

Algoritmos Baseados na Metaheurística GRASP para Implantação de Unidades de Comunicação em Redes Veiculares Garantindo Qualidade de Serviço

Taís Rocha Silva, João Fernando M. Sarubbi, Flavio V. Cruzeiro Martins

Departamento de Computação – Centro Federal de Educação Tecnológica de Minas Gerais
Belo Horizonte – MG – Brazil

silva.taisrocha@gmail.com, joao@decom.cefetmg.br,
flaviocruzeiro@decom.cefetmg.br

Cristiano Maciel da Silva

Departamento de Tecnologia – Universidade Federal de São João Del Rey
Ouro Branco – MG – Brazil

cristiano@ufsj.edu.com

RESUMO

Neste trabalho são propostos três novos algoritmos: (i) DL_GRASP; (ii) Delta-g_GRASP; e, (iii) Delta-r_GRASP. Tais algoritmos foram desenvolvidos a partir da metaheurística GRASP para solucionar o problema da distribuição de unidades de comunicação em redes veiculares V2I (*Vehicle-to-Infrastructure*). O objetivo consiste em encontrar uma quantidade mínima de unidades de comunicação que atenda aos requisitos de desempenho especificados pela Deposição $\Delta_{\rho_2}^{\rho_1}$. A Deposição $\Delta_{\rho_2}^{\rho_1}$ é uma métrica utilizada para avaliar a qualidade no fornecimento do serviço de comunicação em redes V2I. Cada um dos algoritmos propostos são comparados, respectivamente, aos algoritmos DL, Delta-g e Delta-r. Os resultados demonstram que, para as combinações testadas, os algoritmos propostos utilizam um número sempre menor de unidades de comunicação, considerando os mesmos requisitos de desempenho.

PALAVRAS CHAVE. Redes Veiculares, Estratégias de Deposição, Metaheurística GRASP.

ABSTRACT

In this work we propose three new algorithms: (i) DL_GRASP; (ii) Delta-g_GRASP; and, (iii) Delta-r_GRASP. These new algorithms are GRASP like based algorithms developed for solving the allocation of Roadside Units (RSUs) in a Vehicle-to-Infrastructure (V2I) Vehicular Network. The goal consists in find the minimum set of RSUs in order to meet a Deployment $\Delta_{\rho_2}^{\rho_1}$. The Deployment $\Delta_{\rho_2}^{\rho_1}$ is a metric for specifying minimal communication guarantees from the infrastructure supporting the Vehicular Network. Each of the proposed algorithms are compared, respectively, to algorithms DL, Delta-g and Delta-r. The results shows, for tested combinations, that proposed algorithms always requires less Roadside Units in order to achieve the same deployment efficiency.

KEYWORDS. Vehicular Network. Deployment Strategies. GRASP Heuristic.

1. Introdução

O estudo sobre redes veiculares (VANETs, do inglês *Vehicle Ad Hoc Networks*) [Hartenstein e Laberteaux, 2008] constitui um importante segmento de pesquisa há alguns anos. Tal importância é atribuída ao fato de que as VANETs, representantes de um tipo específico de rede móvel, desempenham o papel de ferramenta central em Sistemas Inteligentes de Transporte, sendo responsáveis por uma comunicação bastante eficiente para fins de controle de tráfego, segurança veicular e até mesmo em aplicações direcionadas a entretenimento [Aissaoui et al., 2014].

Em uma rede veicular é possível estabelecer comunicação sob duas formas distintas: por meio de redes V2V (*Vehicle-to-Vehicle*) ou V2I (*Vehicle-to-Infrastructure*) [Blum et al., 2004]. Nas redes V2V a conexão ocorre através do chamado *ad-hoc* puro, em que não há qualquer infraestrutura de apoio e a comunicação é realizada de veículo para veículo. As redes V2I, por sua vez, ocorrem através de conexões entre veículos e unidades de comunicação (RSUs, do inglês *Roadside Units*), que são infraestruturas fixas de suporte posicionadas ao longo de vias.

O emprego de redes V2I, segundo diversos estudos, assegura a eficiência da comunicação em áreas de grande extensão e baixo fluxo de veículos, onde a rede *ad-hoc* apresenta limitações de conectividade. Entretanto, a utilização de RSUs agrega custos significativos de instalação, o que representa um inconveniente financeiro relevante. Sendo assim, existe um impasse entre assegurar o nível de eficiência nas redes veiculares, através da distribuição das unidades de comunicação em larga escala, e restringir os custos gerados pelo processo de instalação dessas RSUs. Sob esse aspecto, é necessário que as unidades de comunicação sejam posicionadas estrategicamente, de forma a garantir uma qualidade mínima no fornecimento do serviço e uma redução do número de RSUs instaladas, a fim de minimizar os custos.

A Deposição $\Delta_{\rho_2}^{\rho_1}$ [Silva e Meira, 2015a] é uma métrica utilizada para avaliar o desempenho de redes V2I e estabelecer os níveis de exigência da comunicação. Por meio dessa ferramenta, o desempenho é especificado em termos da probabilidade e da duração da conexão entre os veículos e as RSUs, de forma a garantir que uma fração esperada de veículos permaneçam conectados durante um tempo suficiente para receber as informações necessárias.

Com o intuito de encontrar uma distribuição das unidades de comunicação que atenda às necessidades apresentadas pela Deposição $\Delta_{\rho_2}^{\rho_1}$, alguns autores propuseram diferentes heurísticas como solução, dentre os quais ressaltamos três heurísticas gulosas. Designadas como DL [Trullols et al., 2010], Delta-g [Silva e Meira, 2015a] e Delta-r [Sarubbi e Silva, 2016], tais heurísticas são fundamentalmente análogas, mas empregam diferentes estratégias na construção de uma solução que corresponda aos requisitos de qualidade especificados.

