formula $GRASP = arrayCosts p \in P \leq min + \alpha*(max - min)$ showed in (line 6) the algorithm 4, being realized a grouping in a subset of potential viable stops, after, a stop is randomly selected, taken from the RCL subset, that has the best stops candidates. The figure 4, show the algorithm for RCL.

Improvement for initial Solution with Algorithm 2-opt

The improvement heuristic 2-Opt was proposed by Croes [15], the idea of this improvement heuristic is quite simple given a set R' that a route built in R_n , eliminate two edges of the solution and reinsert two edges cross, therefore, if the edges were removed peers that connect the stops $(k1, k2)$ and $(j1, j2)$, the inserted edges connect the stops in the following way $(k1, j2)$ and $(j1, k2)$. If this new configuration is better than the previous, so, if the distance reduce, keep the new route. Otherwise, choose again two edges for analysis.

3. GRASP application to SBRP

The figure 5, show one example, where, is considered one Garage, five stops an one school. It is also present the distances between each stop. To demonstrate the functioning of the GRASP applied to SBRP. To begin construction of a school bus route, is considered that the bus moves garage and should visit all the stops to collect the students and let them in school.

In iteration 1, (showed in table 1) calculates the distance between the garage until each stop, therefore, creating an array cost. Applying adaptive parameter $\alpha=0,1$, the following formula:

$$GRASP = p(P) / arrayCosts p \in P \leq min + \alpha*(max - min)$$

It obtained the following result to $GRASP = 2.2$, how can be seen in iteration the table 2. Then, All the stops with lower distance 2.2, will be inserted in the set $RCL = \{P1 \text{ and } P2\}$, After, one stops is randomly selected. After, is raffled P2. See that, P1 and P2 are the closest stops from Garage.

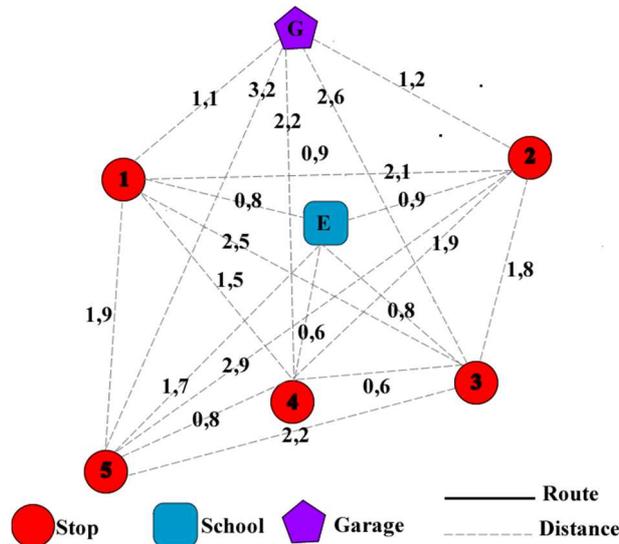


Figure 05. Problem representation to be solved

In iteration 2 (showed in figure 1) is calculated the distance between P2 until each stop unvisited. After, the resulted $GRASP = 2.71$, and the stops P3 and P4 They are inserted into the RCL and is raffled P3.

The procedure is carried until all the stops are visited.

Array the Costs = $(C_i, C_j) + (C_j, C_k)$						
	P1	P2	P3	P4	P5	Randomly
Iteration 1: $G \rightarrow P_n \rightarrow E$	1,9	2,1	3,4	2,8	4,9	P2
Iteration 2: $P2 \rightarrow P_n \rightarrow E$	2,9	-	2,6	2,5	4,6	P3
Iteration 3: $P3 \rightarrow P_n \rightarrow E$	3,3	2,7	-	1,2	3,9	P4
Iteration 4: $P4 \rightarrow P_n \rightarrow E$	2,1	2,8	1,4	-	2,5	P1
Iteration 5: $P1 \rightarrow P_n \rightarrow E$	-	3,0	3,3	2,1	3,6	P5

legend stops unavailable Stops selected Stops unvisited

Table 01 – Criteria used for selection of the stops with GRASP, considering $\alpha = 0.1$

$GRASP = p(P) / arrayCosts \ p \in P \leq \min + \alpha * (\max - \min);$	Stops the RCL	Randomly
Iteration 1: $1,9 + 0,1 * (4,9 - 1,9) = 2,2$	$\{ P1 \ e \ P2 \} \leq 2,2$	P2
Iteration 2: $2,5 + 0,1 * (4,6 - 2,5) = 2,71$	$\{ P3 \ e \ P4 \} \leq 2,71$	P3
Iteration 3: $1,2 + 0,1 * (3,9 - 1,2) = 1,47$	$\{ P4 \} \leq 1,47$	P4
Iteration 4: $2,1 + 0,1 * (2,5 - 2,1) = 2,14$	$\{ P1 \} \leq 2,14$	P1
Iteration 5: $3,6 + 0,1 * (3,6 - 3,6) = 3,6$	$\{ P5 \} \leq 3,6$	P5
Solutio Initial	$R = \{ G, P2, P3, P4, P1, P5, E, G \}$	
After improvement with 2-opt	$R = \{ G, P2, P3, P4, P5, P1, E, G \}$	