Este trabalho propõe novos algoritmos para solucionar o problema da distribuição de unidades de comunicação, a fim de que sejam garantidos os níveis de exigência estabelecidos pela Deposição $\Delta_{\rho_2}^{\rho_1}$. Cada um desses algoritmos foi desenvolvido a partir da metaheurística GRASP [Feo e Resende, 1995], empregando na fase de construção um dos algoritmos gulosos mencionados. Os resultados atingidos pelos algoritmos propostos neste trabalho apresentam um número menor de unidades de comunicação para alcançar os mesmos níveis de desempenho, quando comparados aos resultados obtidos pelos algoritmos estritamente gulosos correspondentes.

Este trabalho está organizado da seguinte forma. A seção 2 apresenta um apanhado sobre trabalhos relacionados. A seção 3 aponta o método utilizado para representação de malhas rodoviárias. A seção 4 formaliza a Deposição $\Delta_{\rho_2}^{\rho_1}$. A seção 5 apresenta os algoritmos de referência. A seção 6 descreve a metaheurística GRASP, bem como a proposta de solução deste trabalho. A seção 7 apresenta os experimentos realizados com os respectivos resultados. Por fim, a seção 8 conclui o trabalho.

2. Trabalhos Relacionados

Diversos autores tem estudado a alocação de unidades de comunicação em redes veiculares. O trabalho de Zheng et al. [2009] minimiza o número de RSUs garantindo que cada caminho de

tamanho α tenha pelo menos uma unidade de comunicação. A probabilidade de contato é considerada em alguns trabalhos como: Zheng et al. [2010], que apresentam a avaliação de uma estratégia de deposição através da possibilidade de contato medindo a fração da distância (ou tempo) que o veículo está em contato com a RSU. Enquanto Trullols et al. [2010] formularam a alocação de RSUs como um Problema de Máxima Cobertura, Lee e Kim [2010] propuseram uma heurística gulosa para alocar as RSUs objetivando aumentar a conectividade dos veículos e reduzir conexões.

A caracterização da mobilidade veicular é estudada por Li et al. [2014], que examinam os limites de previsibilidade de redes veiculares de larga escala usando a teoria da entropia. Da mesma forma, Zhang et al. [2013] investigam a distribuição de taxis na cidade de Shanghai. Os autores discutem como as rodovias mais populares impactam na distribuição de tempos de contatos das RSUs. Tonguz e Viriyasitavat [2013] propuseram a utilização de veículos como unidades de comunicação usando uma rede biologicamente inspirada.

Em relação a Deposição $\Delta_{\rho_2}^{\rho_1}$, apenas três trabalhos apresentaram algoritmos para esse problema. Silva e Meira [2015b] apresentaram uma estratégia gulosa que leva em consideração o tempo de contato de cada veículo em cada célula, a qual chamaram de Delta-g, e compararam esse algoritmo com o algoritmo DL proposto por Trullols et al. [2010]. Sarubbi e Silva [2016] apresentaram uma heurística gulosa que leva em consideração o tempo de contato absoluto, denominada Delta-r, e compararam com os algoritmos DL e Delta-g. Além desses, Sarubbi et al. [2016] desenvolveram um algoritmo genético para o problema que obteve bons resultados mas que foi testado somente para 3 combinações de ρ_1 e ρ_2 .

3. Representação de Malhas Rodoviárias

No âmbito geral, malhas rodoviárias assumem formas e topologias arbitrárias, características que trazem grande complexidade à representação. Em busca de superar essa complexidade, optou-se por empregar a mesma estratégia utilizada por Sarubbi e Silva [2016], em que foi realizado o particionamento da área de interesse em um conjunto de células adjacentes de mesmo tamanho, tal como uma matriz. De acordo com essa metodologia, as células podem assumir dimensões variadas, segundo as necessidades do projetista e a disponibilidade de recursos computacionais para processar o cenário do tráfego. Ao aumentar a dimensão da célula, perde-se na precisão da representação do local, restringindo o número de possíveis pontos para a instalação de uma unidade de comunicação. Isso implica em uma redução considerável do esforço computacional para a obtenção de melhores localidades, além de fazer com que a complexidade para gerar uma solução seja independente do tamanho da região modelada.

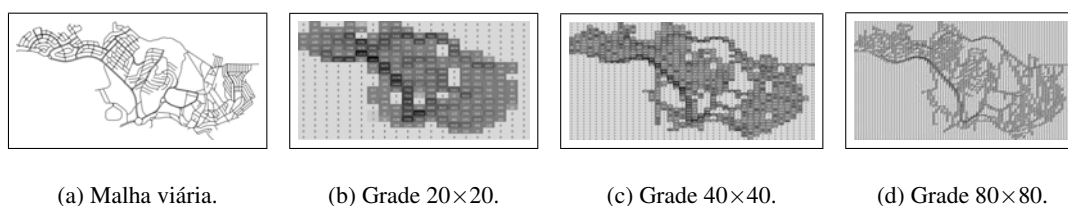


Figura 1: Representação da malha rodoviária em células.

As Figuras (1b) a (1d) ilustram diferentes níveis de precisão na modelagem de uma malha rodoviária real, representada pela figura (1a).

4. Definição do problema: Deposição $\Delta_{\rho_2}^{\rho_1}$

A Deposição $\Delta_{\rho_2}^{\rho_1}$ é uma métrica para avaliar a qualidade no fornecimento do serviço de comunicação. De acordo com essa métrica, o desempenho da rede é especificado em termos de dois parâmetros: ρ_1 e ρ_2 . O parâmetro ρ_1 indica a duração mínima estabelecida para a conexão

V2I, enquanto o parâmetro ρ_2 denota a fração de veículos que deve satisfazer a conectividade de duração ρ_1 . Dessa forma, a Deposição $\Delta_{\rho_2}^{\rho_1}$ pode ser definida como um conjunto de unidades de comunicação que permite que ρ_2 por cento dos veículos estejam conectados durante, pelo menos, ρ_1 por cento do tempo total de viagem. De acordo com a aplicação desejada, é possível a escolha de diferentes valores para os parâmetros ρ_1 e ρ_2 . Consideremos, como exemplo, uma deposição $\Delta_{0.5}^{0.3}$, ela indica que 50% do número total de veículos estejam se comunicando com uma ou mais RSUs durante 30% de toda a viagem.

É necessário ressaltar que a Deposição $\Delta_{\rho_2}^{\rho_1}$ é classificada como uma métrica de qualidade, uma vez que especifica unicamente o nível de exigência na comunicação dos veículos com as infraestruturas de suporte, sem determinar a forma utilizada para alcançar tal desempenho.

A definição 1 apresenta uma descrição formal para a Deposição $\Delta_{\rho_2}^{\rho_1}$.

Definição 1 (Deposição $\Delta_{\rho_2}^{\rho_1}$) *Seja M a representação da malha rodoviária, e $V = \{v_1, v_2, \dots, v_n\}$ um conjunto de veículos viajando por M . Seja $T = \{U_1, U_2, \dots, U_n\}$ o conjunto que representa a trajetória de cada veículo $v \in V$. Logo, cada $v_k \in V$ possui uma trajetória específica $U_k \in T$. Cada $U \in T$ representa uma trajetória de células $U_k = \{u_1^k, u_2^k, \dots, u_n^k\}$ percorrida por um veículo v_k durante a viagem. Seja $C \subset V$ o conjunto de veículos v_k que estejam conectados durante um percentual da viagem $\geq \rho_1 \forall u \in U_k$. Essa rede veicular é considerada $\Delta_{\rho_2}^{\rho_1}$ sempre que $\frac{|C|}{|V|} \geq \rho_2$.*

O objetivo consiste em encontrar o subconjunto mínimo de células onde ρ_2 por cento dos veículos estão conectados durante ρ_1 por cento do tempo total de viagem. Para esse problema Sarubbi e Silva [2016] apresentam um modelo matemático de programação linear. Utiliza-se o seguinte conjunto de variáveis para esse modelo:

$$a_u = \begin{cases} 1 & \text{se a célula } u \text{ pertence à solução} \\ 0 & \text{caso contrário.} \end{cases}$$

$$v_k = \begin{cases} 1 & \text{se o veículo } k \text{ pertence à solução} \\ 0 & \text{caso contrário.} \end{cases}$$

E segue o conjunto de parâmetros:

t_{uk} : tempo de permanência do veículo k na célula u
 tv_k : tempo total de viagem do veículo k

O modelo matemático é descrito por:

$$\min \sum_{u \in U} a_u \tag{1}$$

$$\sum_{u \in U} (t_{uk}/tv_k) a_u \geq \rho_1 v_k \quad \forall k \in V \tag{2}$$

$$\sum_{k \in V} v_k \geq \rho_2 |V| \tag{3}$$

$$a_u \in \{0, 1\} \quad \forall u \in U \tag{4}$$

$$v_k \in \{0, 1\} \quad \forall k \in V \tag{5}$$

O objetivo expresso pela função (1) está em minimizar o número de unidades de comunicação, isto é, o número de células selecionadas. A restrição (2) garante que um veículo é considerado parte da solução, ou seja, que atingiu a meta de desempenho, somente quando permanece

conectado durante ρ_1 por cento do tempo total de viagem. A restrição (3) garante que uma fração de ρ_2 por cento dos veículos estejam submetidos à restrição (2) para compor a solução. As demais restrições, (4) e (5), apenas integram o modelo.

5. Algoritmos de *baseline*

Nesta seção serão apresentados os algoritmos previamente propostos para solucionar a Deposição $\Delta_{\rho_2}^{\rho_1}$, tomados como referência para a composição deste trabalho. Tais algoritmos foram projetados como heurísticas gulosas, sendo diferenciados pela estratégia definida no processo de obtenção das melhores localidades para a instalação das unidades de comunicação. Os algoritmos são denominados DL, Delta-g e Delta-r.

5.1. Algoritmo DL

O algoritmo DL foi proposto por Trullols et al. [2010] como precursor na aplicação de estratégias gulosas para solução da Deposição $\Delta_{\rho_2}^{\rho_1}$. O objetivo desse algoritmo é adicionar iterativamente ao conjunto solução a célula percorrida pela maior densidade de veículos. À medida que cada célula é adicionada, são verificados todos os veículos que atingiram o parâmetro ρ_1 . Caso um veículo k tenha atingido o tempo mínimo de conexão estabelecido, este é retirado do conjunto de veículos utilizado para computar a próxima iteração. O processo descrito é repetido enquanto o percentual ρ_2 de veículos conectados não foi alcançado. No algoritmo (1) esta apresentado o pseudo-código referente ao algoritmo DL.