Table 02 – Results the selection of the stops with GRASP, considering $\alpha = 0.1$

After the iterative procedure, the following solution was generated for $R = \{ G, P2, P3, P4, P1, P5, E, G \}$. As can be seen in Figure 6, the route built is not optimized, for improving the route was used heuristic algorithm 2-opt, to leave the optimal route. The figure 7, shows the optimized solution that is represents by $R = \{ G, P2, P3, P4, P5, P1, E, G \}$.

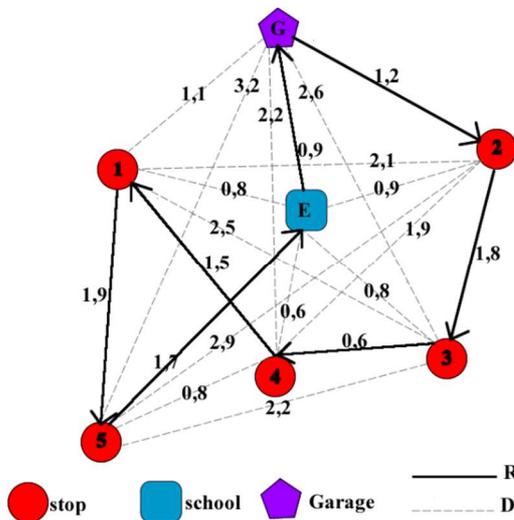


Figure 06. Generated route by GRASP

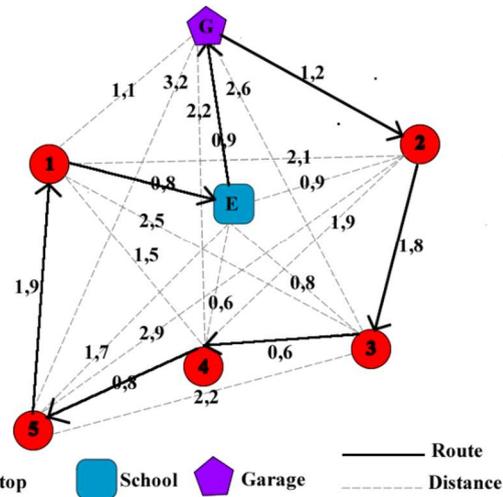


Figure 07. Route with improved algorithm 2-opt

To compare the computational results were implemented: the algorithm LBH the Braca [12], which is much cited in the literature for solving the problem and the metaheuristic GRASP [14].

For this problem, it was considered a total of 87 stops, 8 schools, and 1169 students. the results are showed in table 03.

To obtain the results shown in table 03, have been performed 20 iterations for algorithm LBH and GRASP, where have been performed the average bus used, computational time and the cost total, which is the sum of the distance of all routes in kilometers (KM). The computational results showed a savings of 36,29% in cost total in KM, an economy the 32,11% in quantity the bus necessary and 54,17% reduction in run time, with the purpose of this paper.

Algorithm LBH		Algorithm GRASP	
Necessary bus	7.2	Necessary bus	5.45
Computational time	0.0463309 (s)	Computational time	0.0300501 (s)
Total cost (KM)	111.70	Total cost (KM)	81.953

Table 03 – Results computational using 20 iterations

4. Conclusion

With the utilization of GRASP is obtained good results in comparison with the heuristic LBH. Have been performed 20 iterations for algorithm. The computational results showed GRASP of improvements over the LBH. An economy the 36,29% in cost total, an economy the 32,11% quantity the bus necessary and 54,17% reduction in run time.

Future Works

In future work, we intend to develop a decision support system for SBRP, based on the implementation of meta-heuristic algorithm GRASP.