Algoritmo 1: DL

```

Data:  $M, V, T, \rho_1, \rho_2$ 
1 Solucao  $\leftarrow \emptyset$ ;
2 while  $\frac{|C|}{|V|} < \rho_2$  do
3    $\varphi \leftarrow$  Maior_Densidade( $M$ ); /* Seleciona célula mais densa */
4   Solucao  $\leftarrow \varphi$ ; /* Adiciona a célula ao conjunto solução */
5    $M \leftarrow M - \varphi$ ; /* Remove célula selecionada do conjunto */
6    $C \leftarrow$  Conectados( $M, V, T, Solucao, \rho_1$ );
   /* Verifica veículos que atingiram  $\rho_1$  */
7 end
8 return Solucao;

```

5.2. Algoritmo Delta-g

O algoritmo Delta-g, proposto por Silva e Meira [2015a], emprega a estratégia gulosa para seleção de células, baseando-se no tempo absoluto do contato entre veículos descobertos e unidades de comunicação. Isso significa que este algoritmo considera mais relevante contabilizar o tempo de permanência de cada veículo em uma célula, para avaliar o nível de contribuição dessa célula ao conjunto solução, do que analisar a densidade de veículos que nela percorrem. Este é um processo iterativo, realizado de forma análoga ao descrito na seção (5.1). O algoritmo (2) apresenta um pseudo-código referente ao algoritmo Delta-g.

Algoritmo 2: Delta-g

```

Data:  $M, V, T, \rho_1, \rho_2$ 
1 Solucao  $\leftarrow \emptyset$ ;
2 while  $\frac{|C|}{|V|} < \rho_2$  do
3    $\varphi \leftarrow$  Maior_Tempo_Absoluto( $M, V - C$ );
   /* Seleciona célula com maior tempo absoluto de contato */
4   Solucao  $\leftarrow \varphi$ ; /* Adiciona a célula ao conjunto solução */
5    $M \leftarrow M - \varphi$ ; /* Remove célula selecionada do conjunto */
6    $C \leftarrow$  Conectados( $M, V, T, Solucao, \rho_1$ );
   /* Verifica veículos que atingiram  $\rho_1$  */
7 end
8 return Solucao;

```

5.3. Algoritmo Delta-r

O algoritmo Delta-r, proposto por Sarubbi e Silva [2016], é uma heurística gulosa e iterativa, assim como os algoritmos apresentados nas seções (5.1) e (5.2). Este algoritmo apresenta uma estratégia semelhante ao algoritmo Delta-g, pois ambos avaliam o tempo na composição do conjunto solução. Contudo, a abordagem trazida pela heurística Delta-r diz respeito ao tempo relativo de conexão entre veículos e unidades de comunicação. Para o algoritmo Delta-r, portanto, o critério de seleção é definido em função do somatório dos percentuais de tempo de conexão de cada veículo k em relação ao tempo total da viagem que realizam.

Sendo assim, a cada iteração é adicionada ao conjunto solução a célula que representa o maior somatório dos percentuais de tempo relativo, referentes aos veículos que nela percorrem. A partir da célula acrescida, são registrados os veículos que passaram a ser atendidos em relação ao parâmetro ρ_1 , até que o total de veículos atingidos alcance o valor determinado pelo parâmetro ρ_2 .

O procedimento Delta-r, é representado pelo algoritmo (3). Durante a execução, são utilizadas as seguintes matrizes: (i) $Mperc$ para cada célula u e para cada veículo k , armazena o percentual do tempo relativo de conexão entre o veículo k e a célula u ; (ii) $SMperc$ para cada célula u , soma o percentual de tempo relativo de todos os veículos que nela percorrem.

Algoritmo 3: Procedimento Delta-p

```

Data:  $M, V, T, \rho_1, \rho_2, Atualiza$ 
1  $Solucao \leftarrow \emptyset$ ;
2  $C \leftarrow \emptyset$ ;
3 while  $\frac{|C|}{|V|} < \rho_2$  do
4    $\theta \leftarrow \text{CalculaMperc}(M, V)$ ;
   /* Calcula Mperc de cada célula para veículos restantes */
5    $\eta \leftarrow \text{CalculaSMperc}(\theta)$ ;
   /* Calcula a soma de Mperc de todos os veículos */
6    $\varphi \leftarrow \text{MaiorSMperc}(\eta)$ ;
   /* Seleciona a célula com o maior SMperc */
7    $Solucao \leftarrow \varphi$ ; /* Adiciona a célula ao conjunto solução */
8    $M \leftarrow M - \varphi$ ; /* Remove célula selecionada do conjunto */
9    $C \leftarrow \text{Conectados}(M, V, T, Solucao, \rho_1)$ ;
   /* Verifica veículos que atingiram  $\rho_1$  */
10   $V \leftarrow V - C$ ; /* Remove veículos cobertos do conjunto */
11  if  $Atualiza = true$  then
12     $\text{AtualizaMperc}(M, V, T, \rho_1, Solucao)$ ;
    /* Altera Mperc de veículos cobertos para valor zero */
13  end
14 end
15 return  $Solucao$ ;

```

6. Metaheurística GRASP

Nesta seção, será apresentada a estratégia proposta por este trabalho para solucionar o problema da distribuição de unidades de comunicação. Como metodologia, foi utilizada a metaheurística GRASP, *Greedy Randomized Adaptive Search Procedure*, cuja tradução ao português é "procedimento de busca guloso, adaptativo e aleatório". GRASP é uma técnica iterativa de amostragem aleatória, na qual cada iteração fornece uma solução para o problema modelado. A solução preponderante sobre todas as iterações é mantida como resultado final. O GRASP é composto por duas etapas: a primeira, denominada Fase de Construção, gera uma solução inicial viável, através de uma função gulosa e aleatória; a segunda consiste em aplicar um procedimento de busca local para a solução construída. Ambas as etapas são executadas a cada iteração [Feo e Resende, 1995].

Um pseudo-código genérico para o GRASP está apresentado pelo algoritmo (4), em seguida serão detalhadas cada uma das etapas que compoem o método.

Algoritmo 4: GRASP

```

Data:  $\alpha, max\_Iter$ 
1 for  $it \leftarrow 1$  até  $max\_Iter$  do
2   Solução  $\leftarrow$  Fase_Construção( $\alpha$ );
3   Busca_Local(Solução);
4   if Solução_Encontrada_for_Melhor() then
5     Atualize_Melhor_Solução(Solução, melhor_Solucao);
6   end
7 end
8 return melhor_Solucao;

```

6.1. Fase de Construção

A fase de construção é um processo iterativo, cuja finalidade é aliar aleatoriedade a um critério guloso para gerar soluções viáveis. Cada um dos elementos pertencentes a uma lista de candidatos é submetido a uma função gulosa determinada, de forma a avaliar o custo associado com a incorporação do elemento à solução em construção. Os elementos com os menores custos compoem a chamada lista restrita de candidatos (LRC), de onde serão selecionados os membros integrantes do conjunto solução.

O estágio de seleção de tais elementos ocorre a partir do valor determinado para o parâmetro α , referente ao nível de aleatoriedade utilizado na fase de construção. Esse parâmetro pode assumir valores entre 0 e 1, sendo que para α igual a 0 tem-se uma construção puramente gulosa e para α igual a 1 tem-se uma construção puramente aleatória.

Em decorrência da infinitude de possíveis valores associados ao parâmetro α , são permitidas diversas soluções diferentes para o mesmo critério de desempenho, estabelecido pelos padrões da Deposição $\Delta_{\rho_2}^{\rho_1}$. Sendo assim, a escolha do valor de α é um fator relevante na definição da qualidade dos elementos candidatos, pois a cada iteração será selecionado um elemento dentre os melhores classificados, mas não necessariamente o melhor desses, em função do nível de aleatoriedade definido.

Segundo Feo e Resende [1995], o GRASP é uma heurística adaptativa, pois os custos associados com a incorporação de cada elemento são atualizados em todas as iterações da fase de construção. Esse processo baseia-se no elemento selecionado pela iteração vigente, refletindo nas próximas iterações. Deste modo, torna-se essencial ao processo de desenvolvimento de uma solução utilizando a metodologia GRASP, escolher a função gulosa que se adequa melhor ao domínio do problema, a fim de obter-se melhores resultados.

Tendo em vista a propriedade adaptativa da heurística GRASP e as características inerentes ao problema, este trabalho definiu como estratégia utilizar individualmente as heurísticas DL, Delta-g e Delta-r como funções gulosas para a fase de construção. Essa escolha deu origem aos algoritmos DL_GRASP, Delta-g_GRASP e Delta-r_GRASP, respectivamente.

Dessa forma, os algoritmos propostos por este trabalho são aplicações da metodologia GRASP, de modo que a função de construção da LRC consiste em executar uma das heurísticas gulosas apresentadas. Através desses algoritmos, a fase de construção aplicada ao problema definido pela Deposição $\Delta_{\rho_2}^{\rho_1}$ está submetida aos critérios de desempenho especificados pelos parâmetros ρ_1 e ρ_2 , que determinam o término deste processo iterativo, ou seja, quando o conjunto solução está completo.

Portanto, a solução expressa pela fase de construção representa um conjunto de células onde serão posicionadas unidades de comunicação, a fim de que a Deposição $\Delta_{\rho_2}^{\rho_1}$ seja atendida.

6.2. Busca Local

A fase de construção realizada pela metodologia GRASP não assegura que as soluções inicialmente geradas correspondam a pontos de ótimo local, em relação às soluções possivelmente

Algoritmo 5: Fase de Construção

Data: α

```

1 Solucao  $\leftarrow \emptyset$ ;
2 while Solução não esta completa do
3   |   Constroi_LRC( $\alpha$ );
4   |   Celula_Seleccionada  $\leftarrow$  Escolhe_Aleatoriamente_Elemento(LRC);
5   |   Solucao  $\leftarrow$  Solucao  $\cup$  Celula_Seleccionada;
6 end
7 return Solucao;
```

geradas pela vizinhança. Isso significa que, a princípio, uma solução inicialmente construída é sempre passível de ser melhorada. Um algoritmo de busca local funciona de forma iterativa, substituindo sucessivamente a solução atual por uma solução melhor, pertencente à vizinhança da solução atual. O algoritmo termina quando não é possível obter uma solução melhor do que a atual, ou seja, quando o ponto de ótimo local já foi atingido [Feo e Resende, 1995]. O algoritmo de busca local está representado de forma genérica pelo pseudo-código (6).

Algoritmo 6: Busca Local

Data: *Solucao*

```

1 while Solução não é localmente ótima do
2   |   Encontre_Melhor_Solução();
3   |   Atualize_Solução();
4 end
```

O procedimento de busca local envolve questões fundamentais de projeto, dentre as quais destaca-se a definição da vizinhança e a estratégia de busca na vizinhança.