5. References

- [1] L. de Souza and P. Siqueira, "Heuristic methods applied to the optimization school bus transportation routes: a real case," *Trends Appl. Intell. Syst.*, 2010.
- [2] M. F. Faraj, M. F. ;Sarubbi, J. F. . M.; Silva,C. M.; Porto, "Estudo de Caso: o Problema do Transporte Escolar Rural em Minas Gerais," *Simpósio Bras. Pesqui. Operacional*, pp. 1332–1343, 2013.
- [3] M. Steiner and L. Zamboni, "O problema de roteamento no transporte escolar," *Pesqui. Operacional*, 2000.
- [4] P. Mandujano, R. Giesen, and J.-C. Ferrer, "Model for Optimization of Locations of Schools and Student Transportation in Rural Areas," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2283, pp. 74–80, Dec. 2012.
- [5] R. M. Newton and W. H. Thomas, "Design of school bus routes by computer," *Socioecon. Plann. Sci.*, vol. 3, no. 1, pp. 75–85, Jun. 1969.
- [6] G. Dantzig and J. Ramser, "The truck dispatching problem," *Manage. Sci.*, 1959.
- [7] A. Fügenschuh, A. Martin, and P. Stöveken, "Integrated optimization of school starting times and public bus services," *Oper. Res. Proc.*, 2005.
- [8] R. . Corberán, A.; Fernández, E.; Laguna, M.; Martí, "Heuristic solutions to the problem of routing school buses with multiple objectives," *J. Oper. Res. Soc.*, vol. 53, 2000.
- [9] B. Kamali;, S. J. M. And, and E. A. Pohl, "An Analysis of Special Needs Student Busing," *J. PUBLIC Transp.*, vol. 16.
- [10] K. Kalogirou, E. Chalkia, E. Bekiaris, and F. Diederichs, "An Application for the Information of Children According their School Transportation," *Procedia - Social and Behavioral Sciences*, vol. 48. pp. 363–372, 2012.
- [11] X. Chen, Y. Kong, L. Dang, Y. Hou, and X. Ye, "Exact and Metaheuristic Approaches for a Bi-Objective School Bus Scheduling Problem.," *PLoS One*, vol. 10, no. 7, p. e0132600, Jan. 2015.
- [12] J. Braca, J. Bramel, B. Posner, and D. Simchi-Levi, "A computerized approach to the New York City school bus routing problem," *IIE Trans.*, vol. 28, 8, pp. 693–702, 1997.
- [13] J. Park, H. Tae, and B. I. Kim, "A post-improvement procedure for the mixed load school bus routing problem," *Eur. J. Oper. Res.*, 2012.

[14] T. Feo and M. Resende, "Greedy randomized adaptive search procedures," *J. Glob. Optim.*, 1995.

[15] G. Croes, "A method for solving traveling-salesman problems," *Oper. Res.*, 1958.



Vilson Soares de Siqueira Graduated in Information Systems the State University of Goias (2008). Specialization in Computer Networks For Anhanguera Faculty of Anapolis (2010). Master student in Computational Modeling and Systems at the Federal University of Tocantins (2013). Is currently a professor of Computer Science in Federal Institute of Tocantins, Campus Araguatins, Brazil and Computing Project area coordinator, the institutionally Scholarship Program for Initiation to Teaching, the Federal Institute of Science and Technology Tocantins, Brazil. Born in Ceres, Goias in 1981.



Fernando Jorge Ebrahim Lima e Silva Graduated in Information Systems in the Uninorte (2000). Specialization in Computers Network at the Federal University of Rio Grande do Norte(2003). Master student ins Computational Modeling and Systems at the Federal University of Tocantins (2013) Is currently a professor of Computer Science in Federal Institute of Tocantins, Campus Palmas, Brazil. Born in Recife, Pernambuco in 1975.



Elvis Nascimento da Silva Graduated in Technology in the Data Processing Foundation University of Tocantins (2001), MBA in Information Technology Management for Tocantinense Institute of Postgraduate - ITOP (2008) and specialization in the database by the Catholic University of Tocantins FACT (2009). He is currently a Master in Computer Modelling Systems, Federal University of Tocantins - UFT. He is Professor of Basic Education, Technical and Technological of the Federal Institute of Education, Science and Technology of Tocantins IFTO. It has experience in computer science, with emphasis on

Information Technology and Database.



Raimunda Vieira Santos da Silva A master's degree in Production Engineering and specialized in Audit and Government Management from PUC-GO (2014). Graduated in Business Administration from Federal University of Pará (2004). It has experience in strategic and environmental management; bidding; audit and public finance. Currently working on the environmental agenda and strategic planning at the Federal Institute of Education, Science and Technology of Tocantins. Born in Tocantinópolis, To, Brazil in 1965.



Marcelo Lisboa Rocha is Graduated in Computer Science from the Catholic University of Petropolis (1994), Master in Computer Science from Federal University Fluminense (1997), Master in Electrical Engineering from the Federal University of Rio de Janeiro (1999) and Doctorate in Electrical Engineering from the Federal University of Rio Janeiro (2008). Is currently reviewer the periodic INFOCOMP Journal of Computer Science and associate Teacher 4 of the Federal University of Tocantins, Brazil. It has experience in computer science, acting on the following topics: metaheuristics, combinatorial optimization,

mathematical programming, computer networks and high-performance computing. Born in Rio de Janeiro in 1972.