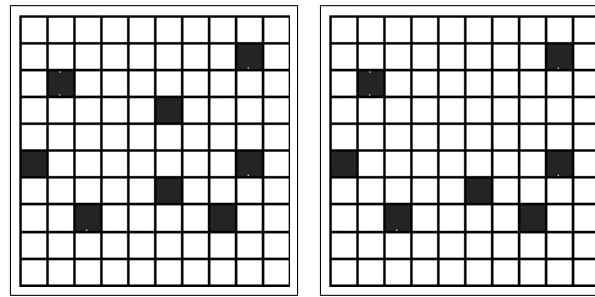
Em relação à Deposição $\Delta_{\rho_2}^{\rho_1}$, o conjunto solução é determinado como um conjunto de células em que serão posicionadas unidades de comunicação, de acordo com os parâmetros ρ_1 e ρ_2 estabelecidos. A vizinhança, por sua vez, foi definida como um conjunto de células que diferem em uma unidade de comunicação da solução atual. A estratégia definida para a busca local foi originada da percepção de que é possível, a partir do conjunto solução gerado pela fase de construção, retirar uma ou mais unidades de comunicação, de forma que os critérios definidos pela Deposição $\Delta_{\rho_2}^{\rho_1}$ ainda sejam garantidos. Existem duas situações que possibilitam essa retirada, são elas:

1. Todos os veículos cobertos pela célula atual são suficientemente atendidos por outras células, mantendo a condição estabelecida pelo parâmetro ρ_1 .
2. Existem veículos que dependem do tempo de conexão realizado com a célula atual para garantir a condição estabelecida pelo parâmetro ρ_1 , entretanto, o percentual referente a esses veículos é igual ou inferior ao percentual de veículos cobertos que excediam a condição estabelecida pelo parâmetro ρ_2 .

A Figura 2 apresenta um exemplo de solução antes e depois da aplicação da busca local proposta. Nota-se que uma célula deixa de ser contemplada por uma unidade de comunicação depois de aplicada a busca local. É importante lembrar que uma unidade de comunicação somente é retirada se as restantes atendem os critérios associados aos parâmetros ρ_1 e ρ_2 . Também é importante salientar que, dessa forma, os algoritmos implementados neste trabalho somente "percorrem" soluções viáveis.

7. Resultados

Nesta seção são apresentados os experimentos realizados para comparar os algoritmos puramente gulosos: DL, Delta-g e Delta-r com os algoritmos propostos neste trabalho: (i) DL_GRASP;



(a) Antes da Busca Local (b) Depois da Busca Local

Figura 2: Representação da Busca Local.

(ii) Delta-g_GRASP; e, (iii) Delta-r_GRASP. Os experimentos são baseados no rastro (*trace*) realístico de mobilidade da cidade de Colônia¹ (Alemanha). O (*trace*) é composto por 10.000 segundos de tráfego com 75.516 veículos. Todos os experimentos são interpretados através do simulador SUMO (<http://sumo-sim.org>) que processa o cenário da cidade de Colônia e fornece as localidades de cada veículo (registros de mobilidade T) ao longo do tempo. Um Programa de Particionamento lê os registros de mobilidade T e particiona o cenário em um gride de $\phi \times \phi$ células, traduzindo os registros de mobilidade de coordenadas cartesianas para coordenadas de grade. Neste trabalho foi utilizado $\phi = 100$.

Neste trabalho são apresentados resultados para 25 diferentes combinações, formadas a partir dos parâmetros $\rho_1 = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ e $\rho_2 = \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Cada uma dessas combinações foi executada para 9 algoritmos:

- DL, Delta-g e Delta-r, implementados como citam seus autores;
- DL_GRASP, Delta-g_GRASP, e Delta-r_GRASP, utilizando como parâmetros $max_Iter = 500$ e $\alpha = \{0.02, 0.05, 0.1, 0.2\}$, sendo representado graficamente o valor mínimo de unidades de comunicação encontrado, considerando a execução de 11 instâncias para cada combinação.
- DL_BL, Delta-g_BL e Delta-r_BL, que são casos especiais dos algoritmos DL_GRASP, Delta-g_GRASP e Delta-r_GRASP, respectivamente, onde $max_Iter = 1$ e $\alpha = \{0.0\}$. Portanto, esses são os algoritmos estritamente gulosos, acrescidos da busca local proposta por este trabalho.

7.1. Comparando Algoritmos DL, DL_BL e DL_Grasp

Nesta seção são apresentados os resultados comparando os algoritmos DL, DL_BL e DL_Grasp. A figura (3) apresenta esses ganhos em relação ao algoritmo DL. No eixo x estão representadas cada uma das instâncias testadas com seus respectivos parâmetros ρ_2 e ρ_1 . No eixo y estão representados os ganhos percentuais relativos a solução encontrada pelo algoritmo DL para os algoritmos DL_BL e DL_GRASP. Esses ganhos são calculados pela seguinte fórmula:

$$Ganho = \frac{Sol_DL - Solucao}{Sol_DL} \times 100\% \quad (6)$$

onde Sol_DL representa a solução (número de RSUs) dada pelo algoritmo DL e $Solucao$ a solução (número de RSUs) encontrada por um dos algoritmos DL_BL ou DL_GRASP.

¹Disponível em <http://kolntrace.project.citi-lab.fr/>

Analisando o gráfico, nota-se que a solução encontrada pelo algoritmo DL_BL sempre apresenta um ganho em relação a solução encontrada pelo algoritmo DL original. Esse comportamento já era esperado pois o algoritmo DL_BL, em suma, trata-se da aplicação da busca local proposta no algoritmo DL estritamente guloso. Tal desempenho será igualmente verificado nos gráficos apresentados pelas seções que seguem.

Em relação ao algoritmo DL_GRASP, verifica-se que os ganhos são ainda maiores, isto é, são encontradas soluções com menos RSUs que as soluções encontradas pelo algoritmo DL_BL. Esses resultados apontam a relevância em aplicar-se um percentual de aleatoriedade na construção de uma solução inicial, destacando todas as etapas que compõe a metodologia GRASP, e não apenas a utilização de uma técnica de busca local.

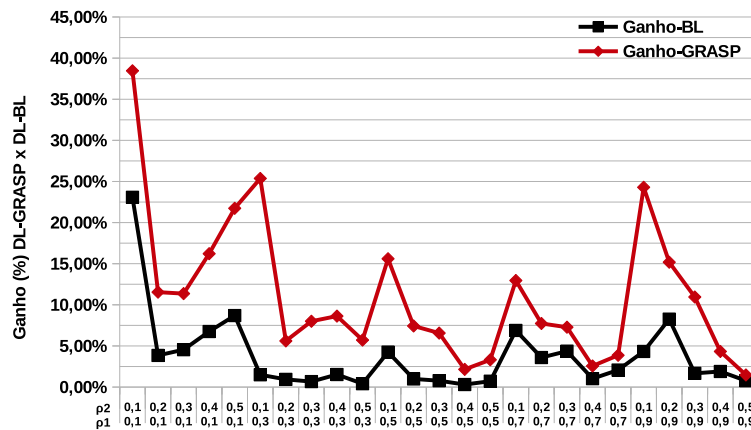


Figura 3: Ganho Percentual - Algoritmos DL_BL e DL_GRASP - Apresentam o ganho percentual em relação ao algoritmo DL original. O eixo x apresenta as 25 instâncias testadas (respectivamente ρ_2 e ρ_1) e o eixo y o ganho relativo. Quanto maior o ganho significa que o algoritmo apresentado conseguiu soluções melhores, isto é, com menos unidades de comunicação.

7.2. Comparando Algoritmos Delta-g, Delta-g_BL e Delta-g_GRASP

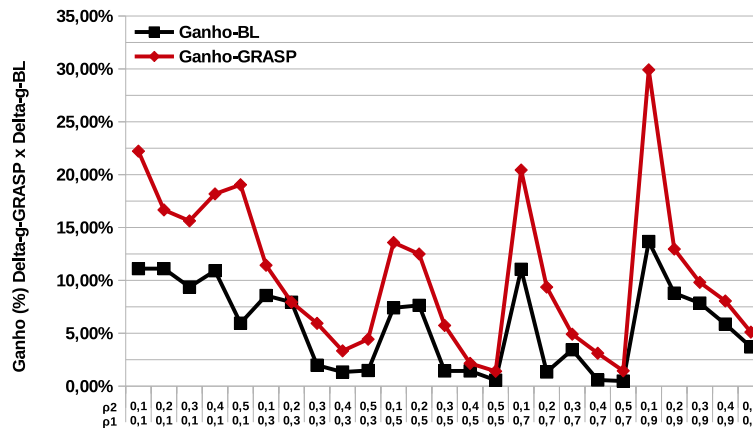


Figura 4: Ganho Percentual - Algoritmos Delta-g_BL e Delta-g_GRASP - Apresentam o ganho percentual em relação ao algoritmo DL original. O eixo x apresenta as 25 instâncias testadas (respectivamente ρ_2 e ρ_1) e o eixo y o ganho relativo. Quanto maior o ganho significa que o algoritmo apresentado conseguiu soluções melhores, isto é, com menos unidades de comunicação.

Nesta seção são apresentados os resultados comparando os algoritmos Delta-g, Delta-g_BL e Delta-g_GRASP. A Figura (4) apresenta esses ganhos em relação ao algoritmo Delta-g. Os eixos e a fórmula para calcular o ganho percentual são os mesmos da seção (7.1).

Os resultados apresentados na Figura (4) mostram que os algoritmos Delta-g_BL e Delta-g_GRASP apresentam ganhos em relação ao algoritmo Delta-g original. Nota-se que esses ganhos são maiores principalmente nas instâncias em que $\rho_2 = 0.1$, mantendo bons resultados para todos os valores associados a ρ_1 . Contudo, nota-se também que os ganhos tendem a diminuir expressivamente com o aumento do parâmetro ρ_2 , o que demonstra que a metodologia GRASP, aplicada ao contexto deste problema, se adapta bem sob exigências maiores de tempo, mas apresenta grande fragilidade em relação a percentuais maiores de veículos a serem atendidos.

7.3. Comparando Algoritmos Delta-r, Delta-r_BL e Delta-r_GRASP

Nesta seção são apresentados os resultados comparando os algoritmos Delta-r, Delta-r_BL e Delta-r_GRASP. A Figura (5) apresenta esses ganhos em relação ao algoritmo Delta-r. Os eixos e a fórmula para calcular o ganho percentual são os mesmos das seções (7.1) e (7.2).

O gráfico expresso pela Figura (5) revela a solução com o ganho mais significativo dentre todos os resultados avaliados neste trabalho, demonstrada por um pico superior a 35% e obtida através do algoritmo Delta-r_GRASP. Por outro lado, uma análise demonstra que este mesmo algoritmo obteve uma média de ganhos igual a 8,19%, valor inferior aos obtidos através dos algoritmos Delta-g_GRASP e DL_GRASP, que apresentaram ganhos médios de 10,61% e 11,13%, respectivamente.

Esse comportamento está possivelmente associado ao fato de que o algoritmo Delta-r é apresentado como a estratégia mais eficiente, dentre os três algoritmos gulosos retirados da literatura, para solucionar o problema estudado. Dessa forma, pressupõe-se que os resultados gerados por algoritmos derivados desta heurística tendem a se aproximar de soluções ótimas, restringindo-se a pontos de mínimo locais durante a realização da busca.

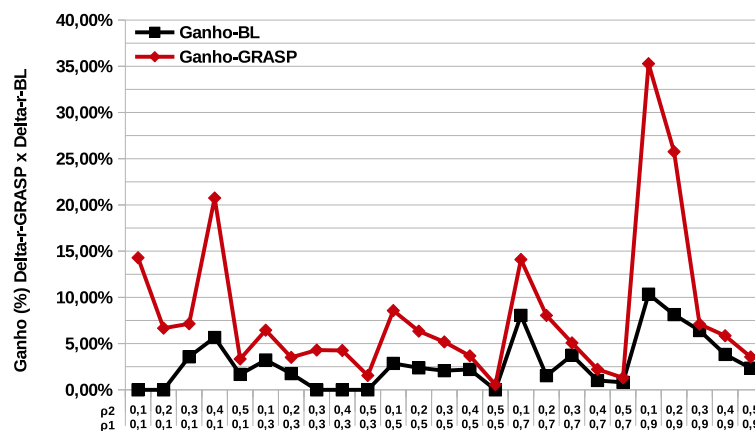


Figura 5: Ganho Percentual - Algoritmos Delta-r_BL e Delta-r_GRASP - Apresentam o ganho percentual em relação ao algoritmo DL original. O eixo x apresenta as 25 instâncias testadas (respectivamente ρ_2 e ρ_1) e o eixo y o ganho relativo. Quanto maior o ganho significa que o algoritmo apresentado conseguiu soluções melhores, isto é, com menos unidades de comunicação.

8. Conclusão

Nesse trabalho são apresentados três novos algoritmos para resolver o problema da Disposição $\Delta_{\rho_2}^{\rho_1}$, uma métrica utilizada para garantir qualidade de serviço em redes veiculares através da disposição eficiente de unidades de comunicação. Esses novos algoritmos, baseados na metaheurística GRASP, foram denominados DR_GRASP, Delta-g_GRASP e Delta-r_GRASP e utilizam como algoritmo guloso responsável pela Construção da Lista Restrita de Candidatos os algoritmos DL [Trullols et al., 2010], Delta-g [Trullols et al., 2010] e Delta-r [Sarubbi e Silva, 2016].

Os resultados apresentados demonstram que a utilização da heurística GRASP da forma como foi implementada melhorou significativamente os resultados da literatura, apontando ganhos

de até 35%. Entretanto, notou-se que para alguns casos (principalmente para valores altos de ρ_2) os algoritmos não apresentaram ganhos relevantes. Para melhorar esses resultados pretende-se aprimorar a técnica de busca local utilizada e explorar a aplicabilidade de algoritmos de perturbação sobre as soluções encontradas, com o intuito de percorrer uma gama maior de soluções.

Agradecimentos

Esse trabalho foi parcialmente financiado por FAPEMIG e CEFET-MG.

Referências

- Aissaoui, R., Menouar, H., Dhraief, A., Filali, F., Belghith, A., e Abu-Dayya, A. (2014). Advanced real-time traffic monitoring system based on v2x communications. In *Communications (ICC), 2014 IEEE International Conference on*, p. 2713–2718. IEEE.
- Blum, J., Eskandarian, A., e Hoffman, L. (2004). Challenges of intervehicle ad hoc networks. *Intelligent Transportation Systems, IEEE Transactions on*, 5(4):347–351. ISSN 1524-9050.
- Feo, T. e Resende, M. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–135.
- Hartenstein, H. e Laberteaux, K. (2008). A tutorial survey on vehicular ad hoc networks. *Communications Magazine, IEEE*, 46(6):164–171. ISSN 0163-6804.
- Lee, J. e Kim, C. (2010). A roadside unit placement scheme for vehicular telematics networks. In Kim, T.-h. e Adeli, H., editors, *Advances in Computer Science and Information Technology*, volume 6059 of *Lecture Notes in Computer Science*, p. 196–202. Springer Berlin Heidelberg. ISBN 978-3-642-13576-7.
- Li, Y., Jin, D., Hui, P., Wang, Z., e Chen, S. (2014). Limits of predictability for large-scale urban vehicular mobility. *Intelligent Transportation Systems, IEEE Transactions on*, 15(6):2671–2682. ISSN 1524-9050.
- Sarubbi, J. F. M., Martins, F. V. C., e Silva, C. M. (2016). A genetic algorithm for deploying roadside units in vanets. In *IEEE Congress on Evolutionary Computation (CEC), 2016 IEEE*.
- Sarubbi, J. F. M. e Silva, C. M. (2016). Delta-r: A novel and more economic strategy for allocating the roadside infrastructure in vehicular networks with guaranteed levels of performance. In *IEEE/IFIP Network Operations and Management Symposium (NOMS), 2016 IEEE*.
- Silva, C. M. e Meira, W. (2015a). Design of roadside communication infrastructure with qos guarantees. In *Symposium on Computers and Communications (ISCC), 2015 IEEE*.
- Silva, C. M. e Meira, W. (2015b). Evaluating the performance of heterogeneous vehicular networks. In *Vehicular Technology Conference (VTC), 2015 IEEE*.
- Tonguz, O. e Viriyasitavat, W. (2013). Cars as roadside units: a self-organizing network solution. *Communications Magazine, IEEE*, 51(12):112–120. ISSN 0163-6804.
- Trullols, O., Fiore, M., Casetti, C., Chiasserini, C., e Ordinas, J. B. (2010). Planning roadside infrastructure for information dissemination in intelligent transportation systems. *Computer Communications*, 33(4):432 – 442. ISSN 0140-3664.
- Zhang, D., Huang, H., Zhou, J., Xia, F., e Chen, Z. (2013). Detecting hot road mobility of vehicular ad hoc networks. *Mobile Networks and Applications*, 18(6):803–813. ISSN 1383-469X.
- Zheng, Z., Lu, Z., Sinha, P., e Kumar, S. (2010). Maximizing the contact opportunity for vehicular internet access. In *INFOCOM, 2010 Proceedings IEEE*, p. 1–9.
- Zheng, Z., Sinha, P., e Kumar, S. (2009). Alpha coverage: Bounding the interconnection gap for vehicular internet access. In *INFOCOM 2009, IEEE*, p. 2831–2835.